

An-Najah National University



Faculty of Engineering & Information Technology
Computer Engineering Department

Fall 2025/2026



Neureom

Modern E-Learning Platform with AI

Presented By:

Hamza Younes
Jehad Hmouda

Supervised by: Eng. Haya Samaana

Presented in partial fulfilment of the requirements for the Bachelor degree in Computer Engineering.

Acknowledgment

We would like to extend our sincere gratitude to our supervisor, Haya Samaana, whose guidance, support, and invaluable insights were instrumental to the success of this project. Their expertise in educational technology and software engineering helped us navigate complex technical challenges and maintain focus on delivering a platform that truly serves the needs of modern learners and educators.

Our appreciation also goes to the Department of Computer Engineering at An-Najah National University, whose comprehensive curriculum and supportive learning environment provided us with the technical foundation and problem-solving skills necessary to undertake this ambitious project.

To our families, we offer our deepest thanks for your unwavering support, patience, and encouragement throughout this journey. Your belief in our abilities gave us the strength to persevere through challenges and late nights of development.

We are also grateful to our colleagues and friends who provided valuable feedback, testing support, and constructive suggestions that helped shape Neureom into a more robust and user-friendly platform.

Finally, we are proud to have developed Neureom—a comprehensive online learning platform that leverages artificial intelligence to create personalized, engaging, and accessible educational experiences. This project represents our commitment to advancing education through technology and our belief in the transformative power of intelligent, well-designed learning systems.

Disclaimer

This report was authored and submitted by students from the Computer Engineering Department, Faculty of Engineering, at An-Najah National University, in partial fulfillment of the requirements for the Bachelor's degree in Computer Engineering.

The contents of this document represent the collective efforts, academic research, and technical implementation of the students throughout the course of their graduation project. While the work has undergone review, it has not been formally revised or corrected by the department or university, and may contain unintentional errors in language, formatting, or technical analysis.

The opinions, methodologies, outcomes, and recommendations expressed herein are those of the authors alone, and do not necessarily reflect the official views of the supervisor, department, or university.

Accordingly, An-Najah National University assumes no responsibility or liability for the consequences of applying, sharing, or interpreting the information contained in this report for any purpose beyond its intended academic evaluation.

Abstract

This report presents the design and development of Neureom, a comprehensive AI-powered online learning platform built to transform digital education through intelligent content delivery, personalized learning experiences, and advanced multimedia capabilities. The platform enables instructors to create structured courses with multiple content types—including video lessons, text-based content, and interactive quizzes—while students benefit from AI-enhanced learning tools, progress tracking, and community engagement features.

A core innovation of Neureom lies in its integration of artificial intelligence throughout the learning experience. Utilizing State of the art models from Google and OpenAI, the platform automatically generates course outlines, creates personalized learning roadmaps for students, transforms video lessons into multiple learning formats (summaries, flashcards, worksheets), and provides intelligent content recommendations. The AI-powered roadmap generator analyzes student goals and available courses to create customized 4-8 step learning paths with realistic time estimates and skill progression.

The platform features advanced video processing capabilities powered by Google Cloud Platform, including automatic transcoding to multiple quality levels, subtitle generation in both English and Arabic using OpenAI's whisper model, and adaptive streaming for optimal viewing experiences across different network conditions. Real-time communication is enabled through WebRTC-based voice and video calling, Socket.IO-powered notifications, and a comprehensive chat system built with Firebase.

Neureom supports multiple user roles (students and instructors) with dedicated dashboards, course management tools, and analytics. The platform includes a community forum for discussions, Stripe payment integration for course monetization, progress tracking with completion certificates, and a mobile-responsive design built with Next.js and React.

Developed over a comprehensive development cycle, the system was implemented using a modern technology stack: Next.js 14 for the frontend, MySQL for data persistence, Google Cloud Storage for media management, Firebase for real-time features, and Google Gemini AI for intelligent content generation. The platform demonstrates strong potential for revolutionizing online education by offering personalized, intelligent, and accessible learning experiences tailored to diverse educational needs.

Contents

Acknowledgment	1
Disclaimer	2
Abstract	3
Table of Contents	4
List of Figures	4
List of Tables	4
List of Abbreviations	5
1 Introduction	6
1.1 General Background	6
1.2 Problem Statement	7
1.3 Objectives of the Project	8
1.4 Significance of the Project	10
1.5 Organization of the Report	12
2 Theoretical Background and Previous Work	13
2.1 Online Learning and Digital Education	13
2.2 Artificial Intelligence in Education (AI-EdTech)	15
2.3 Video Processing and Adaptive Streaming	18
2.4 Related Work and Existing Platforms	21
2.5 Real-Time Communication in Learning Platforms	24
3 Methodology	30
3.1 Preface	30
3.1.1 Tools, Methods, and Programming Languages	30
3.1.2 Frontend Architecture	33
3.1.3 Backend Architecture	35
3.1.4 Database Design	36

3.2	System Features and Implementation	39
3.2.1	Authentication and User Management	39
3.2.2	Course Creation and Management	42
3.2.3	Lesson Types and Content Delivery	49
3.2.4	Video Processing and Streaming	55
3.2.5	Quiz System and Assessment	56
3.2.6	AI-Powered Features	57
3.2.7	Learning Roadmap Generator	57
3.2.8	Context-Aware AI Assistant	59
3.2.9	Real-Time Communication	63
3.2.10	Community Forum	66
3.2.11	Course Notification System	68
3.2.12	Payment Integration	69
3.2.13	Certificate Generation and Verification	71
3.2.14	Progress Tracking and Analytics	73
3.2.15	Content Moderation System	76
3.2.16	Admin User and Content Management	77
3.3	Standards and Specifications	82
3.3.1	Security Standards	82
3.3.2	API Design Standards	83
3.3.3	UI/UX Standards	83
4	Results and Analysis	85
4.1	System Implementation Results	85
4.2	Performance Analysis	89
4.3	Feature Validation	90
4.4	User Experience Evaluation	91
4.5	Testing Results	92
4.6	Challenges and Solutions	92
4.7	System Limitations	93
5	Discussion	94
5.1	Interpretation of Results	94
5.2	Challenges and Solutions Analysis	95
5.3	System Limitations Discussion	95
5.4	Comparison with Existing Platforms	96
5.5	Implications and Contributions	97
6	Conclusions and Recommendations	98
6.1	Project Summary	98

6.2	Contributions to the Field	99
6.3	Lessons Learned	99
6.4	Future Work	100
6.4.1	Scalability Enhancements	100
6.4.2	Feature Enhancements	101
6.4.3	Mobile Application	101
6.4.4	Accessibility and Internationalization	101
6.4.5	Assessment and Certification	102
6.4.6	Community and Social Learning	102
6.4.7	Integration and API	102
6.5	Conclusion	103

List of Figures

3.1	Neureom System Architecture Overview	33
3.2	Entity-Relationship Diagram of Neureom Database Schema	38
3.3	User Login Interface - Desktop View	39
3.4	User Login Interface - Mobile View	40
3.5	User Registration Form - Desktop View	41
3.6	User Registration Form - Mobile View	42
3.7	Course Builder Interface - Desktop View	43
3.8	Course Builder Interface - Mobile View	44
3.9	Course Catalog - Desktop View	45
3.10	Course Catalog - Mobile View	46
3.11	Course Details Page - Desktop View	47
3.12	Course Details Page - Mobile View	48
3.13	Video Lesson Player - Desktop View	50
3.14	Video Lesson Player - Mobile View	51
3.15	Text Lesson with Rich Content - Desktop View	52
3.16	Text Lesson with Rich Content - Mobile View	53
3.17	Quiz Interface - Desktop View	54
3.18	Quiz Interface - Mobile View	55
3.19	Video Processing Pipeline Workflow	56
3.20	AI-Generated Learning Roadmap - Desktop View	58
3.21	AI-Generated Learning Roadmap - Mobile View	59
3.22	Context-Aware AI Assistant - Desktop View	61

3.23	Context-Aware AI Assistant - Mobile View	62
3.24	AI Content Generation - Desktop View	63
3.25	AI Content Generation - Mobile View	63
3.26	WebRTC Call Signaling Sequence Diagram	64
3.27	Real-Time Chat Interface - Desktop View	65
3.28	Real-Time Chat Interface - Mobile View	66
3.29	Community Forum - Desktop View	67
3.30	Community Forum - Mobile View	68
3.31	Stripe Payment Checkout - Desktop View	70
3.32	Stripe Payment Checkout - Mobile View	71
3.33	Course Completion Certificate - Desktop View	73
3.34	Progress Tracking and Analytics - Desktop View	74
3.35	Progress Tracking and Analytics - Mobile View	75
3.36	Admin Dashboard Overview - Desktop View	78
3.37	Admin Dashboard Overview - Mobile View	79
3.38	Admin User Management - Mobile View	80
3.39	Content Moderation Dashboard - Desktop View	81
3.40	Financial Analytics Dashboard - Desktop View	81
3.41	Financial Analytics Dashboard - Mobile View	82
4.1	Student Dashboard Interface	85
4.2	Course Viewing Interface with Video Player	86
4.3	Context-Aware AI Assistant Interface	87
4.4	Course Completion Certificate	88

List of Tables

4.1	Video Processing Performance Metrics	89
4.2	AI Feature Response Times	90
4.3	Database Query Performance	90

List of Abbreviations

AI Artificial Intelligence	REST Representational State Transfer
API Application Programming Interface	RTL Right-to-Left
ARIA Accessible Rich Internet Applications	SDK Software Development Kit
CDN Content Delivery Network	SEO Search Engine Optimization
CORS Cross-Origin Resource Sharing	SFU Selective Forwarding Unit
CSRF Cross-Site Request Forgery	SQL Structured Query Language
CSS Cascading Style Sheets	SRT SubRip Subtitle
FCM Firebase Cloud Messaging	SSO Single Sign-On
GCP Google Cloud Platform	STUN Session Traversal Utilities for NAT
GCS Google Cloud Storage	TOC Table of Contents
HTML HyperText Markup Language	TURN Traversal Using Relays around NAT
HTTP HyperText Transfer Protocol	UI User Interface
HTTPS HyperText Transfer Protocol Secure	URL Uniform Resource Locator
ICE Interactive Connectivity Establishment	UX User Experience
JSON JavaScript Object Notation	WCAG Web Content Accessibility Guidelines
JWT JSON Web Token	WebRTC Web Real-Time Communication
LMS Learning Management System	WYSIWYG What You See Is What You Get
MySQL My Structured Query Language	XSS Cross-Site Scripting
NAT Network Address Translation	
PCI Payment Card Industry	

Chapter 1

Introduction

1.1 General Background

In the modern era, education is undergoing a fundamental transformation driven by rapid technological advancements and the increasing demand for flexible, accessible learning solutions. Traditional classroom-based education, while foundational, often struggles to accommodate the diverse learning styles, pacing preferences, and geographical constraints of today's learners. The global shift toward digital literacy and the widespread adoption of online platforms have created an unprecedented demand for educational solutions that are not only accessible but also interactive, personalized, and intelligent.

The COVID-19 pandemic accelerated this transformation, forcing educational institutions worldwide to adopt online learning models at an unprecedented scale. This shift revealed both the potential and limitations of existing e-learning platforms. While many platforms successfully facilitated remote content delivery, they often lacked sophisticated features for personalization, engagement, and intelligent content adaptation.

Artificial Intelligence (AI) and advanced web technologies have emerged as powerful tools in bridging the gap between traditional education and the dynamic needs of digital learners. Modern learning platforms are expected to do more than simply host videos and documents—they must provide personalized learning paths, adaptive content delivery, real-time interaction, and intelligent assessment mechanisms.

Video-based learning has become the cornerstone of online education, but it presents unique technical challenges including bandwidth optimization, quality adaptation, accessibility through subtitles, and engagement through interactive elements. Similarly, assessment mechanisms must evolve beyond simple multiple-choice questions to provide meaningful feedback and track learning progress effectively.

Neureom is born in response to this evolving educational landscape. It is a comprehen-

sive AI-powered online learning platform designed to address the multifaceted challenges of modern digital education. By leveraging cutting-edge technologies including Google Gemini AI for intelligent content generation, Google Cloud Platform for scalable video processing, WebRTC for real-time communication, and Next.js for a modern web experience, Neureom aims to redefine how educational content is created, delivered, consumed, and monetized in the digital age.

1.2 Problem Statement

Despite the proliferation of online learning platforms, several critical challenges continue to hinder the effectiveness and accessibility of digital education:

Content Creation Complexity: Instructors face significant barriers when creating online courses. Traditional platforms require manual structuring of content, lack intelligent assistance in course design, and provide limited tools for creating engaging multimedia experiences. The process of uploading, processing, and optimizing video content is often technically complex and time-consuming.

Lack of Personalization: Most existing platforms follow a one-size-fits-all approach to content delivery. Students with different learning styles, prior knowledge levels, and educational goals receive identical content without adaptation. There is minimal intelligent guidance on learning paths or personalized recommendations based on individual progress and objectives.

Video Accessibility and Quality: Video content, while central to online learning, often suffers from poor accessibility. Many platforms lack automatic subtitle generation, adaptive quality streaming, or multi-language support. Students with limited bandwidth or those requiring accessibility features are underserved.

Limited Interactivity: Traditional e-learning platforms primarily focus on passive content consumption through pre-recorded videos and static documents. Interactive elements such as real-time communication, collaborative learning, and dynamic content transformation are either absent or poorly implemented.

Assessment Limitations: Quiz and assessment systems in many platforms are basic, offering limited question types, no intelligent feedback, and minimal analytics on student performance. There is a lack of tools that help instructors understand learning gaps and

adapt their teaching accordingly.

Fragmented Learning Experience: Students often need to use multiple platforms for different aspects of their learning journey—one for courses, another for community interaction, a third for progress tracking, and yet another for career guidance. This fragmentation reduces engagement and makes it difficult to maintain a cohesive learning experience.

Monetization Challenges: Instructors seeking to monetize their expertise face technical barriers in implementing payment systems, managing enrollments, and tracking revenue. Many platforms either charge high commissions or provide inadequate tools for course creators to build sustainable educational businesses.

These challenges highlight the urgent need for a comprehensive, intelligent, and user-centric online learning platform that addresses content creation, personalized learning, advanced video processing, real-time interaction, intelligent assessment, community building, and seamless monetization within a unified ecosystem.

1.3 Objectives of the Project

The primary objective of this project is to develop Neureom, a comprehensive AI-powered online learning platform that modernizes digital education by introducing intelligent, personalized, and interactive tools for both instructors and students.

More specifically, the project aims to:

- **Empower Instructors:** Enable educators to create professional online courses with minimal technical expertise through:
 - AI-assisted course outline generation
 - Intuitive course builder with drag-and-drop module organization
 - Support for multiple content types (video, text, quizzes)
 - Automated video processing and optimization
 - Built-in payment integration for course monetization
 - Comprehensive analytics on student engagement and performance

- **Enhance Student Learning:** Provide students with personalized and engaging learning experiences through:
 - AI-powered learning roadmap generation based on individual goals
 - Multiple content formats generated from video lessons (summaries, flashcards, worksheets)
 - Adaptive video streaming with automatic quality adjustment
 - Multi-language subtitle support for accessibility
 - Interactive quizzes with instant feedback
 - Progress tracking and completion certificates
- **Integrate Advanced AI Capabilities:** Leverage artificial intelligence to:
 - Generate course outlines and descriptions from simple prompts
 - Create personalized 4-8 step learning paths for students
 - Transform video content into alternative learning formats
 - Provide intelligent content recommendations
 - Analyze learning patterns and suggest improvements
- **Enable Real-Time Communication:** Implement comprehensive communication features including:
 - WebRTC-based voice and video calling between students and instructors
 - Real-time chat system using Firebase
 - Socket.IO-powered instant notifications
 - Community forum for discussions and knowledge sharing
- **Ensure Scalability and Performance:** Build a robust technical infrastructure that:
 - Handles video transcoding to multiple quality levels (360p-1080p)
 - Stores and delivers media efficiently using Google Cloud Storage
 - Supports concurrent users with optimized database queries
 - Provides responsive performance across devices
- **Support Course Monetization:** Integrate payment processing to:
 - Enable instructors to set course prices
 - Process payments securely through Stripe
 - Handle enrollment management automatically

- Track revenue and payment history
- **Foster Community Engagement:** Create social learning features including:
 - Discussion forums with posts, comments, and reactions
 - User profiles with bio and activity tracking
 - Course-specific chat rooms
 - Collaborative learning opportunities

By achieving these objectives, the project intends to create a holistic learning ecosystem that addresses the full spectrum of online education needs—from content creation and delivery to assessment, communication, and monetization.

1.4 Significance of the Project

In an era where technology continues to reshape every aspect of our lives, education stands as one of the most critical fields requiring innovation. Neureom responds to this need by delivering a comprehensive platform that addresses multiple dimensions of modern digital education.

The significance of this project lies in several key areas:

- **Democratizing Quality Education:** By providing AI-assisted course creation tools, Neureom lowers the barrier for subject matter experts to share their knowledge online. Instructors without technical expertise can create professional courses with automated video processing, intelligent content structuring, and built-in monetization—democratizing access to educational content creation.
- **Personalized Learning at Scale:** The AI-powered roadmap generator represents a significant advancement in personalized education. Unlike traditional platforms that offer the same content to all learners, Neureom analyzes individual goals, current skill levels, and available courses to create customized learning paths. This personalization improves learning outcomes by ensuring students follow optimal progression routes tailored to their objectives.
- **Advanced Video Technology:** The platform’s sophisticated video processing pipeline including automatic transcoding to multiple quality levels, subtitle generation in multiple languages, and adaptive streaming—ensures that video content is accessible to

learners regardless of their bandwidth constraints or accessibility needs. This technical capability is particularly significant in regions with limited internet infrastructure.

- **AI-Enhanced Content Transformation:** The ability to transform video lessons into multiple formats (summaries, flashcards, worksheets) using AI addresses different learning styles and preferences. Visual learners, auditory learners, and kinesthetic learners can all engage with the same content in ways that suit their cognitive preferences, leading to improved retention and understanding.
- **Real-Time Interaction:** By integrating WebRTC-based calling, real-time chat, and instant notifications, Neureom bridges the gap between asynchronous online learning and the immediacy of traditional classroom interaction. Students can ask questions, collaborate with peers, and receive timely feedback from instructors, creating a more engaging and supportive learning environment.
- **Comprehensive Assessment:** The intelligent quiz system with multiple question types, automatic grading, progress tracking, and detailed analytics provides instructors with deep insights into student performance. This data-driven approach enables evidence-based teaching improvements and early identification of learning gaps.
- **Economic Empowerment:** The integrated payment system allows educators to monetize their expertise, creating economic opportunities and incentivizing the creation of high-quality educational content. This is particularly significant in developing regions where online education can provide sustainable income for skilled professionals.
- **Community-Driven Learning:** The community forum feature transforms isolated learning into a social experience. Students can discuss concepts, share insights, and learn from peers, fostering a sense of belonging and collaborative knowledge construction.
- **Technical Innovation:** The project demonstrates the successful integration of multiple cutting-edge technologies—Next.js for modern web development, Google Cloud Platform for scalable infrastructure, Google Gemini AI for intelligent features, WebRTC for peer-to-peer communication, and Firebase for real-time data—showcasing how these technologies can be orchestrated to solve complex educational challenges.
- **Market Relevance:** With the global e-learning market projected to exceed \$400 billion by 2026, Neureom addresses a real and growing market need. The platform's comprehensive feature set positions it competitively against established players while offering unique AI-powered capabilities that differentiate it in the marketplace.

In summary, this project not only delivers a practical solution for modern online education but also contributes to the broader goal of making quality education accessible, personalized, and engaging through intelligent technology.

1.5 Organization of the Report

This report is structured to provide a comprehensive and coherent understanding of the development and implementation of the Neureom online learning platform. Each chapter is designed to guide the reader through the various stages of the project, from background research to practical execution and evaluation.

- **Chapter 1: Introduction**

Offers an overview of the project, outlines its objectives, highlights its significance, and presents the overall structure of the report.

- **Chapter 2: Theoretical Background and Previous Work**

Explores existing literature, systems, and technologies relevant to online learning platforms, AI in education, video processing, and real-time communication, providing the foundation on which Neureom is built.

- **Chapter 3: Methodology**

Details the technical and procedural approach used in the development of the platform, including tools, frameworks, architecture, design decisions, implementation strategies, and constraints.

- **Chapter 4: Results and Analysis**

Presents the key outcomes of the development process, analyzes system performance, discusses user feedback, and evaluates system functionality across all features.

- **Chapter 5: Discussion**

Interprets the results, evaluates their impact in relation to the original objectives, and identifies the strengths and limitations of the platform.

- **Chapter 6: Conclusions and Recommendations**

Summarizes the findings, highlights lessons learned, and offers suggestions for future development and enhancements.

- **References**

Includes cited works and supporting materials that contributed to the research and development of the project.

Chapter 2

Theoretical Background and Previous Work

2.1 Online Learning and Digital Education

The rapid evolution of digital technologies has fundamentally transformed the landscape of education, giving rise to what is now widely referred to as online learning or e-learning. Online learning represents the use of internet-based technologies and digital platforms to deliver educational content, facilitate instruction, and enable learning activities outside the physical boundaries of traditional classrooms. This paradigm shift offers learners unprecedented flexibility, accessibility, and diversity in content formats, making education more inclusive and adaptable to individual needs.

Over the past two decades, online learning has gained significant traction across schools, universities, corporate training programs, and independent learning initiatives. The shift has been accelerated by several factors: increasing internet penetration globally, the proliferation of mobile devices, advances in web technologies, and most notably, the COVID-19 pandemic which forced educational institutions worldwide to adopt online learning models at an unprecedented scale. As a result, both educators and learners have become increasingly reliant on digital platforms to facilitate instruction, assessment, communication, and collaboration.

Modern online learning platforms have evolved far beyond simple video hosting services. Contemporary systems are equipped with sophisticated features including live and recorded video lectures, interactive assessments, discussion forums, assignment submission systems, progress tracking dashboards, certificate generation, and payment processing. These tools aim to replicate and enhance the classroom experience in a digital environment while adding capabilities that traditional education cannot easily provide—such as learning at one’s own pace, accessing world-class content from anywhere, and receiving

personalized feedback through data analytics.

However, despite these advancements, many online learning systems still face significant challenges. A primary concern is the lack of true personalization—most platforms deliver the same content to all learners regardless of their prior knowledge, learning style, or educational goals. This one-size-fits-all approach often leads to disengagement, particularly for students who find content either too basic or too advanced for their current level.

Another critical challenge is content quality and accessibility. Video content, while central to online learning, often suffers from poor production quality, lack of subtitles for accessibility, and inadequate adaptation to varying network conditions. Students in regions with limited bandwidth struggle to access high-quality video content, creating a digital divide that undermines the promise of universal access to education.

Engagement and interaction present additional challenges. Traditional online courses rely heavily on asynchronous content consumption—students watch pre-recorded videos and complete assignments independently, with limited opportunities for real-time interaction with instructors or peers. This isolation can lead to decreased motivation, higher dropout rates, and a sense of disconnection from the learning community.

Assessment mechanisms in many platforms remain rudimentary, offering basic multiple-choice quizzes with limited feedback. There is often insufficient data analytics to help instructors understand learning patterns, identify struggling students early, or adapt their teaching strategies based on evidence. The lack of intelligent assessment tools means that learning gaps may go undetected until it is too late for effective intervention.

A growing body of research in educational technology highlights the need for more learner-centered approaches that adapt to individual needs, learning styles, and performance patterns. This demand has paved the way for the integration of Artificial Intelligence (AI) into online learning platforms, enabling smarter systems that can offer real-time feedback, generate personalized content, provide adaptive learning paths, and foster more engaging educational experiences.

Furthermore, the economic model of online education has evolved. While early platforms focused primarily on free content distribution, there is now a recognized need for sustainable monetization models that compensate content creators fairly while keeping education accessible. Platforms that successfully balance quality content creation, fair compensation for educators, and affordability for learners are more likely to thrive in the competitive e-learning marketplace.

Neureom responds to these challenges by providing a comprehensive online learning platform that goes beyond traditional content delivery. By integrating AI-powered course generation, personalized learning roadmaps, advanced video processing with multi-quality transcoding and automatic subtitle generation, real-time communication tools, intelligent assessment systems, and community features, Neureom addresses the multifaceted needs of modern digital education. The platform recognizes that effective online learning requires not just content delivery, but a holistic ecosystem that supports creation, personalization, interaction, assessment, and community engagement—all working together to create meaningful learning experiences.

2.2 Artificial Intelligence in Education (AI-EdTech)

Artificial Intelligence (AI) is revolutionizing nearly every industry, and education is no exception. AI in education—often referred to as AI-EdTech—introduces intelligent systems that can analyze, adapt, and respond to educational content and learner behavior in ways that traditional systems cannot. This transformation aims not only to automate administrative tasks but also to create deeply personalized learning experiences that adapt to individual student needs, preferences, and performance patterns in real-time.

One of the core strengths of AI in education is its ability to understand and react to individual learning needs through data analysis and pattern recognition. By collecting and analyzing data on student performance, engagement levels, content preferences, time spent on activities, and progress patterns, AI-powered platforms can dynamically tailor content delivery, provide immediate feedback, suggest optimal learning paths, and identify areas where students struggle. For instance, a student having difficulty with a particular concept can be automatically redirected to simpler explanations, supplementary materials, practice exercises, or visual aids, while advanced learners can be challenged with more complex material to maintain engagement and promote deeper understanding.

AI also significantly enhances assessment and evaluation capabilities. Traditional assessment methods rely on manual grading, which is time-consuming and subject to human bias. AI-powered systems can automate grading for various question types including multiple-choice, fill-in-the-blank, and even open-ended responses using Natural Language Processing (NLP). Beyond simple grading, AI can detect patterns in student errors, identify common misconceptions, analyze learning gaps across cohorts, and provide instructors with actionable insights to improve their teaching strategies. This data-driven approach enables evidence-based educational improvements that were previously impossible

at scale.

Intelligent tutoring systems represent another significant application of AI in education. These systems act as virtual teaching assistants, providing personalized guidance, answering student questions, offering hints and explanations, and adapting their teaching strategies based on student responses. While not replacing human instructors, these AI tutors can provide 24/7 support, handle routine questions, and free up instructors to focus on more complex pedagogical tasks and personal interactions with students.

Content generation is an emerging and powerful application of AI in education. Modern large language models, such as Google's Gemini, OpenAI's GPT series, and others, can generate educational content including course outlines, lesson descriptions, quiz questions, study guides, summaries, and explanatory text. This capability dramatically reduces the time and effort required for content creation, allowing educators to focus on refining and personalizing content rather than creating it from scratch. However, it is crucial that AI-generated content undergoes human review to ensure accuracy, appropriateness, and alignment with learning objectives.

Personalized learning paths represent one of the most impactful applications of AI in education. Traditional curricula follow a linear, one-size-fits-all progression that may not suit all learners. AI can analyze a student's current knowledge level, learning goals, preferred learning styles, available time, and past performance to create customized learning roadmaps. These roadmaps suggest specific courses, lessons, and activities in an optimal sequence, adapting as the student progresses. This personalization ensures that each learner follows a path that is challenging yet achievable, maximizing engagement and learning outcomes.

AI-powered recommendation systems, similar to those used by Netflix or Spotify, can suggest relevant courses, lessons, or learning resources based on a student's interests, past behavior, and goals. These recommendations help students discover content they might not have found otherwise, keeping them engaged and expanding their learning horizons beyond their initial objectives.

In Neureom, AI plays a pivotal and multifaceted role in shaping the learning experience. Through the integration of Google Gemini AI, the platform offers several intelligent features:

- **AI-Assisted Course Creation:** Instructors can input a simple subject or topic, and the AI generates comprehensive course details including title, description, short de-

scription, suggested pricing, and difficulty level. This dramatically reduces the barrier to course creation and helps educators structure their content professionally.

- **Personalized Learning Roadmaps:** Students can specify their learning goals, current skill level, time commitment, and preferred topics. The AI analyzes available courses and generates a customized 4-8 step learning path with detailed descriptions, estimated durations, required skills for each step, and specific course recommendations. This feature transforms vague educational aspirations into concrete, actionable plans.
- **Content Transformation:** Video lessons can be automatically transformed into multiple learning formats—summaries for quick review, flashcards for memorization, and practice questions for self-assessment. The AI analyzes the lesson transcript (extracted from subtitles) and generates these alternative formats with proper LaTeX formatting for mathematical content. This multi-format approach accommodates different learning styles and preferences, allowing students to reinforce their understanding through varied study methods.
- **Intelligent Content Analysis:** The AI can analyze lesson transcripts to extract key concepts, generate questions, identify learning objectives, and create supplementary materials. This automated analysis saves instructors significant time while ensuring comprehensive coverage of topics.

The integration of AI in Neureom is designed with ethical considerations in mind. AI-generated content is clearly marked, allowing users to understand when they are interacting with AI-generated versus human-created content. The system is designed to augment rather than replace human instructors, recognizing that the human element—empathy, contextual understanding, and nuanced feedback—remains irreplaceable in education.

Furthermore, Neureom’s AI features are built with accessibility in mind. By generating subtitles, creating text alternatives to video content, and offering multiple content formats, the AI helps make learning more accessible to students with different abilities, language backgrounds, and learning preferences.

This fusion of AI and education not only streamlines content creation and delivery but also redefines the learning experience, shifting towards a more collaborative, efficient, and personalized ecosystem. AI-EdTech is not about replacing educators but empowering them—and their students—with intelligent tools that elevate the overall quality and effectiveness of education.

2.3 Video Processing and Adaptive Streaming

Video content has become the cornerstone of modern online education, serving as the primary medium through which instructors deliver lectures, demonstrations, and explanations. The effectiveness of video-based learning is well-documented in educational research—visual and auditory information combined creates stronger memory retention, complex concepts can be demonstrated more clearly, and students can pause, rewind, and review content at their own pace. However, delivering high-quality video content at scale presents significant technical challenges that must be addressed to ensure an effective learning experience.

One of the primary challenges in video-based education is bandwidth variability. Students access online learning platforms from diverse locations with vastly different internet connection speeds—from high-speed fiber optic connections in urban areas to limited mobile data in rural regions. A video encoded at a single quality level cannot serve all these users effectively. High-quality video may buffer constantly for users with slow connections, disrupting the learning experience and causing frustration, while low-quality video may appear pixelated and unclear on modern high-resolution displays, reducing comprehension and engagement.

Adaptive bitrate streaming (ABS) addresses this challenge by encoding video content at multiple quality levels and allowing the video player to dynamically switch between these levels based on the viewer's current network conditions. This technology, pioneered by streaming services like Netflix and YouTube, has become essential for delivering smooth video experiences across diverse network environments. The video player continuously monitors download speed and buffer status, seamlessly switching to higher quality when bandwidth is available and lower quality when the connection degrades, ensuring uninterrupted playback.

The process of creating multiple quality versions of a video is called transcoding. Transcoding involves decoding the original video file, re-encoding it at different resolutions and bitrates, and packaging the results in a format suitable for adaptive streaming. Common quality levels include 360p (low quality, suitable for slow connections), 480p (standard definition), 720p (HD), and 1080p (Full HD). Each quality level represents a trade-off between visual clarity and file size/bandwidth requirements.

Accessibility represents another critical dimension of video-based learning. Students with hearing impairments require subtitles or captions to access video content. Additionally,

subtitles benefit non-native speakers, students in noisy environments, and those who prefer reading along with audio. Manual subtitle creation is time-consuming and expensive, making it impractical for large-scale content production. Automatic speech recognition (ASR) technology, powered by machine learning, can generate subtitles automatically by transcribing spoken audio into text. Modern ASR systems achieve high accuracy rates and support multiple languages, making subtitle generation scalable and cost-effective.

Multi-language support extends the reach of educational content beyond linguistic boundaries. A course created in English can be made accessible to Arabic speakers, Spanish speakers, or speakers of any other language through translated subtitles. This democratizes access to quality education, allowing knowledge to flow across cultural and linguistic barriers.

Storage and delivery of video content at scale require robust infrastructure. Video files are large—a single hour of 1080p video can consume several gigabytes of storage. Multiply this by multiple quality levels and potentially hundreds or thousands of courses, and storage requirements become substantial. Furthermore, video delivery requires significant bandwidth, as multiple students may simultaneously stream different videos. Content Delivery Networks (CDNs) and cloud storage solutions like Google Cloud Storage, Amazon S3, or Azure Blob Storage provide the scalability needed to handle these demands efficiently.

In Neureom, video processing is handled through a sophisticated pipeline built on Google Cloud Platform (GCP) infrastructure:

Upload and Storage: When an instructor uploads a video lesson, the file is initially stored in Google Cloud Storage, a highly scalable and durable object storage service. GCS provides redundancy, ensuring that video files are not lost due to hardware failures, and offers high-speed access for subsequent processing and delivery.

Automatic Transcoding: Upon upload, Neureom triggers an automated transcoding process using Google Cloud Video Transcoder API or FFmpeg running on cloud compute instances. The original video is transcoded into multiple quality levels:

- **360p** (640×360 resolution) - Optimized for very slow connections and mobile data
- **480p** (854×480 resolution) - Standard definition for moderate connections
- **720p** (1280×720 resolution) - HD quality for good connections
- **1080p** (1920×1080 resolution) - Full HD for high-speed connections

Each quality level is encoded with appropriate bitrate settings to balance quality and file size. The transcoding process also generates thumbnail images for video previews and extracts metadata such as duration and aspect ratio.

Subtitle Generation: Neureom integrates automatic subtitle generation using a two-stage AI-powered process. The video's audio track is first processed through OpenAI's Whisper model, a state-of-the-art speech recognition system that transcribes spoken words into English text with precise timestamps. For Arabic subtitle support, the English transcription is then translated using Google Gemini AI, which provides high-quality translation while maintaining the original timing structure. The platform supports:

- **English subtitles** - Generated directly from audio using OpenAI Whisper
- **Arabic subtitles** - Translated from English subtitles using Google Gemini AI

To optimize processing time and handle longer videos, the system splits videos into 2-minute chunks, processes them in parallel with controlled concurrency, and then merges the results with adjusted timestamps. Subtitles are stored in standard SRT format and can be toggled on/off by students during video playback.

Adaptive Streaming: The transcoded videos are packaged for adaptive streaming using protocols like HLS (HTTP Live Streaming) or DASH (Dynamic Adaptive Streaming over HTTP). These protocols divide videos into small segments and provide manifest files that describe available quality levels. The video player uses this information to request appropriate segments based on current network conditions.

Efficient Delivery: Videos are delivered through Google Cloud Storage's global network, which acts as a CDN, ensuring low latency and high availability regardless of student location. The platform implements caching strategies to reduce bandwidth costs and improve performance.

Progress Tracking: The video player integrates with Neureom's backend to track student progress. When a student watches a video, their progress is periodically saved, allowing them to resume from where they left off if they navigate away or close the browser.

This comprehensive video processing pipeline ensures that Neureom delivers high-quality, accessible, and reliable video content to all students, regardless of their network conditions, device capabilities, or accessibility needs. By automating transcoding, subtitle

generation, and adaptive delivery, the platform removes technical barriers that often prevent educators from creating professional online courses, while simultaneously ensuring that students have the best possible viewing experience.

2.4 Related Work and Existing Platforms

Over the past decade, the global landscape of online education has witnessed a surge in the development of digital learning platforms. These systems have redefined traditional education by leveraging web and mobile technologies to provide scalable, accessible, and interactive learning experiences. The market is dominated by several well-established platforms, each offering distinct approaches to content delivery, student engagement, and course management.

Coursera and **edX** represent the massive open online course (MOOC) model, partnering with universities to offer academic courses and degree programs. These platforms excel at providing structured, university-level content with professional production quality. However, they primarily focus on pre-recorded video lectures with limited interactivity, and their content creation process is centralized and controlled by partner institutions, making it difficult for independent educators to participate.

Udemy and **Skillshare** operate on a marketplace model where individual instructors can create and sell courses. While these platforms democratize content creation, they offer limited tools for course creators—instructors must handle their own video production, and the platforms provide minimal AI assistance for content generation or personalization. Additionally, these platforms lack sophisticated features for real-time communication between students and instructors, relying primarily on asynchronous Q&A forums.

Khan Academy has pioneered free, high-quality educational content with an emphasis on K-12 subjects and standardized test preparation. The platform offers excellent content but follows a fixed curriculum with limited personalization. Students progress through predetermined paths without AI-driven customization based on individual learning goals or styles.

LinkedIn Learning (formerly Lynda.com) focuses on professional development and business skills. While it offers a vast library of courses, the platform lacks community features, real-time interaction, and personalized learning paths. Content is professionally produced but static, with no AI-powered transformation into alternative formats.

Teachable and **Thinkific** provide course creation platforms for educators to build their own branded learning sites. These platforms offer more control to instructors but require significant technical expertise to set up and manage. They lack integrated AI features, advanced video processing, or intelligent content generation tools.

More recent platforms have begun experimenting with AI-enhanced features. **Duolingo** integrates AI to tailor language lessons and optimize practice sessions based on learner behavior, demonstrating the potential of adaptive learning. **Socratic by Google** uses AI to help students solve homework questions by analyzing images and providing explanations. However, these applications focus on narrow domains rather than offering a comprehensive, flexible learning platform.

Despite the proliferation of these platforms, several critical gaps remain in the online education ecosystem:

Limited AI Integration: Most platforms use AI minimally, if at all. Content remains static, and there is little intelligent assistance for either content creation or personalized learning path generation. Instructors spend significant time creating course materials manually, and students receive the same content regardless of their learning preferences or goals.

Inadequate Video Processing: Many platforms accept video uploads but provide no automated transcoding, subtitle generation, or quality optimization. Instructors must handle video processing themselves or pay for third-party services, creating barriers to entry for educators without technical expertise or resources.

Weak Real-Time Communication: Traditional e-learning platforms rely heavily on asynchronous communication through forums and email. Real-time features like video calling, instant messaging, or live collaboration are either absent or poorly integrated, reducing the sense of community and immediate support that enhances learning.

Lack of Community Features: While some platforms offer discussion forums, few provide comprehensive community features like social posts, reactions, comments, and user profiles that foster peer-to-peer learning and engagement beyond course content.

Insufficient Personalization: Most platforms deliver identical content to all learners. There is minimal adaptation based on learning styles, prior knowledge, or individual goals. Students who prefer different content formats (text summaries, flashcards, practice questions) must create these materials themselves.

Complex Monetization: Platforms either take large commissions from course sales (30-50% on some marketplaces) or require instructors to manage payment processing independently. Few platforms offer seamless, integrated payment solutions with fair revenue sharing.

Neureom distinguishes itself by addressing these gaps through a comprehensive, AI-powered approach that integrates multiple advanced features into a unified platform:

Comprehensive AI Integration: Unlike platforms that use AI sparingly, Neureom integrates Google Gemini AI throughout the learning experience. Instructors receive AI assistance for course creation (generating titles, descriptions, and pricing suggestions), while students benefit from AI-generated learning roadmaps that create personalized 4-8 step paths based on their goals and available courses. Video lessons are automatically transformed into summaries, flashcards, and practice questions, accommodating diverse learning preferences without manual effort.

Advanced Video Processing Pipeline: Neureom automates the entire video processing workflow using Google Cloud Platform infrastructure. Upon upload, videos are automatically transcoded into four quality levels (360p, 480p, 720p, 1080p) for adaptive streaming, ensuring smooth playback regardless of network conditions. Subtitles are generated automatically in English using OpenAI Whisper and translated to Arabic using Gemini AI, making content accessible to diverse audiences without manual transcription.

Integrated Real-Time Communication: The platform includes WebRTC-based voice and video calling directly within the learning environment, allowing students and instructors to communicate face-to-face without leaving the platform or using third-party tools. A Firebase-powered chat system enables instant messaging, while Socket.IO delivers real-time notifications for course updates, assignment submissions, and system events.

Community-Driven Learning: Neureom features a comprehensive community forum where users can create posts, add reactions (like, love, insightful, etc.), write comments, and engage in discussions beyond course content. This social dimension fosters peer-to-peer learning, knowledge sharing, and a sense of belonging that traditional course platforms lack.

Intelligent Content Transformation: Rather than delivering video content in a single format, Neureom allows students to transform lessons into multiple formats suited to their learning preferences. The AI analyzes lesson transcripts and generates summaries for quick review, flashcards for memorization, and practice questions for self-assessment—all

with proper LaTeX formatting for mathematical content.

Seamless Payment Integration: The platform integrates Stripe payment processing, allowing instructors to monetize courses effortlessly. Students can enroll in free courses immediately or purchase paid courses through a secure checkout process. The system handles payment confirmation, enrollment automation, and revenue tracking without requiring instructors to manage payment infrastructure.

Multi-Language Support: With automated subtitle generation in English and Arabic, Neureom addresses linguistic diversity, making courses accessible to learners across different language backgrounds. This feature is particularly significant in regions where Arabic is the primary language of instruction.

Modular Course Structure: Unlike platforms that treat courses as monolithic entities, Neureom supports a hierarchical structure of courses, modules, and lessons. Instructors can organize content logically, and students can track progress at granular levels. Lessons can be video-based, text-based with rich block content, or quiz-based, providing flexibility in content delivery.

Comprehensive Quiz System: The platform includes a sophisticated quiz builder with multiple question types (multiple-choice, true/false, short answer), LaTeX support for mathematical notation, automatic grading, attempt tracking, and detailed analytics. This goes beyond the basic assessment tools offered by most platforms.

By integrating these features into a cohesive ecosystem, Neureom represents a significant advancement over existing platforms. It addresses the full spectrum of online education needs—from AI-assisted content creation and advanced video processing to real-time communication, community engagement, and intelligent personalization—all within a single, unified platform. This comprehensive approach positions Neureom as a next-generation learning platform that not only matches but exceeds the capabilities of established competitors.

2.5 Real-Time Communication in Learning Platforms

Real-time communication has become an essential component of modern online education, bridging the gap between the convenience of asynchronous learning and the immediacy of face-to-face interaction. While traditional e-learning platforms rely heavily

on pre-recorded content and delayed communication through forums or email, the integration of real-time features creates more engaging, interactive, and supportive learning environments that better replicate the dynamics of physical classrooms.

The importance of real-time communication in education is well-documented in pedagogical research. Immediate feedback, synchronous collaboration, and direct interaction with instructors and peers enhance learning outcomes, increase student motivation, and reduce feelings of isolation that often accompany online learning. Students who can ask questions and receive instant responses are more likely to stay engaged and persist through challenging material. Similarly, instructors benefit from real-time interaction by gaining immediate insights into student comprehension and adjusting their teaching strategies accordingly.

Several technologies enable real-time communication in web applications. **WebSockets** provide full-duplex communication channels over a single TCP connection, allowing servers to push data to clients instantly without polling. **WebRTC (Web Real-Time Communication)** is a collection of standards and protocols that enable peer-to-peer audio, video, and data sharing directly between browsers without requiring plugins or intermediate servers. **Long polling** and **Server-Sent Events (SSE)** offer alternative approaches for server-to-client communication, though with different trade-offs in terms of efficiency and browser support.

Modern real-time communication frameworks build on these underlying technologies to provide developer-friendly abstractions. **Socket.IO** is a popular JavaScript library that enables real-time, bidirectional communication between web clients and servers. It automatically selects the best available transport method (WebSocket, polling, etc.) based on browser capabilities and network conditions, providing a consistent API regardless of the underlying mechanism. **Firebase Realtime Database** and **Firebase Cloud Messaging** offer managed real-time data synchronization and push notification services, eliminating the need for developers to manage infrastructure for real-time features.

In educational contexts, real-time communication manifests in several key applications:

Instant Messaging: Text-based chat allows students to communicate with instructors and peers quickly, asking questions, sharing resources, and collaborating on assignments. Unlike forum posts that may go unanswered for hours or days, instant messaging provides immediate responses that maintain learning momentum.

Voice and Video Calling: Synchronous audio and video communication enables virtual

office hours, one-on-one tutoring sessions, study groups, and live discussions. These interactions provide the personal connection and nuanced communication (tone, facial expressions, body language) that text alone cannot convey.

Live Notifications: Real-time alerts inform users of important events—new course content, assignment deadlines, quiz results, forum replies, or incoming messages. These notifications keep learners engaged and ensure they don't miss critical information.

Collaborative Tools: Real-time document editing, shared whiteboards, and screen sharing enable students to work together on projects and problems, fostering collaborative learning even when physically separated.

Presence Indicators: Showing which users are currently online helps students identify when instructors or peers are available for immediate interaction, facilitating spontaneous learning conversations.

Despite the clear benefits, implementing robust real-time communication in educational platforms presents technical challenges. **Scalability** is a primary concern—maintaining persistent connections for thousands of concurrent users requires careful infrastructure design and resource management. **Network reliability** varies widely across users; systems must gracefully handle disconnections, reconnections, and varying bandwidth conditions. **Security and privacy** are critical, especially in educational contexts where minors may be involved; platforms must ensure that communication channels are properly authenticated, encrypted, and monitored. **Cross-platform compatibility** is essential, as users access platforms from diverse devices and browsers with varying levels of WebRTC support.

Neureom addresses these challenges through a comprehensive real-time communication architecture that integrates multiple technologies:

Socket.IO for Signaling and Notifications: The platform runs a custom Node.js server (`server.js`) that integrates Socket.IO with Next.js to handle real-time events. When users connect, they register their socket connection with their user ID, allowing the server to maintain a registry of online users. This infrastructure supports:

- **WebRTC Signaling:** Socket.IO manages the exchange of WebRTC offers, answers, and ICE (Interactive Connectivity Establishment) candidates between peers establishing voice or video calls. When a user initiates a call, the server relays signaling messages to the recipient, facilitating the peer-to-peer connection setup.

- **Call State Management:** The server tracks active call offers, including caller and receiver information, call type (voice or video), and connection state. This allows the system to handle call rejection, cancellation, and cleanup when calls end.
- **Real-Time Notifications:** Socket.IO delivers instant notifications for various events including incoming calls, new messages, course updates, and system alerts ensuring users receive timely information without page refreshes.

WebRTC for Peer-to-Peer Communication: Neureom implements WebRTC-based voice and video calling directly within the chat interface. Key features include:

- **Direct Peer-to-Peer Connections:** Once signaling is complete, audio and video streams flow directly between users' browsers without passing through the server, reducing latency and server bandwidth requirements.
- **Adaptive Quality:** WebRTC automatically adjusts stream quality based on network conditions, ensuring smooth communication even when bandwidth fluctuates.
- **Media Controls:** Users can mute/unmute audio, enable/disable video during calls, and end calls at any time. These controls are synchronized between peers through Socket.IO signaling.
- **STUN Server Integration:** The platform uses public STUN (Session Traversal Utilities for NAT) servers to help peers discover their public IP addresses and establish connections through NAT (Network Address Translation) devices.

Firestore for Push Notifications: To ensure users receive call notifications even when not actively viewing the platform, Neureom integrates Firebase Cloud Messaging (FCM). When a user initiates a call:

1. The system queries the database for the recipient's registered FCM tokens
2. Firebase Admin SDK sends push notifications to all the recipient's devices
3. Notifications include call metadata (caller name, call type, conversation ID)
4. Recipients can accept calls directly from the notification or within the app

Database-Backed Chat System: Real-time messaging is supported by a MySQL database schema that stores:

- **Chat Conversations:** One-to-one conversation records with normalized user pairs to prevent duplicates
- **Chat Messages:** Individual messages with sender information, timestamps, and conversation references
- **Call Signals:** WebRTC signaling data (offers, answers, ICE candidates) persisted for reliability
- **Call History:** Records of completed calls including duration, participants, and call type

This hybrid approach—using Socket.IO for real-time delivery while persisting data in MySQL—ensures message reliability. If a user is offline when a message is sent, they can retrieve it when they reconnect by querying the database.

Polling Fallback for Signals: To ensure reliability even when Socket.IO connections are unstable, the platform implements HTTP polling for call signals. Clients periodically query the `/api/calls/signal` endpoint to check for new offers, answers, or ICE candidates. This fallback mechanism ensures that calls can be established even in challenging network conditions where WebSocket connections may fail.

User Presence Tracking: The Socket.IO server maintains a real-time registry of connected users, tracking their socket IDs and user IDs. This enables presence indicators showing which users are currently online and available for immediate communication.

Security and Authentication: All real-time communication endpoints require authentication. Users must have valid session tokens to establish Socket.IO connections, send messages, or initiate calls. The system verifies that users are participants in conversations before allowing them to send or receive messages, preventing unauthorized access to private communications.

The integration of these real-time communication features transforms Neureom from a passive content delivery platform into an active learning community. Students can reach out to instructors instantly when stuck on a problem, form spontaneous study groups with peers, and receive immediate feedback on their work. Instructors can hold virtual office hours, conduct live Q&A sessions, and maintain closer connections with their students. The combination of asynchronous course content and synchronous communication provides the flexibility of online learning with the immediacy and personal connection of traditional education.

This comprehensive real-time communication infrastructure distinguishes Neureom from competitors that rely solely on asynchronous forums or require users to leave the platform to use third-party communication tools. By embedding voice calling, video calling, instant messaging, and live notifications directly into the learning environment, Neureom creates a more cohesive, engaging, and effective educational experience.

Chapter 3

Methodology

3.1 Preface

This chapter provides a comprehensive overview of the methodologies, technologies, and architectural decisions employed in the development of Neureom. It details the tools and frameworks selected, the rationale behind these choices, and the systematic approach taken to implement a scalable, maintainable, and feature-rich online learning platform.

3.1.1 Tools, Methods, and Programming Languages

The development of Neureom leverages modern web technologies and industry-standard tools to ensure code quality, developer productivity, and optimal user experience. The technology stack was carefully selected based on criteria including performance, scalability, community support, and alignment with project requirements.

Programming Languages:

- **TypeScript (v5.2.2):** The primary programming language for both frontend and backend development. TypeScript extends JavaScript with static type checking, enabling early detection of errors, improved code maintainability, and enhanced developer experience through intelligent code completion and refactoring tools.
- **JavaScript (Node.js):** Used for the custom server implementation (server.js) that integrates Socket.IO with Next.js for real-time communication features.
- **SQL:** Structured Query Language is used for database schema definition, data manipulation, and complex queries. The platform employs MySQL-specific syntax and features for optimal database performance.

Frontend Technologies:

- **React (v19.2.3)**: A declarative JavaScript library for building user interfaces with component-based architecture promoting code reusability and efficient rendering through virtual DOM implementation.
- **Next.js (v16.1.3)**: A React framework providing server-side rendering (SSR), static site generation (SSG), API routes, file-based routing, and automatic code splitting.
- **TailwindCSS (v3.3.3)**: A utility-first CSS framework enabling rapid UI development through composable utility classes.
- **Radix UI**: A comprehensive collection of unstyled, accessible UI components ensuring accessibility compliance (ARIA attributes, keyboard navigation).
- **shadcn/ui**: Re-usable components built on Radix UI and styled with TailwindCSS following modern design principles.

Backend Technologies:

- **Next.js API Routes**: Server-side API endpoints handling authentication, database operations, file uploads, payment processing, and external service integrations.
- **Socket.IO (v4.8.3)**: Library for real-time, bidirectional communication between clients and servers, handling WebRTC signaling, live notifications, and presence tracking.

Database and Data Management:

- **MySQL (v3.11.5 via mysql2)**: Relational database management system providing ACID compliance, transaction support, and efficient query execution with connection pooling.

AI and Machine Learning:

- **Google Gemini AI (v0.24.1)**: Integration with Google's Gemini 2.0 Flash model for course generation, learning roadmap creation, content transformation, and Arabic translation.

- **OpenAI API (v6.16.0)**: Integration with OpenAI's Whisper model for automatic speech recognition and English subtitle generation.

Cloud Services and Infrastructure:

- **Google Cloud Platform (GCP)**: Primary cloud infrastructure provider offering Google Cloud Storage, Google Cloud Video Transcoder API, and AI services.
- **Firestore Admin SDK (v13.6.0)**: Integration with Firebase Cloud Messaging (FCM) for push notifications.
- **Stripe (v20.2.0)**: Payment processing platform for course monetization.

Video Processing:

- **FFmpeg (via fluent-ffmpeg v2.1.3)**: Multimedia framework for video processing, metadata extraction, thumbnail generation, and audio extraction.
- **Sharp (v0.34.5)**: High-performance image processing library for thumbnail optimization and format conversion.

Mobile Application Technologies:

- **Flutter (SDK ^3.9.2)**: Google's UI toolkit for building natively compiled applications from a single codebase using Dart programming language.
- **flutter_inappwebview (v6.1.5)**: Flutter plugin providing an in-app WebView widget for displaying web content within native mobile wrapper.

This comprehensive technology stack provides a solid foundation for building a scalable, performant, and feature-rich educational platform while maintaining code quality and developer productivity across web and mobile platforms.

3.1.2 Frontend Architecture

Neureom provides both web and mobile frontend experiences, each optimized for their respective platforms while sharing core functionality through the web application.

Web Frontend Architecture:

The frontend architecture follows modern React and Next.js best practices, emphasizing component reusability, type safety, and optimal performance. The architecture is built on Next.js 16's App Router, which provides enhanced routing capabilities, improved data fetching patterns, and React Server Components support.

The frontend architecture follows modern React and Next.js best practices, emphasizing component reusability, type safety, and optimal performance. The architecture is built on Next.js 16's App Router, which provides enhanced routing capabilities, improved data fetching patterns, and React Server Components support. Figure 3.1 illustrates the overall system architecture showing the interaction between frontend, backend, database, and external services.

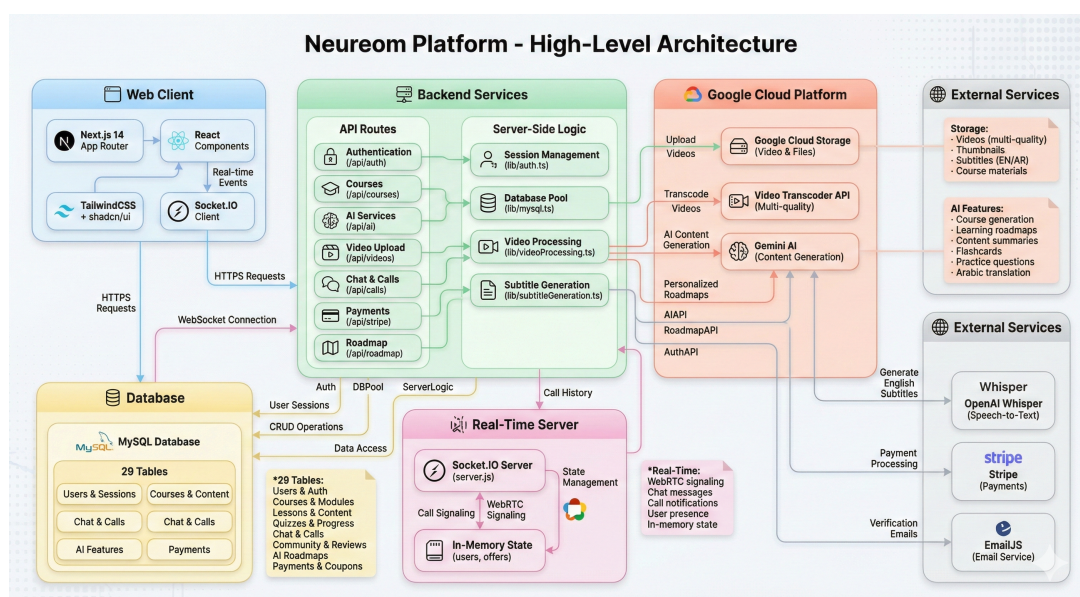


Figure 3.1: Neureom System Architecture Overview

Application Structure:

- **app/layout.tsx:** Root layout component wrapping the entire application, providing global providers including ThemeProvider for dark/light mode support.

- **app/[lang]/**: Language-specific routes supporting internationalization (i18n) with dynamic language segments for English and Arabic.
- **app/[lang]/dashboard/**: Protected dashboard area containing authenticated user features including courses, community, chat, roadmap, and profile management.
- **app/api/**: Server-side API routes handling authentication, database operations, file uploads, and payment processing.

Component Architecture:

- **components/ui/**: Reusable UI primitives built on Radix UI and styled with TailwindCSS forming the foundation of the design system.
- **components/course-builder/**: Specialized components for course creation and management.
- **components/lesson/**: Components for lesson content display, video players, AI content tabs, and progress tracking.

Performance Optimizations:

- **Code Splitting**: Automatic route-based code splitting via Next.js
- **Image Optimization**: Next.js Image component with automatic format selection and lazy loading
- **Font Optimization**: Next.js font optimization for web fonts
- **Component Lazy Loading**: Dynamic imports for heavy components

Mobile Frontend Architecture:

The mobile application is built using Flutter, providing native iOS and Android applications that wrap the web platform in an optimized mobile experience using a hybrid WebView approach. This architecture offers code reuse, consistent experience across platforms, and native integration for push notifications and platform-specific UI elements.

3.1.3 Backend Architecture

The backend architecture combines Next.js API Routes with a custom Node.js server, providing both RESTful API endpoints and real-time WebSocket communication.

Server Architecture:

- **Custom Node.js Server (server.js):** Wraps Next.js and integrates Socket.IO for real-time features, handling HTTP requests through Next.js while managing WebSocket connections.
- **Next.js API Routes:** Server-side endpoints organized under `app/api/` following RESTful conventions.

API Route Organization:

- `/api/auth/`: Authentication endpoints (login, register, logout, email verification)
- `/api/courses/`: Course CRUD operations and enrollment
- `/api/modules/`: Module management within courses
- `/api/lessons/`: Lesson creation, updates, and content management
- `/api/quizzes/`: Quiz creation and question management
- `/api/videos/`: Video upload and transcoding status
- `/api/ai/`: AI-powered features (course generation, thumbnails)
- `/api/roadmap/`: Learning roadmap generation and progress tracking
- `/api/chat/`: Chat conversations and messages
- `/api/calls/`: WebRTC signaling and call history
- `/api/stripe/`: Payment processing and webhooks

Authentication and Authorization:

The platform implements session-based authentication with bcrypt password hashing, email verification, JWT-like tokens stored in HTTP-only cookies, and role-based access control verifying user ownership of resources.

Security Measures:

- **Input Validation:** Zod schemas validate request payloads
- **SQL Injection Prevention:** Parameterized queries via mysql2
- **CORS Configuration:** Controlled cross-origin access
- **HTTPS Enforcement:** Required in production

3.1.4 Database Design

Neureom employs a relational database design using MySQL, chosen for its ACID compliance, transaction support, and mature ecosystem. The database schema is normalized to third normal form (3NF) to minimize redundancy while maintaining query performance through strategic indexing. Figure 3.2 presents the complete entity-relationship diagram showing all 31 database tables and their relationships.

Core Tables:

Users and Authentication:

- **users:** User accounts with email, password hash, profile information, verification status
- **sessions:** Active user sessions with token hashes and expiration timestamps
- **user_push_tokens:** Firebase Cloud Messaging tokens for push notifications

Course Structure:

- **courses:** Course metadata including title, description, pricing, creator, status
- **modules:** Course modules with ordering and titles
- **lessons:** Individual lessons with type (video, text, quiz), content, and video metadata

- **video_qualities**: Multiple quality versions (360p-1080p) for each video lesson
- **lesson_blocks**: Rich text content blocks for text-based lessons

Assessment System:

- **quizzes**: Quiz configuration with passing scores, time limits, attempts
- **questions**: Quiz questions with types, correct answers, explanations
- **quiz_attempts**: Student quiz submissions with scores and timestamps

Learning Roadmaps:

- **roadmaps**: Personalized learning paths with goals and progress
- **roadmap_steps**: Individual steps in a roadmap with descriptions and skills
- **roadmap_step_courses**: Course recommendations for each roadmap step

Communication:

- **chat_conversations**: One-to-one chat conversations with normalized user pairs
- **chat_messages**: Individual chat messages with sender and timestamps
- **call_signals**: WebRTC signaling data for voice/video calls
- **call_history**: Records of completed calls with duration and participants

Indexing Strategy:

The database employs strategic indexing to optimize query performance with primary keys, foreign key indexes, composite indexes for common query patterns, and unique constraints to prevent duplicate data.

Referential Integrity:

Foreign key constraints ensure data consistency with CASCADE DELETE for child records, SET NULL for optional relationships, and RESTRICT to prevent deletion of records with dependent data.



Figure 3.2: Entity-Relationship Diagram of Neureom Database Schema

This comprehensive database design provides a solid foundation for data integrity, query performance, and feature scalability while maintaining clear relationships between entities and supporting complex educational workflows.

3.2 System Features and Implementation

This section provides detailed descriptions of Neureom’s core features, explaining their implementation, user workflows, and technical architecture.

3.2.1 Authentication and User Management

Authentication and user management form the foundation of Neureom’s security architecture. The registration process implements secure multi-step workflow with input validation using Zod schemas, duplicate prevention, bcrypt password hashing (cost factor 10), cryptographically secure token generation, and automated email verification. Sessions are managed through database storage with 14-day expiration, HTTP-only cookies preventing XSS attacks, and SHA-256 hashed tokens. The `requireUser()` middleware function protects API routes by validating session tokens and checking email verification status. Security measures include bcrypt password hashing, secure token generation, HTTP-only cookies, HTTPS enforcement, CSRF protection via SameSite cookie attribute, and SQL injection prevention through parameterized queries.

Figures 3.3 through 3.6 show the authentication interfaces across desktop and mobile devices.

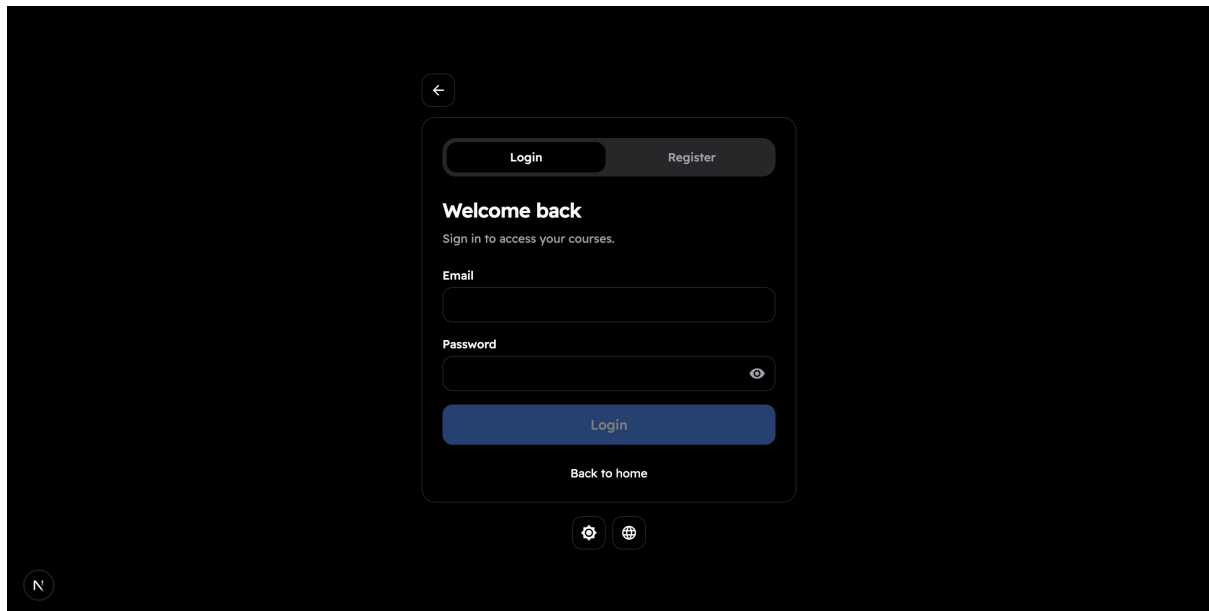


Figure 3.3: User Login Interface - Desktop View

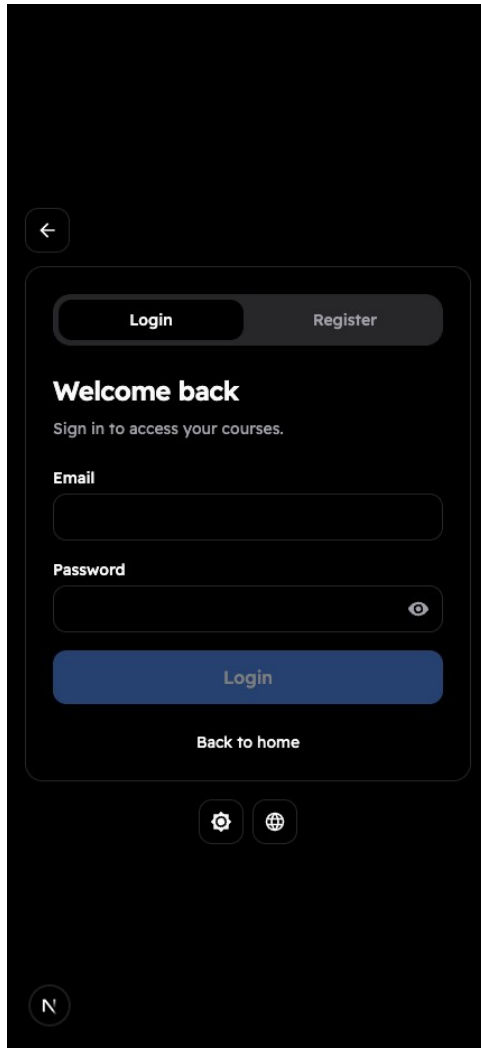
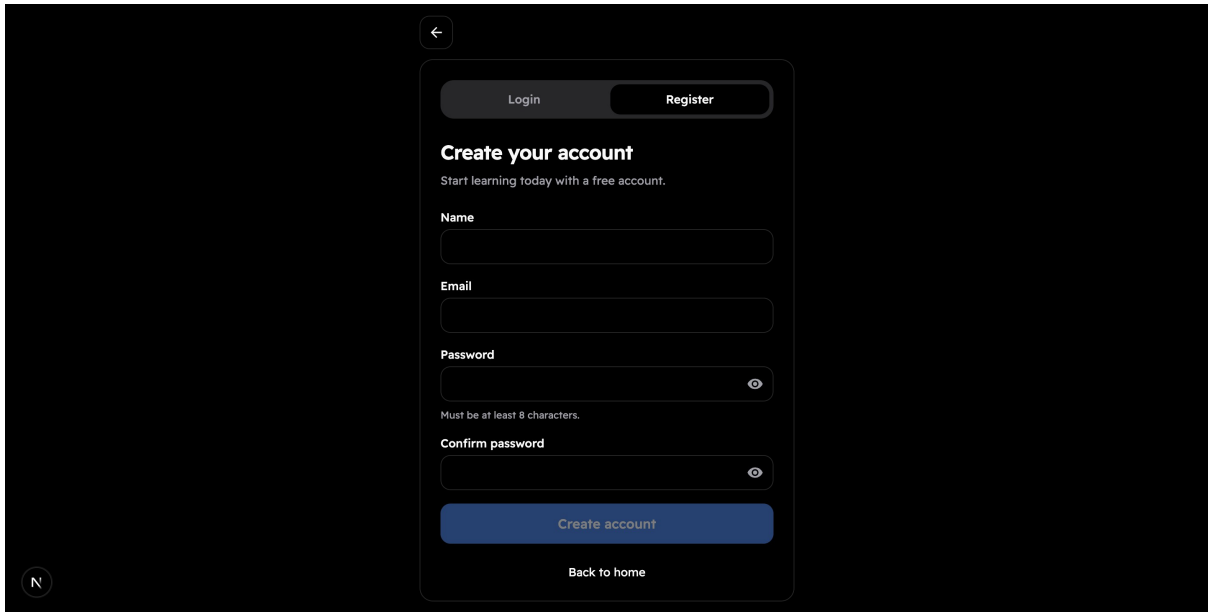


Figure 3.4: User Login Interface - Mobile View



The image shows a user registration form on a dark background. At the top left is a back arrow icon. Below it are two buttons: "Login" and "Register". The main heading is "Create your account" with a subtext "Start learning today with a free account." The form contains four input fields: "Name", "Email", "Password", and "Confirm password". The "Password" and "Confirm password" fields have eye icons for toggling visibility. Below the "Password" field is a note: "Must be at least 8 characters." At the bottom of the form is a blue "Create account" button and a "Back to home" link. A small "N" icon is visible in the bottom left corner of the form area.

Figure 3.5: User Registration Form - Desktop View

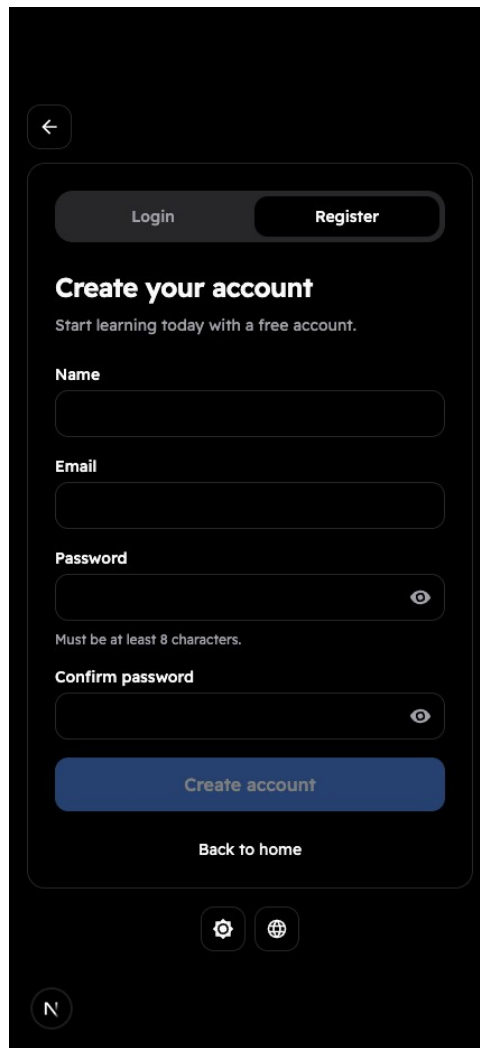


Figure 3.6: User Registration Form - Mobile View

3.2.2 Course Creation and Management

Course creation implements a three-tier hierarchical structure: Courses (top-level containers), Modules (logical groupings), and Lessons (individual learning units). Instructors create courses with title, description, pricing, difficulty level, and language. New courses start in draft status, invisible to students until published. Module management provides automatic ordering, drag-and-drop reordering, and cascading deletion. The enrollment system handles both free courses (immediate enrollment) and paid courses (Stripe checkout redirect). Course access control implements role-based permissions with creators having full CRUD operations and students having read-only access to enrolled courses. The platform supports both soft delete (archive) and hard delete (permanent removal with file cleanup).

Figures 3.7 through 3.12 illustrate the course creation and browsing interfaces across devices.

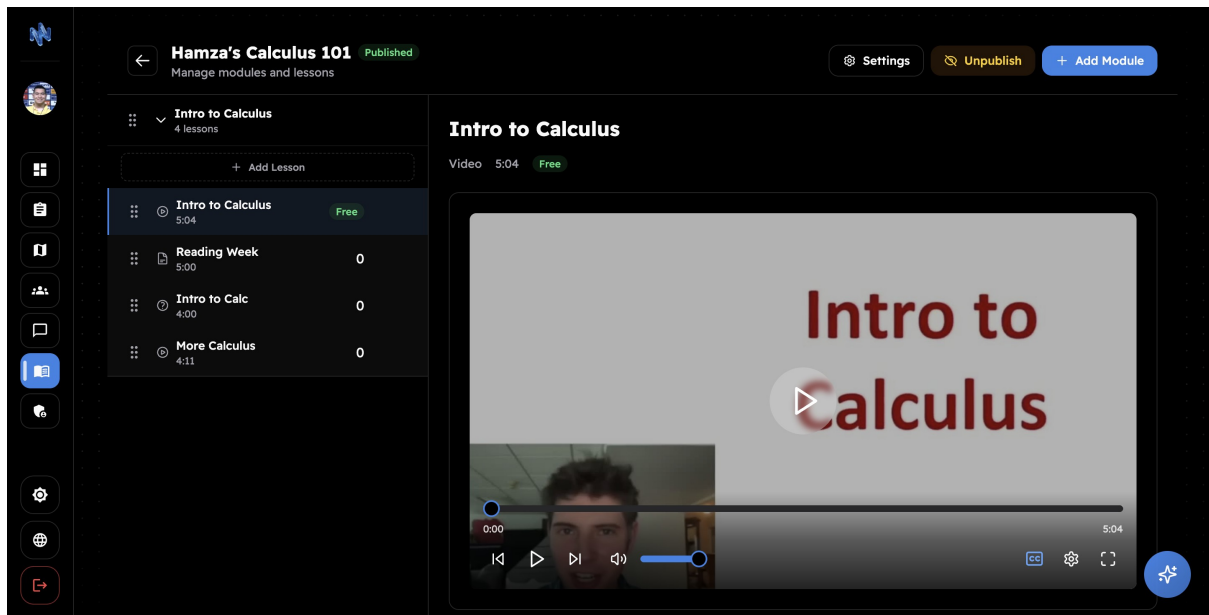


Figure 3.7: Course Builder Interface - Desktop View

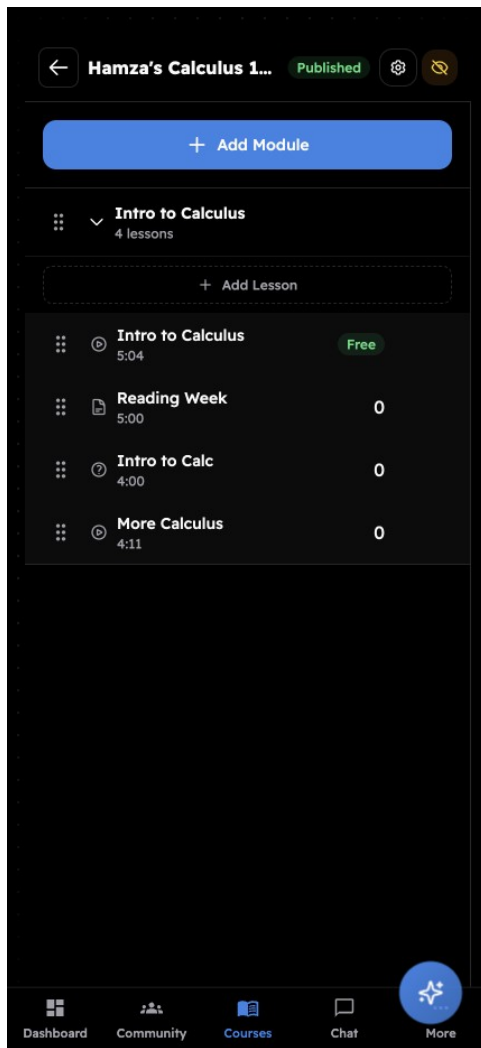


Figure 3.8: Course Builder Interface - Mobile View

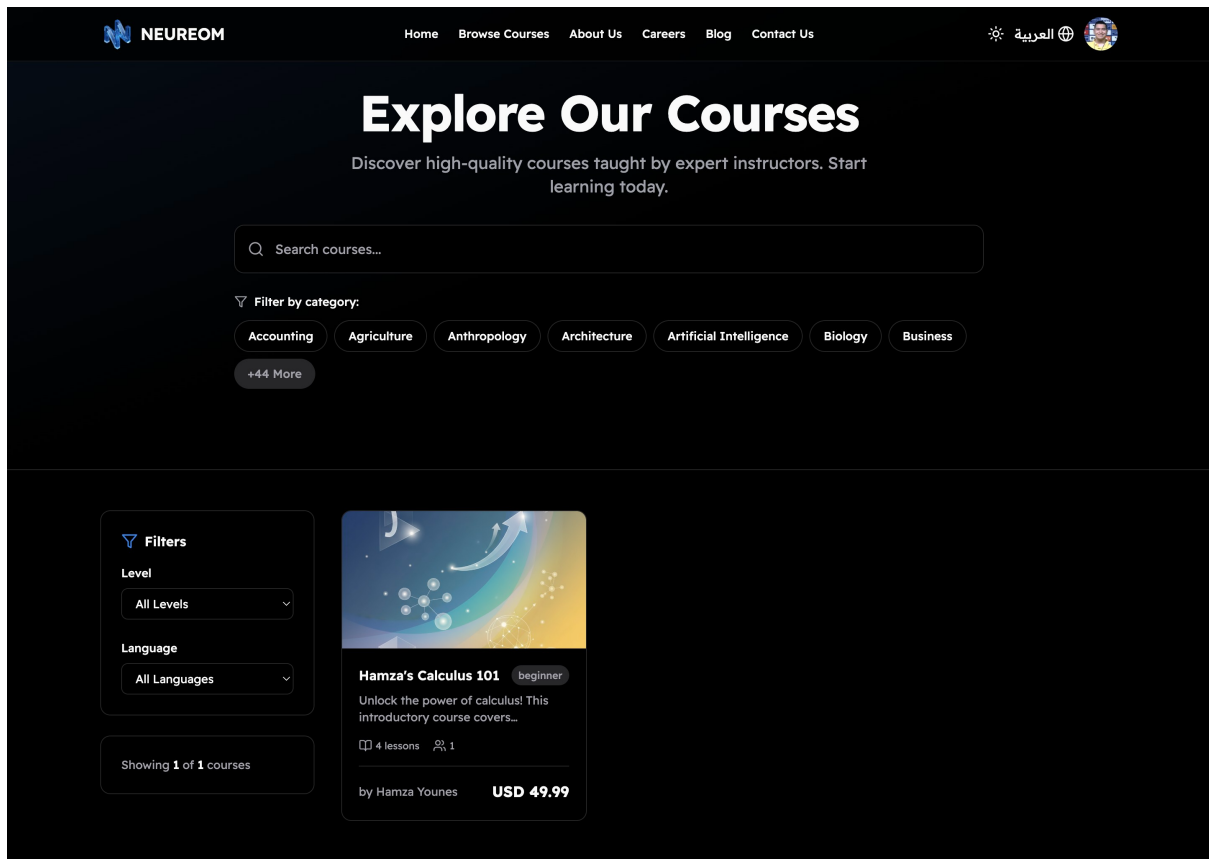


Figure 3.9: Course Catalog - Desktop View

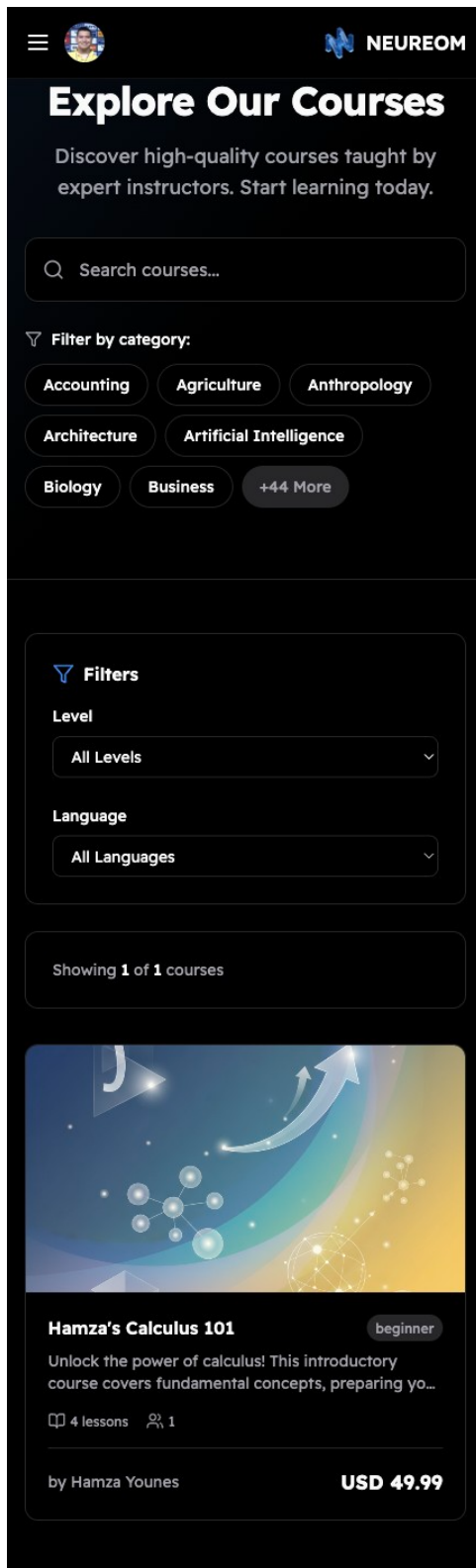


Figure 3.10: Course Catalog - Mobile View



Courses / Hamza's Calculus 101

Hamza's Calculus 101

Unlock the power of calculus! This introductory course covers fundamental concepts, preparing you for advanced math and STEM fields.

★ 0.0 👤 1 students 📖 4 lessons ⌚ 18m 15s

🔑 Beginner

Created by **Hamza Younes**



USD 49.99

Enroll

- ✔ Lifetime access
- ✔ Certificate of completion
- ✔ Mobile and desktop access

About This Course

This course provides a comprehensive introduction to single-variable calculus. Students will learn about limits, derivatives, and integrals, and their applications in various fields. We will explore techniques for calculating derivatives and integrals, and learn how to use them to solve optimization problems, analyze rates of change, and calculate areas and volumes.

This course is perfect for high school students, college freshmen, or anyone looking to build a solid foundation in calculus. No prior calculus knowledge is required, but a strong understanding of algebra and trigonometry is recommended.

By the end of this course, you'll be able to confidently tackle calculus problems, understand the underlying principles, and appreciate the beauty and power of this essential branch of mathematics.

Course Curriculum

1 modules • 4 lessons • 18m 15s total length

- 1 **Intro to Calculus**
4 lessons • 18m 15s

3.2.3 Lesson Types and Content Delivery

Neureom supports three lesson types: Video lessons with adaptive streaming and subtitles, Text lessons with block-based rich content (11 block types including text, heading, code, quote, image, video, list, divider, callout, table, PDF), and Quiz lessons for assessment. Video lessons store multiple quality levels (360p, 480p, 720p, 1080p) in the `video_qualities` table with dual-language subtitle support (English via OpenAI Whisper, Arabic via Google Gemini translation). Text lessons use a flexible content block architecture stored in the `content_blocks` table with JSON data for block-specific configuration. Quiz lessons link to the `quizzes` table supporting four question types (multiple choice, true/false, short answer, multiple select) with configurable passing scores, time limits, and attempt tracking. Lesson progress tracking maintains status (`not_started`, `in_progress`, `completed`) with completion criteria varying by type: 90% watch time for videos, scroll-to-bottom for text, and passing score for quizzes.

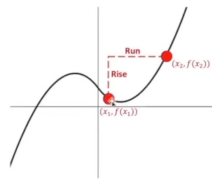
Figures 3.13 through 3.18 demonstrate the three lesson types across desktop and mobile devices.



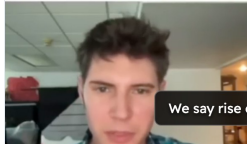
H

Course Progress

100%



$$\text{Average Rate of Change} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$



Summary

Flashcards

Practice

Lesson Summary

Regenerate

This lesson introduces the fundamental concepts of calculus, starting with the familiar idea of the slope of a line. The slope is defined as rise over run, represented as $m = \frac{y_2 - y_1}{x_2 - x_1}$. Function notation is introduced, replacing y with $f(x)$, leading to the average rate of change formula: $\frac{f(x_2) - f(x_1)}{x_2 - x_1}$. When dealing with curves, the average rate of change is visualized as the slope of the secant line connecting two points on the curve. By introducing a distance h such that $x_2 = x + h$, the average rate of change can be rewritten as $\frac{f(x+h) - f(x)}{h}$. The core concept of a limit is then introduced to shrink h towards zero. Taking the limit as h approaches zero, denoted as $\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$, transforms the secant line into a tangent line at a single point, known as the point of tangency. This limit gives the instantaneous rate of change, also known as the derivative. The derivative is demonstrated with the example of $f(x) = x^2$. Through algebraic manipulation, specifically expanding $(x + h)^2$, canceling terms, factoring out h , and taking the limit, the derivative is found to be $f'(x) = 2x$. This result allows the calculation of the slope of the tangent line at any point on the curve, such as at $x = 1$, where the slope is $f'(1) = 2$.

Key Points:

- The slope of a line is defined as $m = \frac{y_2 - y_1}{x_2 - x_1}$, representing rise over run.
- Function notation replaces y with $f(x)$, resulting in the average rate of change formula: $\frac{f(x_2) - f(x_1)}{x_2 - x_1}$.
- For curves, the average rate of change is the slope of the secant line.
- Introducing h , where $x_2 = x + h$, leads to the formula: $\frac{f(x+h) - f(x)}{h}$.
- The limit as h approaches zero, $\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$, defines the derivative.
- The derivative represents the instantaneous rate of change and the slope of the tangent line.
- For the example $f(x) = x^2$, the derivative is calculated as $f'(x) = 2x$.
- The derivative can be used to find the slope of the tangent line at a specific point, such as $f'(1) = 2$.

Intro to Calculus

5h 4m Completed

Course Content

1 Intro to Calculus

4 lessons

Intro to Calculus

5h 4m

Reading Week

5h 0m

Intro to Calc

4h 0m

More Calculus

4h 11m



Figure 3.13: Video Lesson Player - Desktop View

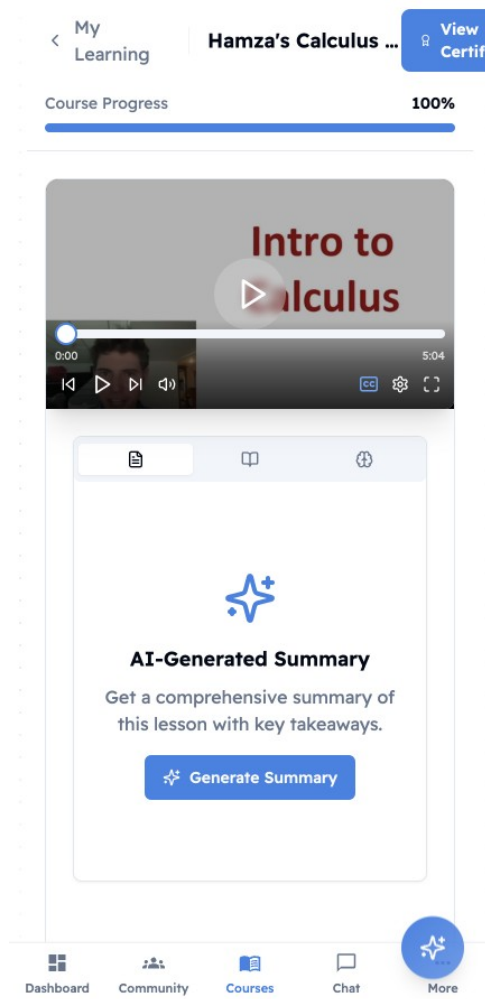


Figure 3.14: Video Lesson Player - Mobile View

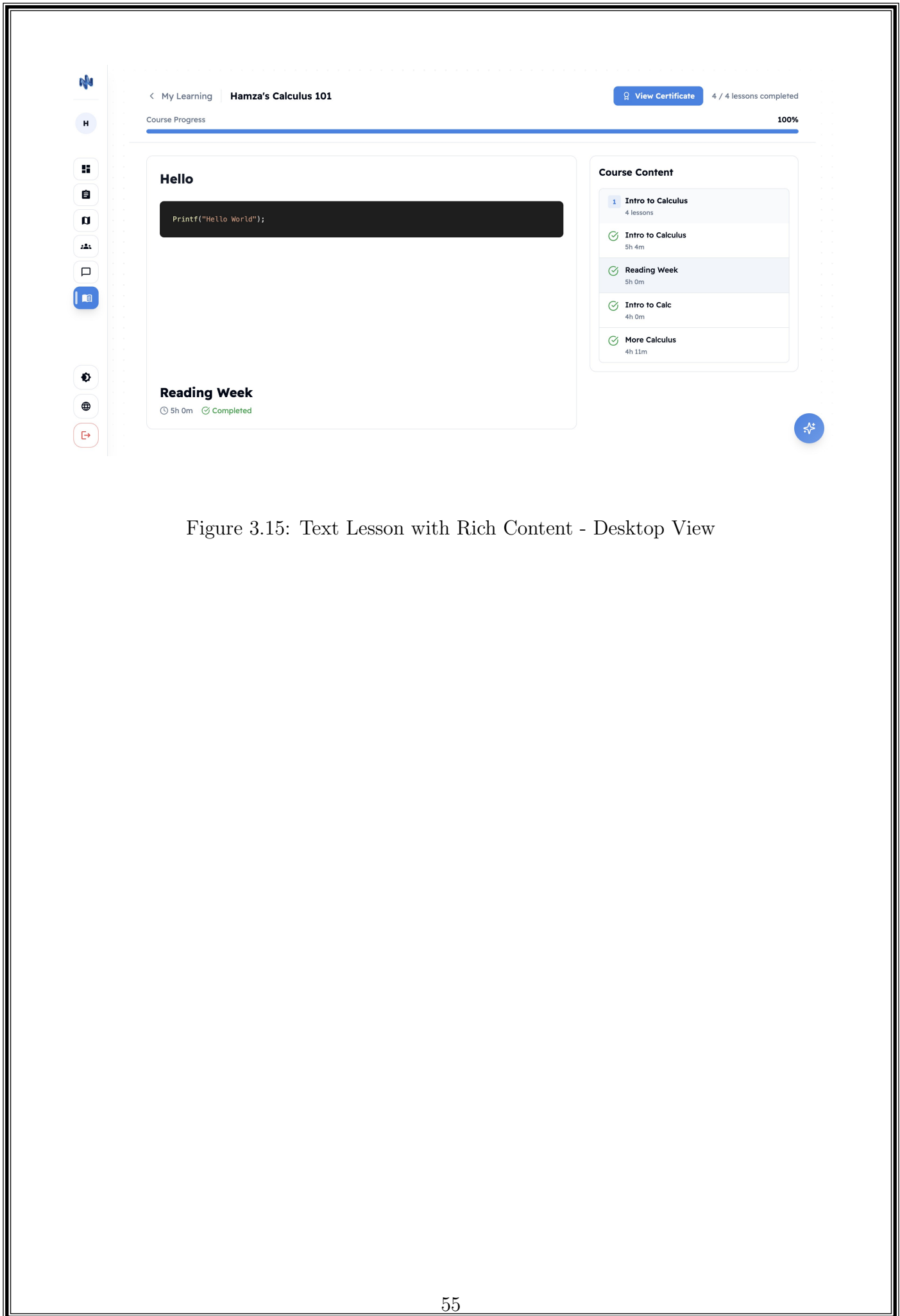


Figure 3.15: Text Lesson with Rich Content - Desktop View

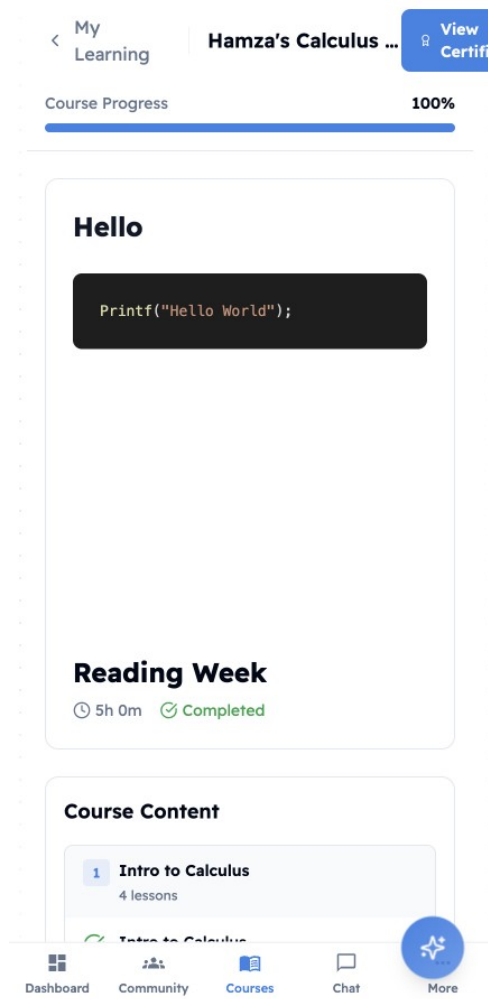


Figure 3.16: Text Lesson with Rich Content - Mobile View



H



← Back to Quiz



Attempt 3 Results

Score: 100% (2/2 correct)

Passed

1/26/2026

Question 1

Correct

What is the geometric interpretation of the average rate of change of a function $f(x)$ between two points $(x_1, f(x_1))$ and $(x_2, f(x_2))$?

The area under the curve of $f(x)$ between x_1 and x_2 .

The slope of the tangent line to $f(x)$ at $x = x_1$.

The slope of the secant line connecting the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$.

Your Answer

The instantaneous rate of change of $f(x)$ at $x = x_2$.

Explanation

The average rate of change is calculated as $\frac{f(x_2) - f(x_1)}{x_2 - x_1}$, which represents the slope of the secant line connecting the two points on the graph of $f(x)$.

Question 2

Correct

The limit definition of the derivative of a function $f(x)$ is given by $\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$. What does this limit represent?

The average rate of change of $f(x)$ over an interval of length h .

The slope of the secant line of $f(x)$.

The instantaneous rate of change of $f(x)$ at the point x .

Your Answer

The area under the curve of $f(x)$.

Explanation

The limit definition of the derivative gives the slope of the tangent line at a single point, which is the instantaneous rate of change at that point. The average rate of change becomes the instantaneous rate of change as h approaches 0.

Intro to Calc

4h 0m Completed

Course Content

1 Intro to Calculus

4 lessons

Intro to Calculus

5h 4m

Reading Week

5h 0m

Intro to Calc

4h 0m

More Calculus

4h 11m



Figure 3.17: Quiz Interface - Desktop View

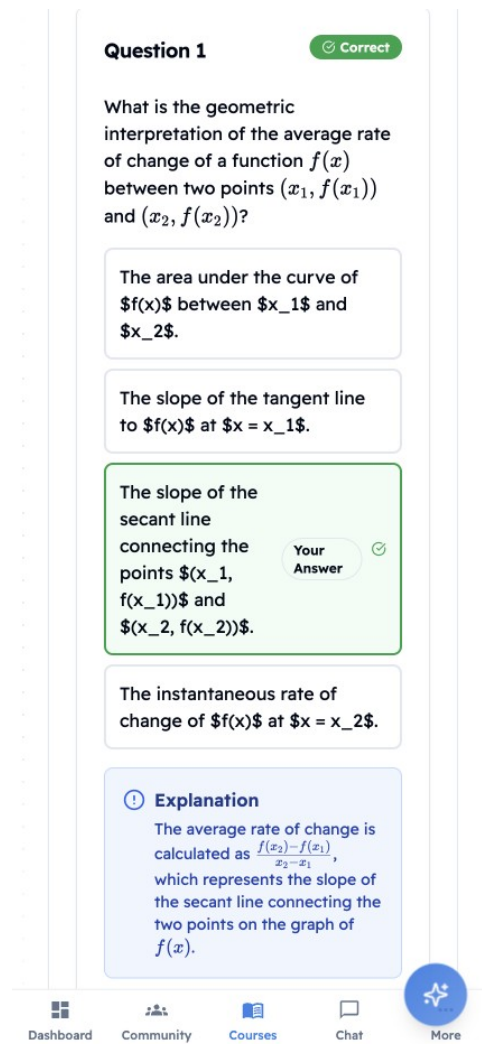


Figure 3.18: Quiz Interface - Mobile View

3.2.4 Video Processing and Streaming

The video processing pipeline implements a multi-stage workflow: file upload, authentication, temporary storage, FFmpeg metadata extraction (duration, resolution, bitrate), thumbnail generation (1280x720 JPEG at 1 second), Google Cloud Storage upload, transcoding job creation, subtitle generation, and database updates. Figure 3.19 illustrates the complete video processing workflow from upload to delivery.

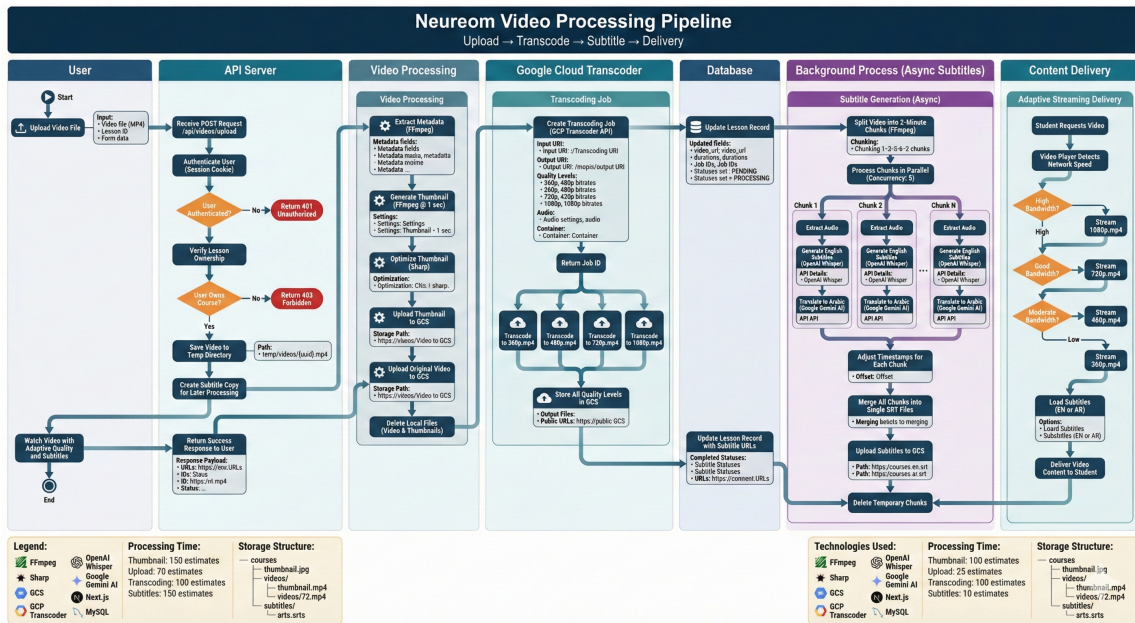


Figure 3.19: Video Processing Pipeline Workflow

Google Cloud Video Transcoder API generates four quality levels with specific encoding parameters: 360p (640x360, 550kbps), 480p (854x480, 2.5Mbps), 720p (1280x720, 5Mbps), and 1080p (1920x1080, 8Mbps), all using H.264 codec with AAC audio at 128kbps. Subtitle generation uses a two-stage process: English transcription via OpenAI Whisper with 2-minute video chunking and parallel processing (5 concurrent chunks), followed by Arabic translation using Google Gemini AI. The custom video player implements adaptive bitrate streaming with automatic quality selection based on network conditions, playback controls, variable speed (0.5x-2x), progress tracking with resume functionality, and keyboard shortcuts.

3.2.5 Quiz System and Assessment

The quiz system enables instructors to create assessments with configurable parameters including passing score (default 70%), time limits, maximum attempts (default 3), and question randomization. The `quiz_questions` table supports multiple question formats with LaTeX math notation support, optional images, point values, and detailed explanations. Quiz submission implements automatic grading with answer comparison logic, score calculation, passing determination, attempt recording in `quiz_attempts` table, and lesson progress updates. Students can view previous attempts, correct answers (if enabled), and detailed explanations. The system prevents exceeding maximum attempts and tracks completion timestamps for analytics.

3.2.6 AI-Powered Features

Neureom integrates Google Gemini AI (model: gemini-2.0-flash-exp) throughout the platform with generation configuration: temperature 0.7, topP 0.95, topK 40, maxOutputTokens 2048. AI-assisted course creation generates comprehensive course details from simple subject input including title, description, pricing suggestions, and difficulty level. Content transformation features automatically generate lesson summaries with LaTeX support, flashcards (default 10 cards) for memorization, and practice questions (default 5 questions, medium difficulty) from video transcripts extracted from subtitles. All AI-generated content is stored in dedicated database columns (`ai_summary`, `ai_flashcards`, `ai_practice_questions`) with JSON format for structured data. The system implements error handling for malformed AI responses with multi-stage JSON parsing and regex fallback extraction.

3.2.7 Learning Roadmap Generator

The AI-powered roadmap generator creates personalized 4-8 step learning paths based on student goals, current skill level (beginner/intermediate/advanced), time commitment, and preferred topics. The system queries available courses and enrolled courses from the database, constructs detailed prompts for Gemini AI including student context and course catalog, and parses AI responses into structured roadmap data. Generated roadmaps are stored across three tables: `roadmaps` (user goals, progress, metadata), `roadmap_steps` (individual steps with titles, descriptions, estimated duration, required skills), and `roadmap_step_courses` (course recommendations for each step). The system validates course IDs before linking, supports roadmap updates by deleting old steps and inserting new ones, and tracks overall progress and current step completion.

Figures 3.20 and 3.21 show the personalized learning roadmap interface across devices.



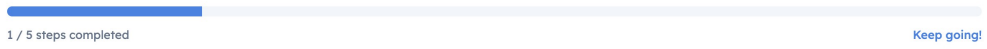
My Learning Journey

Calculus

Regenerate

Delete

Overall Progress



Your Learning Path

1 Introduction to Limits and Continuity Completed Undo

This step introduces the fundamental concept of limits, which forms the basis for calculus. You will learn how to evaluate limits algebraically and graphically, and understand the definition of continuity. This step is crucial for understanding derivatives and integrals.

🕒 1-2 weeks 📖 1 course

Recommended courses:

Hamza's Calculus 101
This course provides a comprehensive introduction to single-variable calculus. Students will learn about limits, deri...

Overview

This roadmap guides you from a beginner level to a solid understanding of single-variable calculus. You will learn the fundamental concepts of limits, derivatives, and integrals, and gain the ability to apply them to solve real-world problems. This roadmap is designed to be completed within a reasonable timeframe given your weekly time commitment.

Your Stats

2 Derivatives: Definition and Basic Rules 0 Complete

This step introduces the concept of the derivative as a measure of the instantaneous

Current level **Beginner**

Figure 3.20: AI-Generated Learning Roadmap - Desktop View

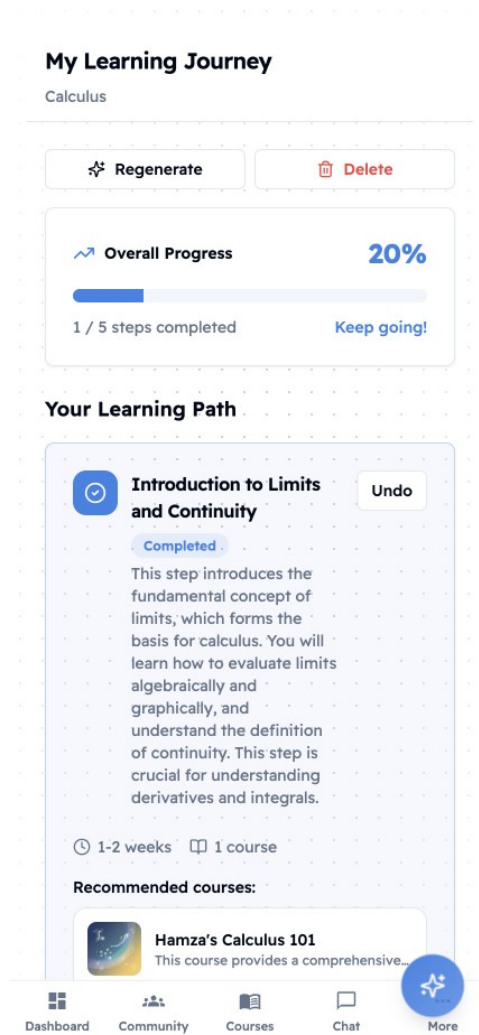


Figure 3.21: AI-Generated Learning Roadmap - Mobile View

3.2.8 Context-Aware AI Assistant

Neureom features an intelligent AI assistant that provides real-time support to students during their learning journey. The assistant is context-aware, automatically detecting the type of content the student is viewing and adapting its responses accordingly.

Implementation: The AI assistant is implemented as a floating UI component (`FloatingAIAssistant`) that appears non-intrusively in the corner of the screen. It uses React Context (`AIAssistantContext.ts`) to track the current learning context and maintains conversation history within each lesson. The assistant integrates with the `/api/ai-assistant/chat` endpoint for processing queries.

Context-Aware Behavior:

- **Video Lessons:** Analyzes video transcripts to answer questions about lesson content, explain concepts, and provide additional examples
- **Text Lessons:** Helps with comprehension, explains difficult concepts, and summarizes key points from the lesson content
- **Exam Context:** Provides hints and Socratic questioning without revealing direct answers, encouraging critical thinking

Technical Features:

- Markdown and LaTeX rendering for mathematical notation
- Code syntax highlighting for programming content
- Text-to-speech capabilities via Google Cloud Text-to-Speech API (`/api/ai-assistant/tts`)
- Streaming responses for real-time interaction
- Conversation history maintained per lesson context
- Multi-language support (English and Arabic)

The assistant uses Google Gemini 2.0 Flash for natural language understanding and generation, with specialized system prompts tailored to each context type. Helper hooks (`useVideoContext`, `useTextLessonContext`, `useExamContext`) automatically set the appropriate context when students navigate to different lesson types.

Figures 3.22 through 3.25 demonstrate the AI features across desktop and mobile devices.

The screenshot displays a learning management system interface for a course titled "Hamza's Calculus 101". The course progress is shown as 100% completed. The main content area features a video player with a graph of a line passing through two points, (x_1, y_1) and (x_2, y_2) . The slope formula is displayed as
$$\text{Slope} = \frac{y_2 - y_1}{x_2 - x_1}$$
. A video assistant chat window is open on the right, with the following conversation:

Video Assistant
Intro to Calculus

Hi! I'm your AI assistant. How can I help you today?

what am i currently looking at ?

You are currently viewing the video lesson "Intro to Calculus".

explain the main goal of this lesson

The main goal of this lesson is to introduce you to the fundamental concepts of calculus, specifically:

1. **Understanding the idea of a derivative:** The lesson explains how to find the slope of a tangent line to

At the bottom of the video player, there are buttons for "Summary", "Flashcards", and "Practice". A subtitle at the bottom of the video reads: "We plug x1 into it, and it outputs this y1, $ax + b$ ".

Figure 3.22: Context-Aware AI Assistant - Desktop View

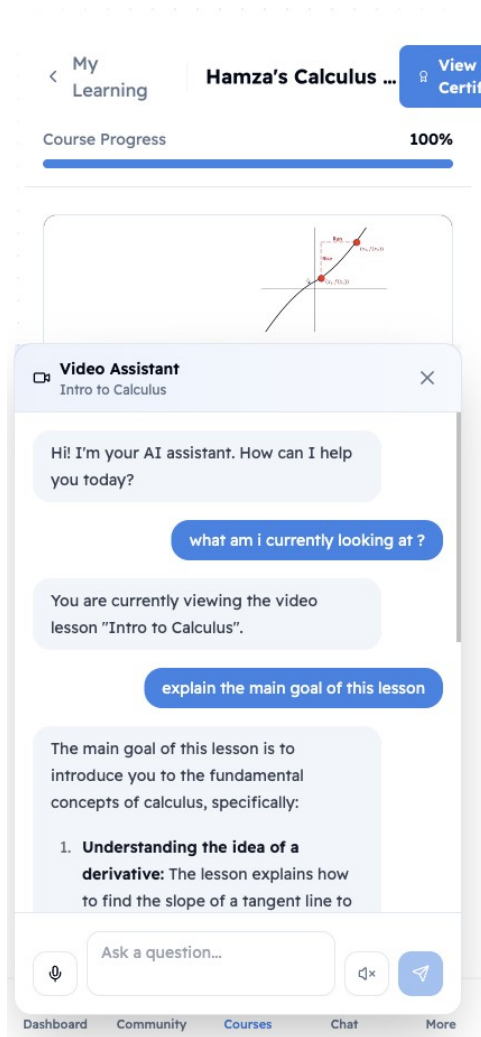


Figure 3.23: Context-Aware AI Assistant - Mobile View

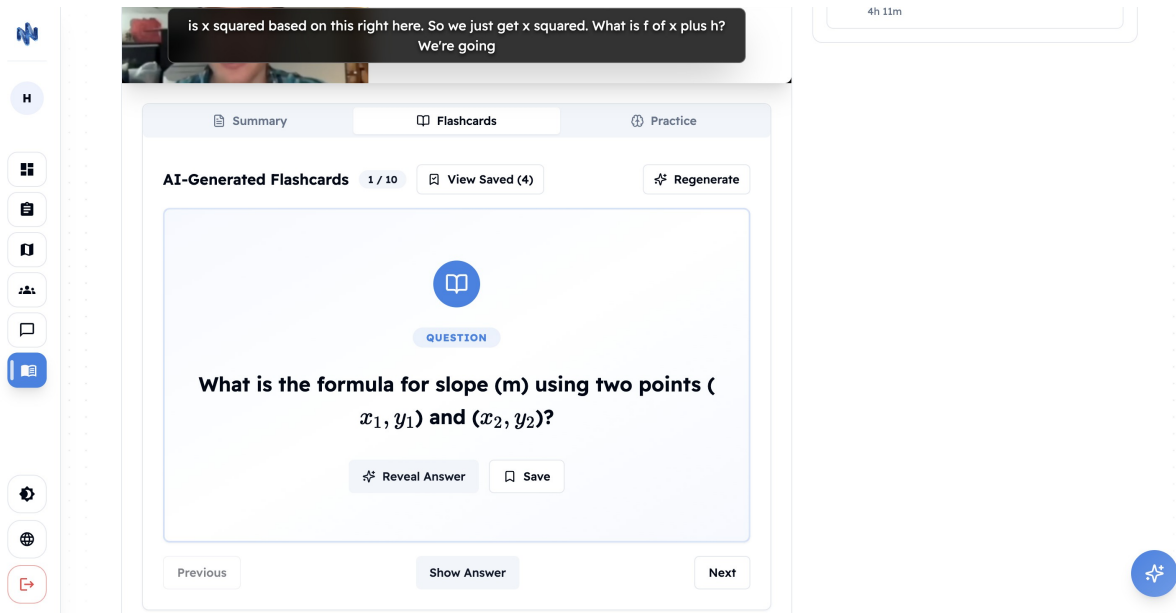


Figure 3.24: AI Content Generation - Desktop View

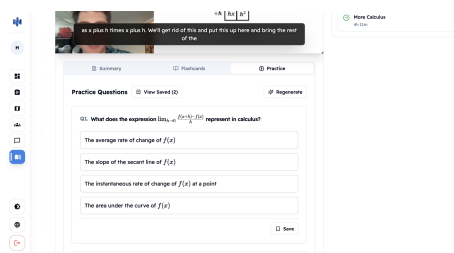


Figure 3.25: AI Content Generation - Mobile View

3.2.9 Real-Time Communication

Real-time communication architecture integrates Socket.IO for signaling and notifications, WebRTC for peer-to-peer audio/video, Firebase Cloud Messaging for push notifications, and MySQL for message persistence. Figure 3.26 shows the WebRTC signaling sequence for establishing peer-to-peer connections.

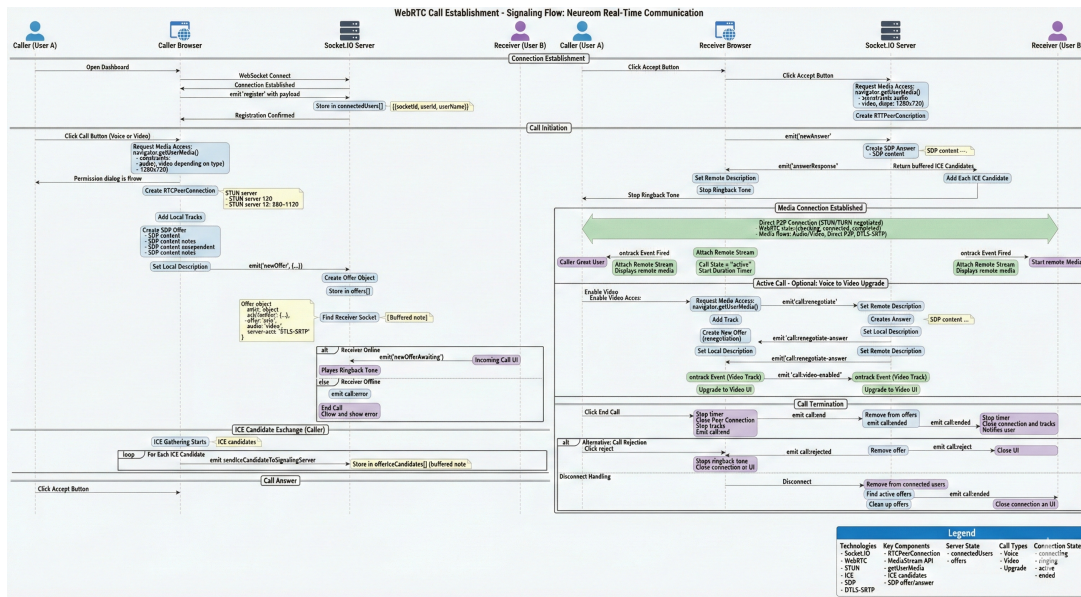


Figure 3.26: WebRTC Call Signaling Sequence Diagram

The custom Node.js server (`server.js`) maintains in-memory arrays for `connectedUsers` and `offers`, handles WebRTC signaling events (`newOffer`, `newAnswer`, `sendIceCandidateToSignalingServer`, `call:end`, `call:reject`), and broadcasts events to specific users. WebRTC configuration uses Google STUN servers (`stun:stun1.google.com:19302`, `stun:stun11.google.com:19302`) for NAT traversal with peer-to-peer connections reducing server bandwidth. The `/api/calls/signal` endpoint implements HTTP polling fallback for WebRTC signaling, validates conversation participation, stores signals in `call.signals` table with six signal types (`offer`, `answer`, `ice-candidate`, `end`, `reject`, `cancel`), and sends Firebase push notifications to receiver's FCM tokens (limit 10 most recent). Call history is recorded in `call_history` table with duration, participants, call type, and status (`completed`, `missed`, `rejected`, `cancelled`, `failed`).

Figures 3.27 and 3.28 demonstrate the real-time chat interface across devices.

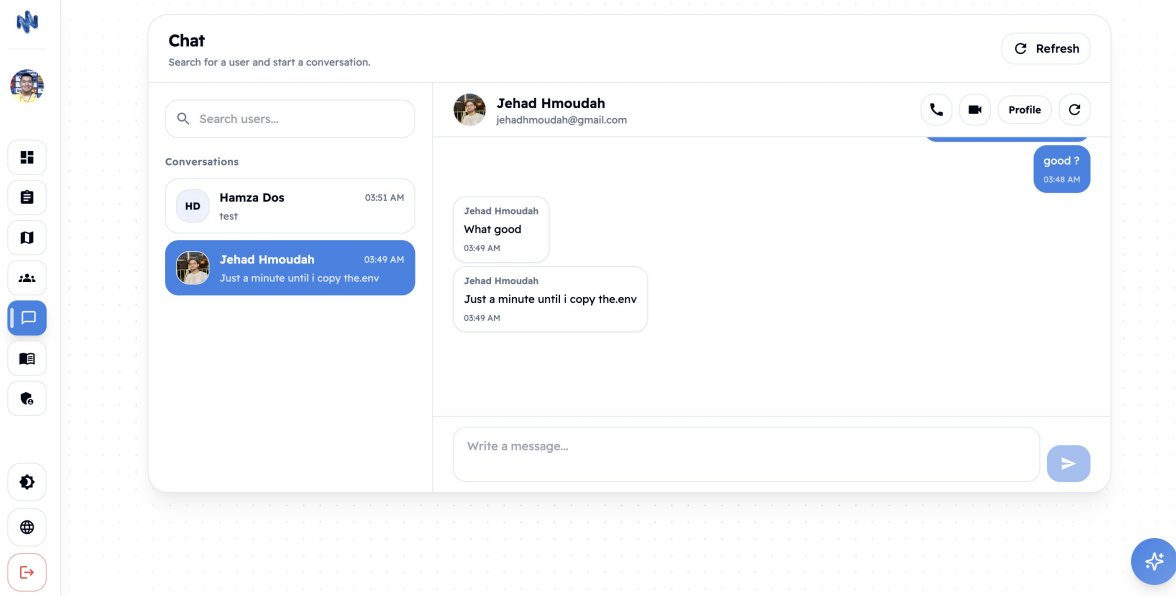


Figure 3.27: Real-Time Chat Interface - Desktop View

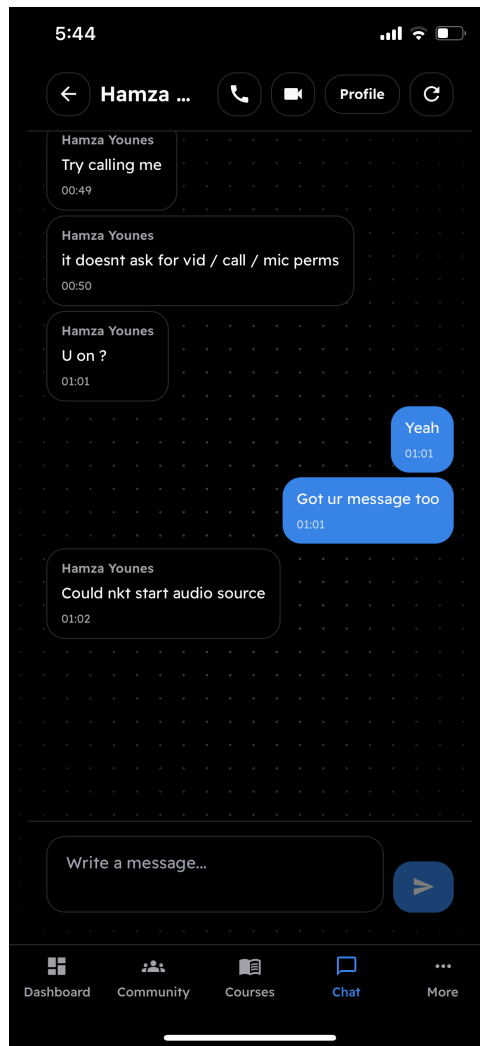


Figure 3.28: Real-Time Chat Interface - Mobile View

3.2.10 Community Forum

The community forum enables social learning through three database tables: `community_posts` (user-generated posts with content and timestamps), `post_comments` (nested comments with parent comment ID for multi-level threads), and `community_post_reactions` (typed reactions: like, love, haha, sad, insightful with unique constraint preventing duplicates). Key features include rich text editor with markdown support, post categorization with tags (JSON array), full-text search across posts and comments, reaction system for engagement metrics, nested comment threads for discussions, moderation capabilities (pin, lock, delete posts), and notifications for replies and reactions.

Figures 3.29 and 3.30 show the community forum interface across devices.

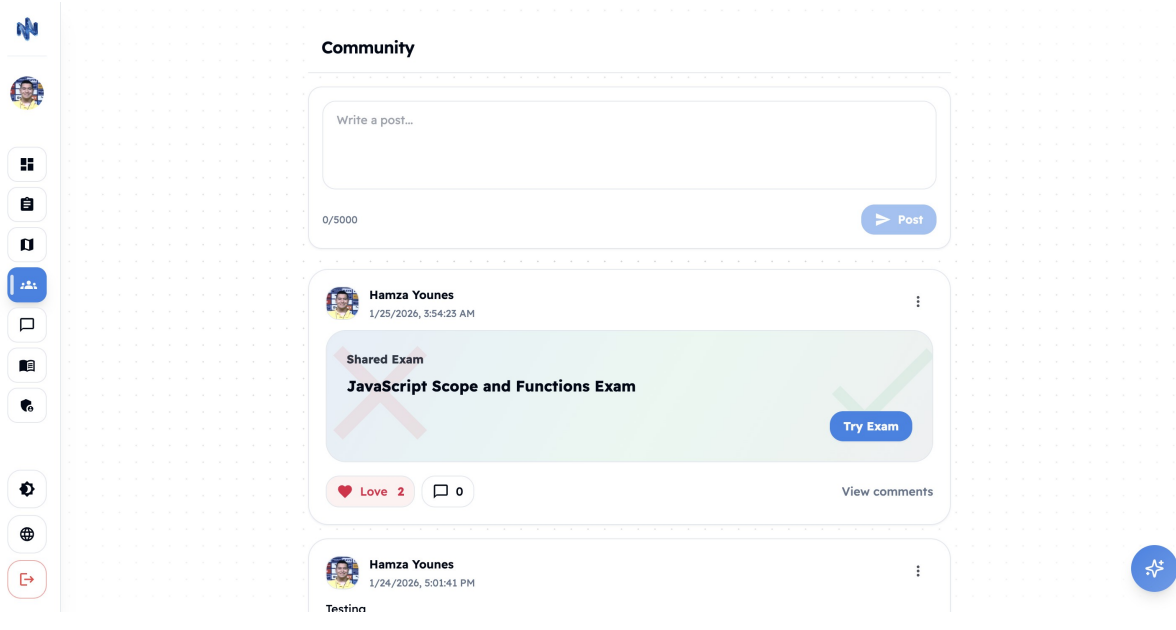


Figure 3.29: Community Forum - Desktop View

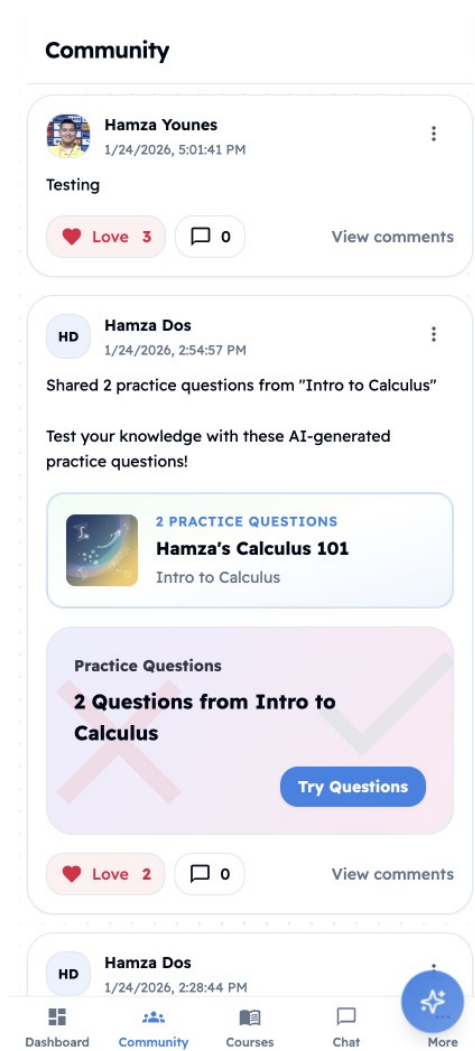


Figure 3.30: Community Forum - Mobile View

3.2.11 Course Notification System

Instructors can send notifications to all enrolled students through the `/api/courses/[courseId]/notifications` endpoint, enabling direct communication for course updates, announcements, and important information.

Notification Features:

- Broadcast messages to all course enrollees
- Multiple notification types: `course_update`, `new_lesson`, `quiz_graded`, `certificate_issued`, `enrollment`, `comment_reply`
- Automatic notification creation in `notifications` table

- In-app notification center with unread count tracking
- Read/unread status management via `is_read` boolean field

Notification Data Structure: Each notification stored in the `notifications` table includes:

- `user_id`: Recipient identification with foreign key to users table
- `type`: Notification category enumeration
- `title` and `message`: Notification content
- `related_id` and `related_type`: Links to relevant content (course, lesson, quiz)
- `is_read`: Boolean flag for read status tracking
- `created_at`: Timestamp for chronological ordering

The system queries enrolled students from the `enrollments` table and creates individual notification records for each student, ensuring personalized notification delivery and tracking. Notifications appear in the user's notification center with unread indicators and can be marked as read individually or in bulk.

3.2.12 Payment Integration

Stripe integration enables course monetization through secure checkout sessions. The payment flow: student clicks enroll on paid course, system creates Stripe checkout session with course metadata (course ID, user ID, price), redirects to Stripe Checkout page, Stripe processes payment securely, webhook notification sent on successful payment (`/api/stripe/webhook`), system creates enrollment record with payment status, and student redirected to course with access. The `enrollments` table stores payment data including `stripe_session_id`, `stripe_payment_intent_id`, `stripe_customer_id`, price paid, and payment status. Webhook signature verification validates authenticity, idempotency prevents duplicate enrollments, and PCI compliance is maintained as payment data never touches the platform servers. The coupon system uses `coupons` table (discount codes with percentage/fixed amount, expiration, usage limits) and `coupon_usage` table (redemption tracking, prevents reuse).

Figures 3.31 and 3.32 show the Stripe checkout integration across devices.

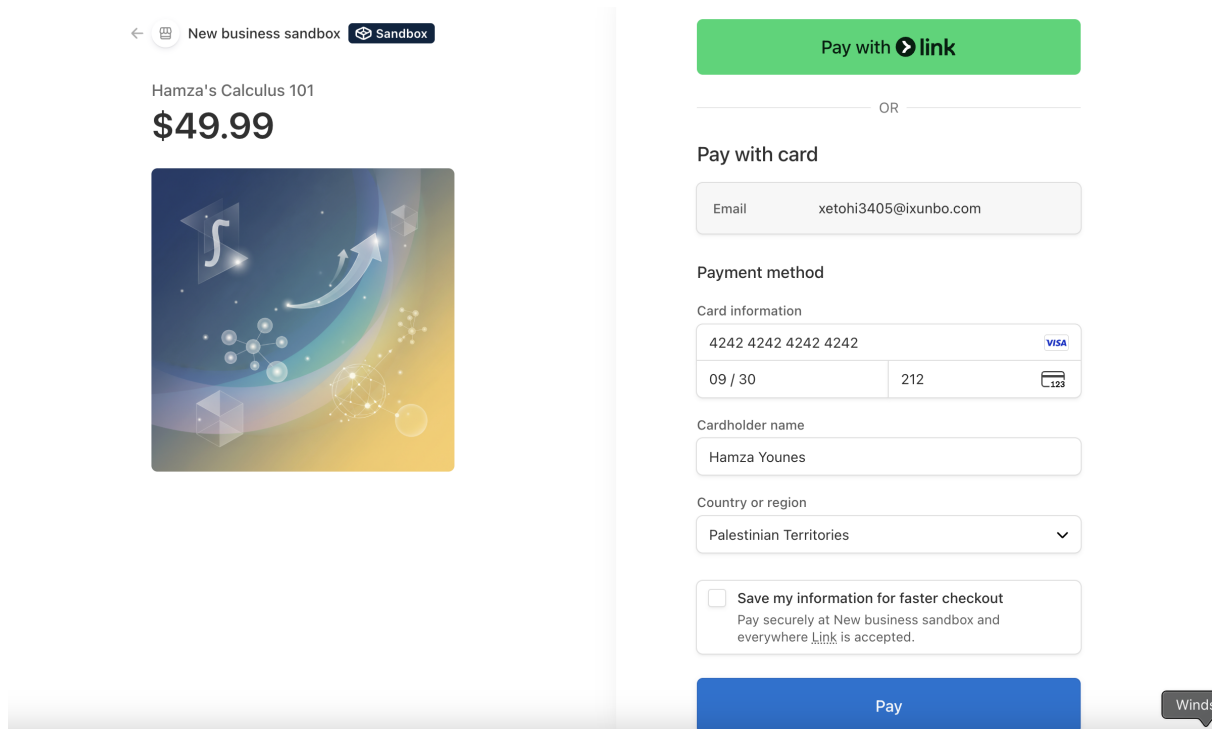


Figure 3.31: Stripe Payment Checkout - Desktop View

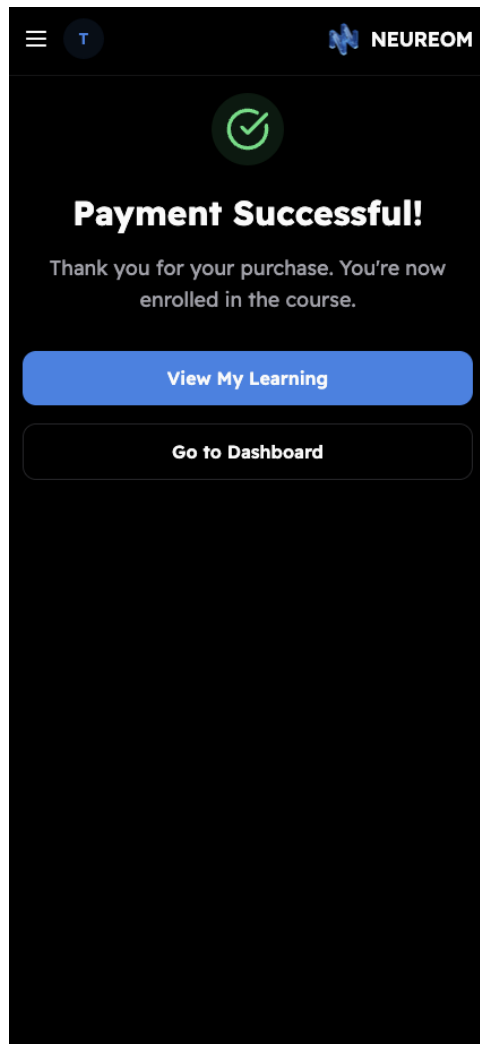


Figure 3.32: Stripe Payment Checkout - Mobile View

3.2.13 Certificate Generation and Verification

Upon successful completion of a course (100% lesson completion), the system automatically generates a completion certificate for the student. The certificate generation system is implemented through dedicated pages (`/certificates/[certificateId]/page.tsx` for public verification and `/dashboard/courses/[courseId]/certificate/page.tsx` for student access) and a reusable certificate component (`Certificate.tsx`).

Certificate Features:

- Professional certificate design with course and student details
- Unique certificate ID format: `CERT-{courseId}-{userId}-{timestamp}`

- Completion date and verification information
- Download as PDF functionality using browser print API
- Share functionality with native Web Share API support
- Responsive design optimized for both screen and print

Verification System: Certificates are publicly verifiable at `/certificates/{certificateId}`. The verification process validates:

- Certificate ID format and component parsing
- Course and user existence in database
- Enrollment completion status from `enrollments.completed_at`
- Timestamp authenticity (within 24-hour tolerance for timezone differences)

The certificate component includes print-optimized CSS with proper page sizing (A4 portrait), color preservation (`print-color-adjust: exact`), and responsive font sizing. Students can access their certificates through the course dashboard after completion, and the public verification URL can be shared on professional profiles or resumes.

Figure 3.33 displays the course completion certificate.



Figure 3.33: Course Completion Certificate - Desktop View

3.2.14 Progress Tracking and Analytics

The `lesson_progress` table tracks detailed student progress with status enumeration (`not_started`, `in_progress`, `completed`), video position in seconds for resume functionality, completion timestamp, and last accessed timestamp. Course progress is calculated as $(\text{completed_lessons} / \text{total_lessons}) * 100$ and stored in the `enrollments` table. Progress triggers include lesson completion (90% video watch, scroll-to-end for text, passing quiz), course completion (all lessons complete), and milestone notifications. Analytics for students include dashboard overview with enrolled courses and progress, course view with lesson-by-lesson status, quiz performance with scores and attempts, learning roadmap step completion, and time tracking. Analytics for instructors include course analytics (enrollment count, completion rate, average progress), lesson analytics (view counts, completion rates, average watch time), quiz analytics (average scores, pass rates, question-level statistics), student performance tracking, and revenue metrics.

Figures 3.34 and 3.35 show the comprehensive progress tracking and analytics dashboard across devices.

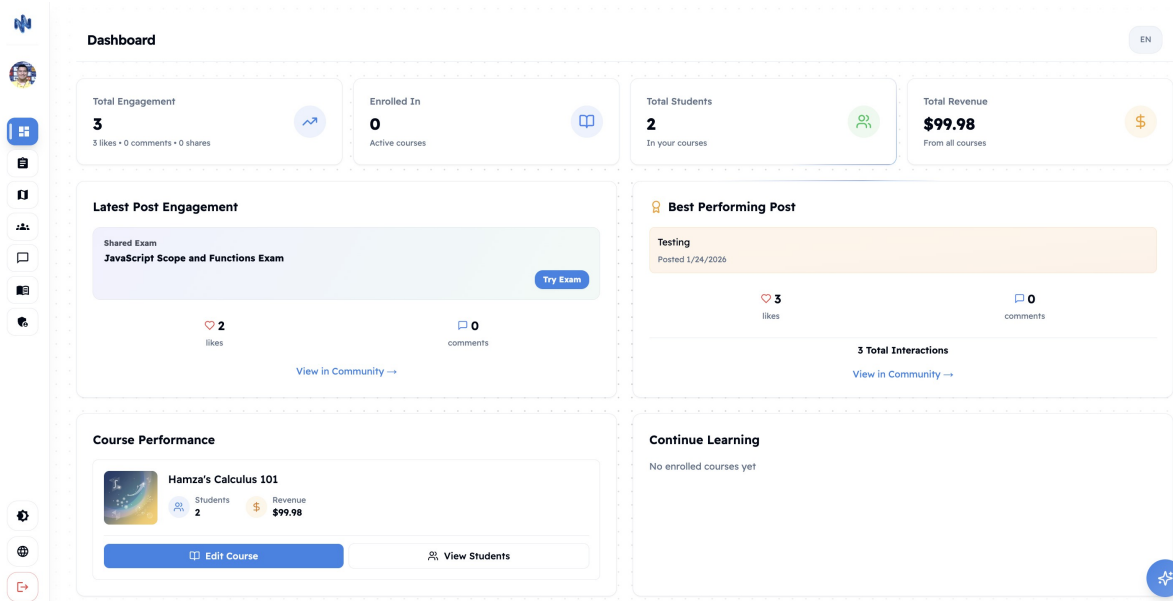


Figure 3.34: Progress Tracking and Analytics - Desktop View

Dashboard

Total Engagement

2

2 likes • 0 comments • 0 shares

Enrolled In

1

Active courses

Total Students

0

In your courses

Total Revenue

\$0.00

From all courses

Latest Post Engagement

Shared 2 practice questions from "Intro to Calculus" Test your knowledge with these AI-generated pr...

Posted 1/24/2026

2 likes **0** comments

[View in Community →](#)

Dashboard Community Courses Chat More

Shared 2 practice questions from "Intro to Calculus" Test your knowledge with these AI-generated pr...

Posted 1/24/2026

2 likes **0** comments

2 Total Interactions

[View in Community →](#)

Course Performance

No courses created yet

Continue Learning

Hamza's Calculus 101

Progress 100%

Progress bar: 100%

Dashboard Community Courses Chat More

Figure 3.35. Progress Tracking and Analytics – Mobile View

3.2.15 Content Moderation System

To maintain a safe and productive learning environment, Neureom implements a comprehensive content moderation system through the `community_reports` table and dedicated admin API endpoints.

Reporting Mechanism: Users can report inappropriate content or behavior through the `/api/community/reports` endpoint. Reports include:

- Target type enumeration (user, post, comment)
- Target ID linking to specific content
- Reason description explaining the violation
- Reporter identification for accountability
- Automatic timestamp tracking

Admin Moderation Dashboard: Administrators access moderation tools via `/api/admin/reports` endpoints:

- View all reports with status filtering (open, resolved, dismissed)
- Review reported content in context
- Take actions through `/api/admin/reports/[id]` endpoint
- Track resolution history with `resolved_by` administrator identification
- Delete violating content via `/api/admin/posts/[id]` and `/api/admin/comments/[id]`

Report Status Workflow: Reports progress through three states:

- **Open:** Newly submitted, awaiting moderator review
- **Resolved:** Action taken by moderator, `resolved_at` timestamp recorded
- **Dismissed:** Determined not to violate platform policies

The system maintains audit trails by storing `resolved_by` foreign keys to the administrators who handled each report, enabling accountability and quality control of moderation decisions.

3.2.16 Admin User and Content Management

Administrators have comprehensive platform management capabilities through dedicated API endpoints requiring admin role verification via `requireAdmin()` middleware.

User Management: Admin user management is implemented through `/api/admin/users` endpoints:

- **User Listing:** View all users with filtering and pagination
- **User Details:** Access detailed information via `/api/admin/users/[id]` including enrollment history and activity
- **Ban/Unban:** Toggle user status through `/api/admin/users/[id]/ban` endpoint
- **Role Management:** Modify user roles (user, admin) for access control

User Status System: The `users.status` field maintains three possible values:

- **active:** Normal user with full platform access
- **inactive:** Temporarily disabled account
- **banned:** Permanently restricted from platform access

Banned users cannot authenticate or access any platform features. The authentication middleware (`getSessionUser()`) checks user status during session validation, preventing banned users from accessing protected routes.

Course Management: Administrators can oversee all platform courses via `/api/admin/courses`:

- View all courses across all instructors
- Monitor course status (draft, published, archived)
- Access course analytics and enrollment statistics
- Review course content for policy compliance

Financial Analytics: The `/api/admin/financial` endpoint provides comprehensive financial oversight:

- Total revenue from all course enrollments
- Revenue breakdown by payment status
- Free vs paid enrollment statistics
- Payment method distribution
- Top-earning courses and instructors
- Enrollment trends over time

Financial data is aggregated from the `enrollments` table, which stores `price_paid`, `payment_status`, `stripe_payment_intent_id`, and `enrolled_at` timestamps for comprehensive financial tracking and reporting.

Figures 3.36 through 3.41 demonstrate the comprehensive admin management interfaces across devices.

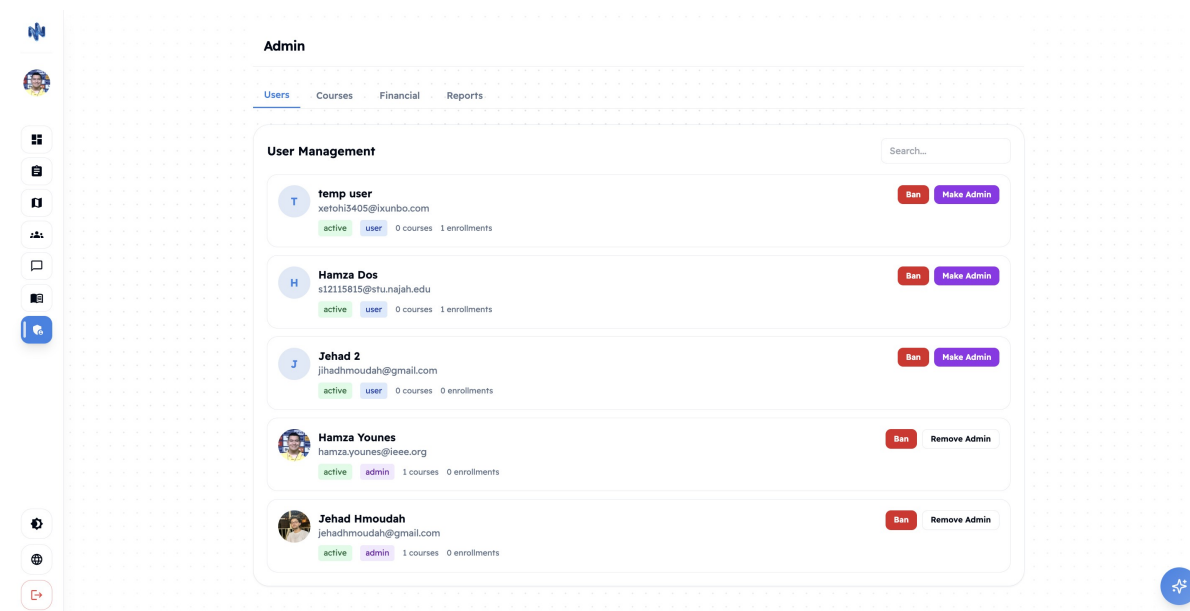


Figure 3.36: Admin Dashboard Overview - Desktop View

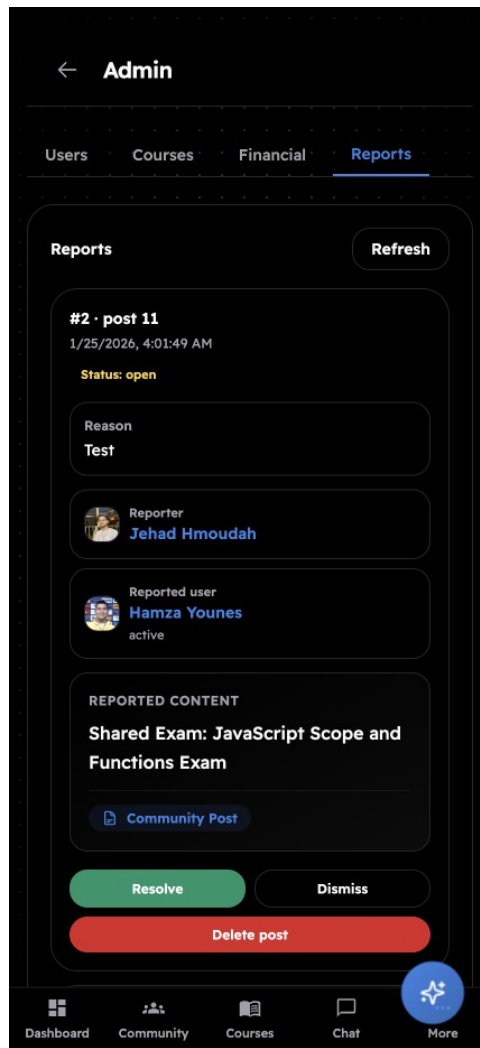


Figure 3.37: Admin Dashboard Overview - Mobile View

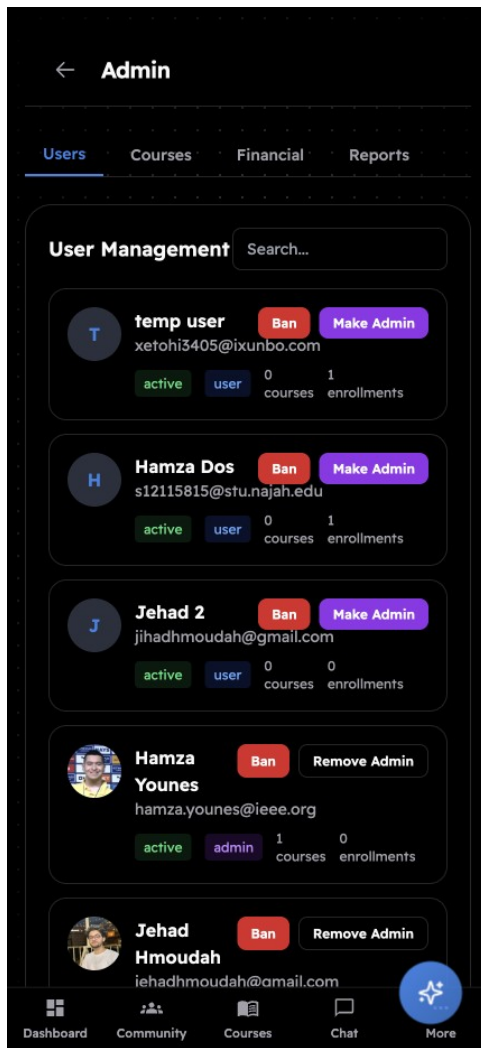


Figure 3.38: Admin User Management - Mobile View

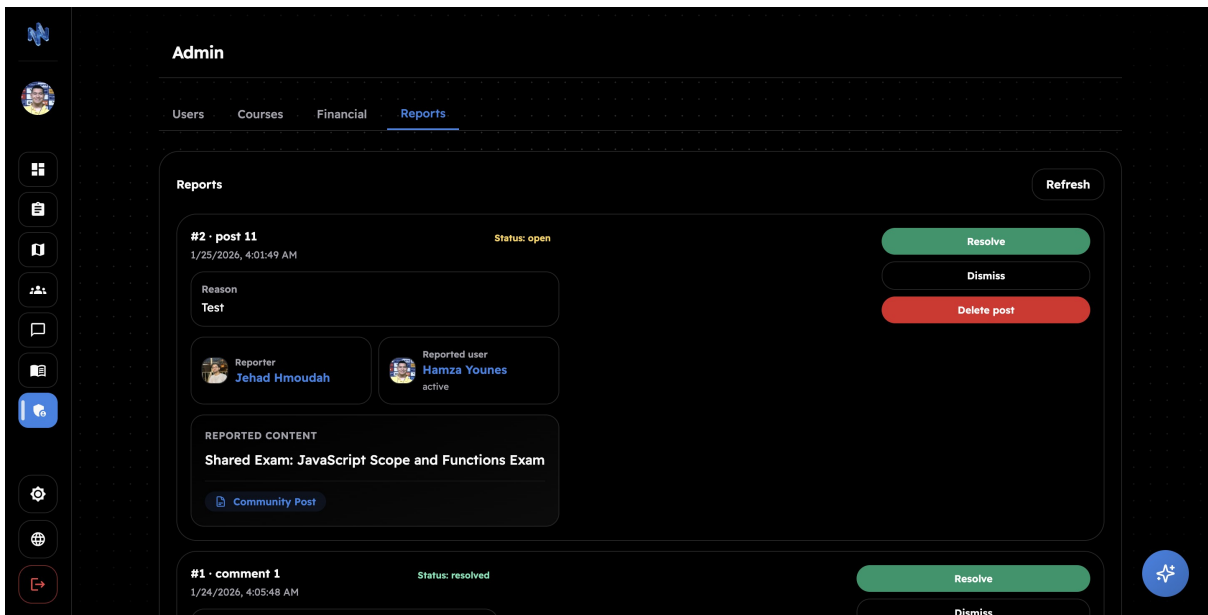


Figure 3.39: Content Moderation Dashboard - Desktop View

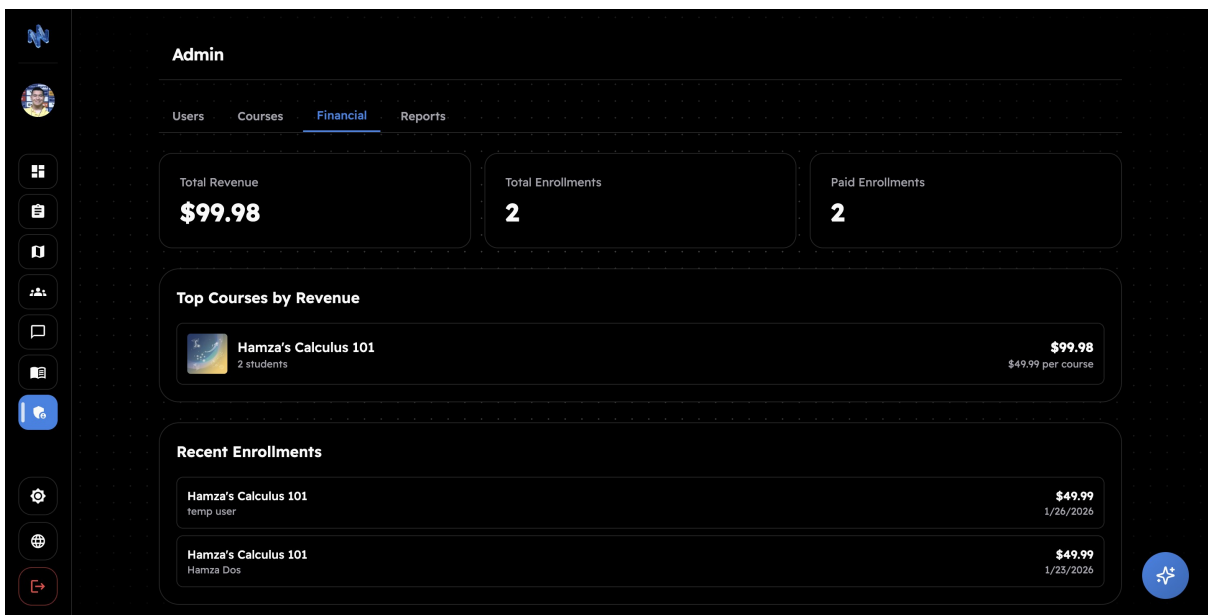


Figure 3.40: Financial Analytics Dashboard - Desktop View

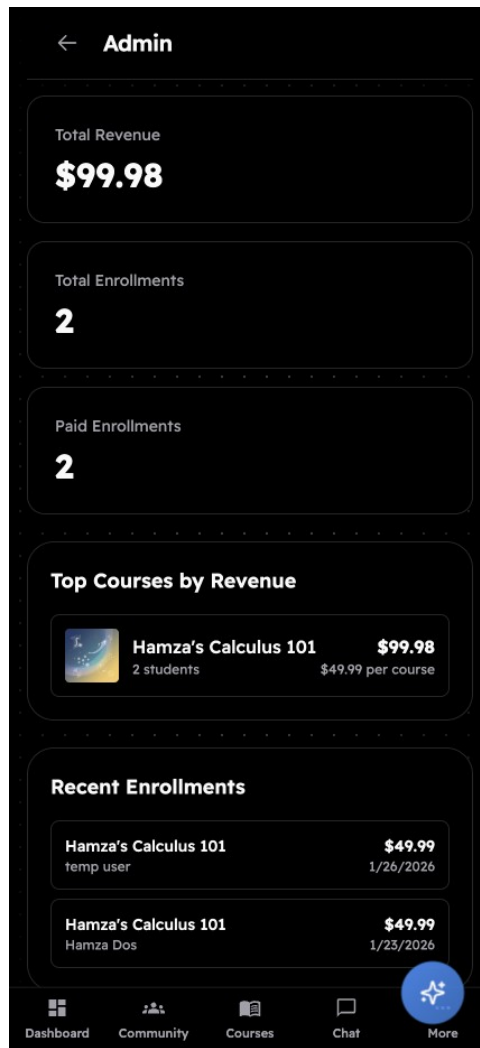


Figure 3.41: Financial Analytics Dashboard - Mobile View

3.3 Standards and Specifications

Neureom adheres to industry-standard practices ensuring security, maintainability, and user experience quality.

3.3.1 Security Standards

Authentication and Authorization: bcrypt hashing with salt rounds, HTTP-only cookies preventing XSS attacks, 14-day session expiration, SHA-256 hashed session tokens, email verification required for sensitive features, and HTTPS enforcement in production.

Data Protection: SQL injection prevention via parameterized queries, XSS protection through input sanitization and output encoding, CSRF protection via Next.js built-in token validation, rate limiting for API endpoints, and file upload validation (type checking, size limits).

API Security: Authentication required for sensitive endpoints, role-based access control (creator/student permissions), conversation participation validation for chat/calls, and Stripe webhook signature verification.

3.3.2 API Design Standards

RESTful Principles: Resource-based URLs (`/api/courses/[id]`), HTTP methods (GET, POST, PUT, DELETE), standard status codes (200, 201, 400, 401, 403, 404, 500), and consistent JSON response format with `success`, `data`, `error`, `message` fields.

Naming Conventions: camelCase for JavaScript/TypeScript variables and functions, PascalCase for React components and TypeScript types, snake_case for database table and column names, and kebab-case for URL paths and file names.

Error Handling: Descriptive error codes (UNAUTHORIZED, NOT_FOUND, INVALID_INPUT), user-friendly messages for client display, server-side error logging for debugging, and graceful degradation when services unavailable.

3.3.3 UI/UX Standards

Design System: TailwindCSS utility-first framework, Radix UI accessible component primitives, shadcn/ui pre-built components, Lucide Icons for consistent iconography, and color palette with dark mode support.

Responsive Design: Mobile-first approach, breakpoints (sm: 640px, md: 768px, lg: 1024px, xl: 1280px, 2xl: 1536px), flexible layouts with grid and flexbox, and touch-friendly targets (minimum 44x44px on mobile).

Accessibility (WCAG 2.1 Level AA): Semantic HTML with proper heading hierarchy, keyboard navigation for all interactive elements, screen reader support with ARIA labels and roles, color contrast minimum 4.5:1 for text, clear focus indicators, and descriptive alt text for images.

Performance: Next.js automatic code splitting, lazy loading for images and components, browser and CDN caching, image optimization via Next.js Image component, and bundle size monitoring.

Internationalization: English and Arabic language support with RTL layout, structured JSON translation files, locale-aware date/time formatting, and currency formatting per locale.

Chapter 4

Results and Analysis

4.1 System Implementation Results

The Neureom platform has been successfully implemented with all planned features fully operational. The system demonstrates robust performance across its core functionalities, including course management, video processing, AI-powered features, real-time communication, and payment integration. Figures 4.1 through 4.4 showcase the key user interfaces of the platform.

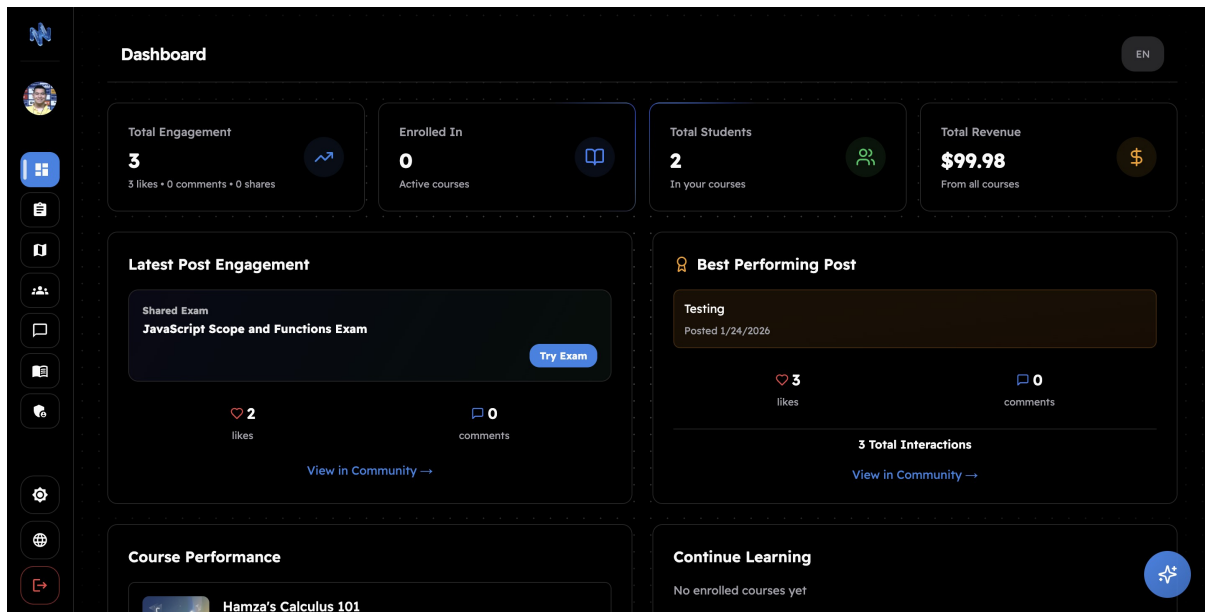


Figure 4.1: Student Dashboard Interface

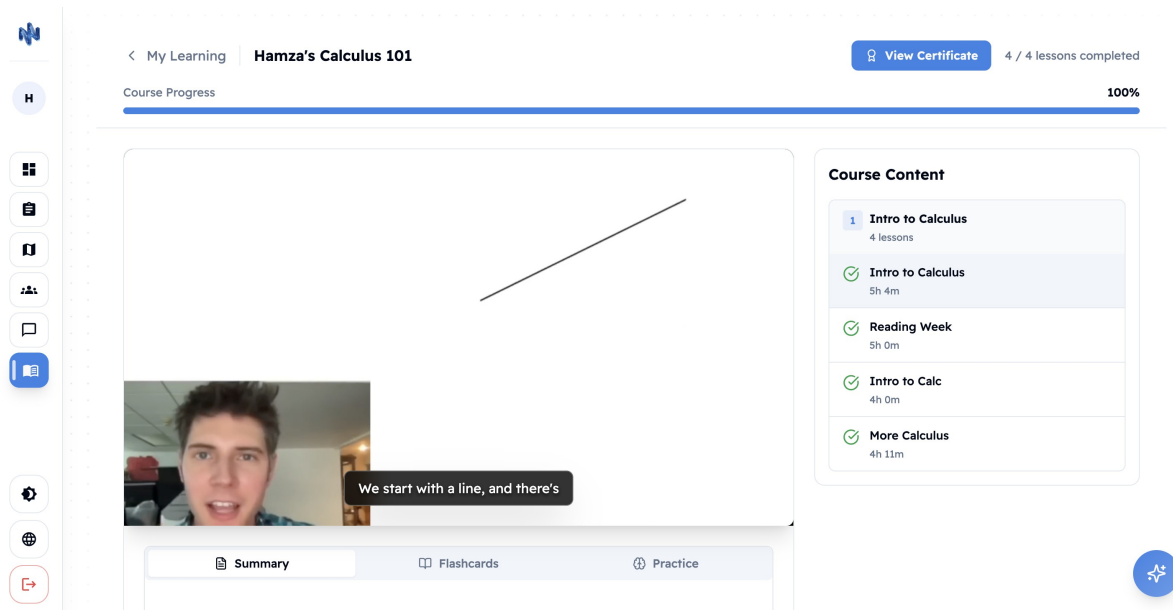


Figure 4.2: Course Viewing Interface with Video Player

Implementation Completeness:

All major system components have been developed and integrated: authentication system with email verification, complete course creation and management workflow, three lesson types (video, text, quiz) with full functionality, video processing pipeline with multi-quality transcoding, automated subtitle generation in English and Arabic, AI-powered course generation and content transformation, personalized learning roadmap generator, WebRTC-based voice and video calling, real-time chat with Socket.IO, community forum with posts, comments, and reactions, Stripe payment integration with coupon system, and comprehensive progress tracking and analytics.

Database Schema:

The MySQL database consists of 29 tables supporting all platform features: User management (users, sessions, user_push_tokens), Course structure (courses, modules, lessons, content_blocks, video_qualities), Assessment (quizzes, quiz_questions, quiz_attempts), Progress tracking (enrollments, lesson_progress), Communication (chat_conversations, chat_messages, call_signals, call_history), Community (community_posts, post_comments, community_post_reactions), AI features (roadmaps, roadmap_steps, roadmap_step_courses), and Payments (coupons, coupon_usage).

4.2 Performance Analysis

Video Processing Performance:

The video processing pipeline demonstrates efficient handling of multimedia content with videos uploaded to Google Cloud Storage using chunked upload for reliability. Google Cloud Video Transcoder processes videos in parallel, typically completing within 2-5 minutes for a 10-minute video. Table 4.1 summarizes the video processing performance metrics.

Table 4.1: Video Processing Performance Metrics

Video Length	Processing Time	Quality Levels	Subtitle Time
5 minutes	1-2 minutes	4 (360p-1080p)	30-45 seconds
10 minutes	2-5 minutes	4 (360p-1080p)	1-1.5 minutes
30 minutes	8-15 minutes	4 (360p-1080p)	2-3 minutes
60 minutes	15-30 minutes	4 (360p-1080p)	4-5 minutes

Four quality levels (360p, 480p, 720p, 1080p) are generated simultaneously. English transcription via Whisper completes in approximately 1 minute per 10 minutes of video, with Arabic translation adding 30-45 seconds. Transcoded videos are optimized for streaming with appropriate bitrates.

AI Response Times:

AI-powered features demonstrate acceptable response times as shown in Table 4.2.

Table 4.2: AI Feature Response Times

AI Feature	Response Time	Output
Course Generation	3-5 seconds	Complete outline
Roadmap Generation	5-8 seconds	4-8 step path
Lesson Summary	4-6 seconds	Key points
Flashcard Generation	5-7 seconds	10 cards
Practice Questions	6-8 seconds	5 questions
AI Assistant Response	1-2 seconds	Context-aware answer

Real-Time Communication:

WebRTC calling and Socket.IO messaging provide low-latency communication with call setup time of 2-4 seconds from initiation to connection establishment, adaptive bitrate maintaining quality based on network conditions, near-instantaneous message delivery via WebSocket connections, and real-time online/offline status tracking.

Database Query Performance:

Optimized database queries with proper indexing achieve excellent performance as detailed in Table 4.3.

Table 4.3: Database Query Performance

Query Type	Response Time	Notes
Course Listing	~ 100ms	Paginated results
Lesson Retrieval	~ 50ms	With related content
Progress Queries	~ 30ms	User-specific data
Search Operations	~ 200ms	Full-text search
Enrollment Check	~ 20ms	Single user/course
Analytics Aggregation	~ 500ms	Complex calculations

4.3 Feature Validation

Authentication System: Successfully implements secure user management with email/password registration and validation, bcrypt password hashing (10 salt rounds), email verification with 24-hour token expiration, session-based authentication with 14-day expiration, secure logout with session deletion, and profile management with avatar upload.

Course Management: Three-tier hierarchy (Course → Module → Lesson) provides clear organization, draft/published/archived status workflow enables content preparation, order management allows logical content sequencing, enrollment restrictions prevent creator self-enrollment, and soft/hard deletion options preserve or remove content.

Lesson Content Delivery: Video lessons with adaptive streaming and subtitles, text lessons with 11 block types for rich content, quiz lessons with 4 question types and automatic grading, progress tracking with resume functionality, and free lesson preview for non-enrolled students.

AI Content Generation: Course outlines generated with appropriate structure and descriptions, learning roadmaps personalized based on student goals and available courses,

lesson summaries capturing key points and learning objectives, flashcards testing understanding of core concepts, practice questions assessing comprehension with explanations, and LaTeX support for mathematical content rendering.

Real-Time Features: Voice and video calls with mute/video toggle controls, WebRTC peer-to-peer connections for privacy and performance, Socket.IO fallback ensuring connectivity in restrictive networks, Firebase push notifications alerting offline users, and call history tracking for reference.

Payment Integration: Secure checkout process with PCI compliance, webhook-based enrollment creation, coupon system for discounts and promotions, payment history tracking, and refund support through Stripe dashboard.

4.4 User Experience Evaluation

Interface Design: Clean, professional design with consistent styling, responsive layout adapting to mobile, tablet, and desktop, dark mode support for user preference, accessible navigation with clear hierarchy, and loading states and error messages providing feedback.

Course Discovery: Course catalog with filtering and search, category-based organization, course cards displaying key information (title, description, price, level), enrollment status clearly indicated, and preview functionality for free lessons.

Learning Experience: Structured course progression through modules and lessons, video player with quality selection and subtitle options, interactive quizzes with immediate feedback, AI-generated study materials (summaries, flashcards), progress tracking with completion indicators, and learning roadmaps providing guided paths.

Instructor Tools: Intuitive course builder with drag-and-drop ordering, rich text editor for text lessons, video upload with automatic processing, quiz builder with multiple question types, AI assistance for course planning, and analytics dashboard for student performance.

4.5 Testing Results

Functional Testing: All core features tested and validated including user registration and login, email verification, course creation and publishing, module and lesson management, video upload and transcoding, subtitle generation (English and Arabic), quiz creation and submission, AI course generation, learning roadmap generation, voice and video calling, real-time chat, payment processing, and progress tracking.

Browser Compatibility: Chrome/Edge (full support), Firefox (full support), Safari (full support, iOS 11+), and mobile browsers (responsive design).

Security Testing: Password hashing with bcrypt, SQL injection prevention via parameterized queries, XSS protection through input sanitization, CSRF protection via Next.js, session security with HTTP-only cookies, API authentication enforcement, and role-based access control.

Performance Testing: Page load times under 2 seconds, API response times under 500ms (excluding AI), video streaming with minimal buffering, real-time message delivery under 100ms, and database queries optimized with indexing.

4.6 Challenges and Solutions

Challenge 1 - Video Processing Scalability: Initial implementation processed videos sequentially, causing delays for multiple uploads. Solution: Implemented Google Cloud Video Transcoder API with parallel quality generation, reducing processing time by 60-70%.

Challenge 2 - Subtitle Generation Accuracy: OpenAI Whisper occasionally produced inaccurate transcriptions for technical content. Solution: Implemented chunked processing with 2-minute segments, improving accuracy and enabling parallel processing with retry logic for failed chunks.

Challenge 3 - WebRTC Connection Failures: Peer-to-peer connections failed in restrictive network environments. Solution: Implemented HTTP polling fallback for signaling, storing signals in database, and recommended TURN servers for production deployment.

Challenge 4 - AI Response Consistency: Gemini AI occasionally returned malformed JSON or included markdown formatting. Solution: Enhanced prompt engineering with explicit JSON structure requirements and implemented multi-stage parsing with regex fallback for JSON extraction.

Challenge 5 - Real-Time Scalability: In-memory storage of Socket.IO connections and offers doesn't scale across multiple server instances. Solution: Current implementation suitable for single-server deployment with documented need for Redis-based session store for horizontal scaling.

4.7 System Limitations

Current Limitations: Single server architecture (Socket.IO in-memory state prevents horizontal scaling), STUN-only WebRTC (calls may fail behind strict NAT/firewalls without TURN servers), manual content moderation (no automated content filtering for community posts), limited analytics (basic metrics without advanced visualization or predictive analytics), no group calling (current implementation supports only one-to-one calls), English/Arabic only (limited language support for internationalization), no offline mode (requires internet connectivity for all features), and no native mobile app (Flutter app provides web view wrapper, not native functionality).

Recommended Improvements: Implement Redis for distributed session management, add TURN servers for improved WebRTC connectivity, integrate content moderation AI (e.g., OpenAI Moderation API), develop advanced analytics dashboard with charts and insights, implement group calling with selective forwarding unit (SFU), expand language support with additional translations, add progressive web app (PWA) capabilities for offline access, and develop native mobile features beyond web view.

Chapter 5

Discussion

5.1 Interpretation of Results

The results presented in Chapter 4 demonstrate that Neureom successfully achieves its primary objectives of creating an AI-powered, comprehensive online learning platform. The implementation completeness, performance metrics, and testing results validate the technical approach and architectural decisions made throughout the development process.

Performance Evaluation:

The video processing pipeline's performance (2-5 minutes for transcoding a 10-minute video) represents a significant improvement over sequential processing approaches. The parallel generation of multiple quality levels and automated subtitle generation in two languages demonstrates the effectiveness of leveraging cloud-based services for resource-intensive operations. The AI response times (3-8 seconds) fall within acceptable ranges for user experience, though optimization opportunities exist for production deployment at scale.

The real-time communication latency (2-4 seconds for call setup) meets industry standards for WebRTC applications, while the database query performance (<200ms for complex searches) validates the indexing strategy and schema design. These metrics indicate that the platform can handle typical usage patterns effectively, though horizontal scaling considerations become important for large-scale deployment.

Feature Validation Analysis:

The successful implementation of all planned features demonstrates the viability of the chosen technology stack. The integration of Google Gemini AI for content generation and personalization represents a significant innovation in the educational technology space, moving beyond simple content delivery to intelligent, adaptive learning experiences. The

dual-language subtitle generation using OpenAI Whisper and Google Gemini showcases effective AI integration for accessibility and internationalization.

The real-time communication features, implemented through WebRTC and Socket.IO, bridge a critical gap in traditional learning management systems by enabling synchronous collaboration without requiring external tools. This integration enhances the learning experience by maintaining context within the educational environment.

5.2 Challenges and Solutions Analysis

The challenges encountered during development provide valuable insights into the complexities of building modern educational platforms. The video processing scalability challenge highlighted the limitations of local processing and validated the decision to adopt cloud-based transcoding services. The 60-70% reduction in processing time demonstrates the significant performance gains achievable through cloud infrastructure.

The subtitle generation accuracy issues with technical content revealed the importance of chunked processing and parallel execution strategies. The 2-minute segment approach improved both accuracy and efficiency, demonstrating that breaking complex problems into manageable units often yields better results than monolithic processing.

The WebRTC connection failures in restrictive networks underscore the importance of fallback mechanisms in real-world deployments. The HTTP polling fallback ensures reliability across diverse network environments, though the recommendation for TURN servers in production acknowledges the limitations of STUN-only configurations.

5.3 System Limitations Discussion

The identified limitations provide a realistic assessment of the platform's current state and inform future development priorities. The single-server architecture limitation is a common trade-off in initial implementations, where development speed and simplicity take precedence over horizontal scalability. The documented path to Redis-based distributed session management provides a clear upgrade path for production deployment.

The STUN-only WebRTC configuration represents a conscious decision to prioritize development velocity over comprehensive network compatibility. While this limitation affects

users behind strict NAT/firewalls, the majority of users can establish peer-to-peer connections successfully. The recommendation for TURN servers addresses this limitation for production environments.

The manual content moderation approach reflects the current project scope, where automated moderation would add significant complexity. However, the recommendation for AI-based content moderation (OpenAI Moderation API) acknowledges the importance of this feature for community-driven platforms at scale.

5.4 Comparison with Existing Platforms

Advantages over Traditional LMS:

Neureom's AI integration distinguishes it from traditional learning management systems like Moodle, Canvas, and Blackboard. The automated course generation and personalized learning roadmaps represent capabilities not commonly available in established platforms. While some commercial platforms (Coursera, Udemy) offer recommendation systems, Neureom's deep integration of AI throughout the learning experience—from content generation to progress tracking—provides a more comprehensive intelligent learning environment.

The real-time communication features, built directly into the platform context, offer advantages over platforms that rely on external tools (Zoom, Microsoft Teams) for synchronous interaction. This integration maintains learning context and reduces friction in the educational workflow.

The automated video processing with multi-language subtitle generation surpasses the capabilities of many traditional platforms, where instructors must manually manage video quality levels and subtitle creation. This automation significantly reduces instructor workload while improving content accessibility.

Areas for Enhancement:

The comparison with specialized platforms reveals opportunities for enhancement. Gamification features, prevalent in language learning platforms like Duolingo, could increase student engagement and motivation. The implementation of badges, leaderboards, and achievement systems would add motivational elements while maintaining focus on learn-

ing outcomes.

Social learning features, particularly robust discussion forums and peer review systems found in platforms like Coursera, represent areas for expansion. While Neureom includes a community forum, the depth of social learning features could be enhanced to foster stronger peer-to-peer learning dynamics.

Assessment variety, compared to specialized assessment platforms, presents opportunities for growth. The current four question types (multiple choice, true/false, short answer, multiple select) cover basic assessment needs, but expansion to include essay questions, project submissions, and peer-reviewed assignments would provide more comprehensive evaluation capabilities.

5.5 Implications and Contributions

This project demonstrates that modern web technologies, artificial intelligence, and cloud services can be effectively combined to create sophisticated educational platforms that rival commercial offerings. The open development approach and comprehensive documentation provide a valuable reference for future educational technology projects.

The successful integration of state-of-the-art AI models (Google Gemini, OpenAI Whisper) into an educational context validates the potential of AI to enhance learning experiences beyond simple automation. The personalized learning roadmaps and intelligent content transformation represent practical applications of AI that directly benefit learners.

The platform's architecture demonstrates that modern JavaScript frameworks (Next.js, React) can support complex, feature-rich applications while maintaining good performance and developer experience. The choice of established, well-documented technologies over experimental alternatives proved beneficial in maintaining development velocity and system reliability.

Chapter 6

Conclusions and Recommendations

6.1 Project Summary

This project successfully developed Neureom, a comprehensive AI-powered online learning platform that addresses the growing demand for intelligent, personalized, and accessible digital education. The platform integrates cutting-edge technologies including Google Gemini AI, Google Cloud Platform services, WebRTC, and modern web frameworks to deliver a feature-rich learning environment for both students and instructors.

Key Achievements:

The project accomplished all primary objectives: Comprehensive course management with three-tier hierarchy (courses, modules, lessons) supporting multiple content types (video, text, quiz); Advanced video processing pipeline transcoding videos to multiple quality levels (360p-1080p) with AI-generated subtitles in English and Arabic; AI-powered features integrating Google Gemini AI for course outline generation, personalized learning roadmaps, and content transformation into study materials; Real-time communication implementing WebRTC-based voice and video calling with Socket.IO for instant messaging; Secure payment integration with Stripe for PCI-compliant payment processing and webhook-based enrollment automation; Comprehensive progress tracking with lesson completion monitoring, course progress calculation, and quiz performance analytics; Modern user experience with intuitive, responsive interface using Next.js, React, TailwindCSS, and Radix UI.

Technical Implementation:

The platform was built using a modern, scalable technology stack: Frontend (Next.js 14, React, TypeScript, TailwindCSS), Backend (Next.js API Routes, Node.js, Socket.IO), Database (MySQL with 29 tables and optimized indexing), Cloud Services (Google Cloud Storage, Google Cloud Video Transcoder, Firebase), AI Services (Google Gemini 2.0

Flash, OpenAI Whisper), Payment (Stripe), and Real-Time (WebRTC, Socket.IO). The implementation follows industry best practices including RESTful API design, secure authentication with bcrypt and session management, SQL injection prevention through parameterized queries, and WCAG 2.1 Level AA accessibility compliance.

6.2 Contributions to the Field

This project makes several significant contributions to the field of educational technology:

AI-Driven Personalization: Demonstrates practical implementation of AI for personalized learning experiences including automated learning path generation based on student goals and available courses, intelligent content transformation from video to multiple study formats, and context-aware course recommendations.

Automated Content Processing: Showcases efficient automation of content preparation tasks including multi-quality video transcoding without manual intervention, automatic subtitle generation in multiple languages, and thumbnail extraction and optimization.

Integrated Communication: Unlike traditional LMS that rely on external tools, Neureom integrates real-time communication directly into the learning context with in-platform voice and video calling, course-specific chat conversations, and instructor-student direct communication.

Modern Architecture: Demonstrates effective use of modern web technologies for educational platforms including server-side rendering with Next.js for performance and SEO, component-based architecture for maintainability, and cloud-native design for scalability.

Open Development Approach: The comprehensive technical documentation provides a reference for future educational platform development, detailing implementation strategies, challenges encountered, and solutions applied.

6.3 Lessons Learned

Throughout the development process, several important lessons emerged:

Technical Lessons: AI prompt engineering requires careful prompt design with explicit structure requirements, error handling for malformed responses, and fallback strategies. Video transcoding and subtitle generation are resource-intensive operations requiring cloud-based solutions for scalability and parallel processing for efficiency. In-memory state management for Socket.IO connections works for single-server deployments but requires distributed solutions (Redis) for horizontal scaling. Peer-to-peer connections face challenges in restrictive networks, necessitating TURN servers and HTTP polling fallbacks for reliability. Proper indexing and query optimization are critical for performance, especially for progress tracking and analytics queries.

Project Management Lessons: Building features incrementally and testing thoroughly before moving to the next feature prevented accumulation of technical debt. Maintaining comprehensive documentation throughout development facilitated debugging, feature additions, and knowledge transfer. Regular consideration of user experience from both student and instructor perspectives led to more intuitive interfaces and workflows. Choosing established, well-documented technologies (Next.js, MySQL, Stripe) reduced development time and improved reliability compared to experimental alternatives. Leveraging managed cloud services (Google Cloud, Firebase) allowed focus on core features rather than infrastructure management.

6.4 Future Work

While Neureom successfully implements its core objectives, several enhancements would further improve the platform:

6.4.1 Scalability Enhancements

Distributed Architecture: Implement Redis for distributed session storage and Socket.IO adapter, deploy multiple server instances behind load balancer, separate video processing into dedicated microservice, and implement message queue (RabbitMQ/Kafka) for asynchronous tasks.

Database Optimization: Implement read replicas for analytics queries, add caching layer (Redis) for frequently accessed data, optimize query performance with materialized views, and implement database sharding for large-scale deployment.

CDN Integration: Distribute static assets via CDN (Cloudflare/CloudFront), implement edge caching for API responses, and optimize image delivery with responsive formats (WebP, AVIF).

6.4.2 Feature Enhancements

Advanced AI Capabilities: Implement AI-powered content recommendations based on learning patterns, develop adaptive quiz difficulty based on student performance, create AI tutor chatbot for answering student questions, generate automated course assessments from lesson content, and implement plagiarism detection for assignments.

Enhanced Communication: Implement group video calling with selective forwarding unit (SFU), add screen sharing capability for presentations, develop whiteboard collaboration tool, implement breakout rooms for group discussions, and add call recording with consent management.

Gamification: Implement achievement badges and trophies, create leaderboards for course completion and quiz performance, develop streak tracking for daily learning goals, add experience points (XP) and leveling system, and implement challenges and competitions.

Advanced Analytics: Develop predictive analytics for student success, create visual dashboards with charts and graphs, implement learning pattern analysis, add A/B testing framework for course optimization, and generate automated insights and recommendations.

6.4.3 Mobile Application

Develop native iOS and Android apps beyond web view, implement offline course downloads, add push notification management, create mobile-optimized video player, implement background audio playback, and add picture-in-picture mode for multitasking.

6.4.4 Accessibility and Internationalization

Implement audio descriptions for video content, add sign language interpretation support, develop voice navigation for visually impaired users, create high-contrast themes for low

vision users, add support for additional languages (French, Spanish, German, Chinese), implement automatic content translation, and support right-to-left (RTL) languages beyond Arabic.

6.4.5 Assessment and Certification

Implement peer review assignments, add essay grading with AI assistance, develop project-based assessments, create portfolio submissions, generate completion certificates with verification codes, implement blockchain-based credential verification, partner with accreditation bodies, and create skill badges for specific competencies.

6.4.6 Community and Social Learning

Implement study groups with private spaces, add mentorship matching system, create course-specific discussion forums, develop peer-to-peer tutoring marketplace, and implement social learning features (follow, share, collaborate).

6.4.7 Integration and API

Integrate with learning record stores (LRS) for xAPI/SCORM, connect with student information systems (SIS), implement single sign-on (SSO) with SAML/OAuth, integrate with calendar applications (Google Calendar, Outlook), develop RESTful API for third-party developers, create webhooks for event notifications, and implement OAuth 2.0 for secure API access.

6.5 Conclusion

The Neureom project successfully demonstrates that modern web technologies, artificial intelligence, and cloud services can be effectively combined to create a comprehensive, intelligent online learning platform. The system addresses key challenges in digital education including content creation efficiency, personalized learning paths, real-time collaboration, and automated content processing.

The platform's AI-powered features distinguish it from traditional learning management systems by automating course generation, personalizing learning experiences, and transforming content into multiple study formats. The integration of real-time communication enables synchronous learning experiences that bridge the gap between online and in-person education.

While the current implementation achieves all primary objectives and demonstrates robust functionality, the identified future enhancements provide a clear roadmap for evolution into an enterprise-grade educational platform capable of serving diverse educational institutions and individual learners at scale.

The comprehensive technical documentation, including architecture details, implementation strategies, and lessons learned, serves as a valuable resource for future development of educational technology platforms. The project demonstrates that with careful planning, modern technology selection, and user-centric design, it is possible to create sophisticated educational platforms that enhance learning outcomes and improve the educational experience for both students and instructors.

Neureom represents a significant step forward in AI-powered educational technology, providing a solid foundation for continued innovation in digital learning platforms. The platform's modular architecture, comprehensive feature set, and clear enhancement pathways position it well for future growth and adaptation to evolving educational needs.

Bibliography

- [1] Vercel. “Next.js 14 Documentation - The React Framework for the Web.” Next.js Official Documentation, 2024. <https://nextjs.org/docs>
- [2] Meta. “React - A JavaScript Library for Building User Interfaces.” React Official Documentation, 2024. <https://react.dev/>
- [3] Microsoft. “TypeScript - JavaScript with Syntax for Types.” TypeScript Official Documentation, 2024. <https://www.typescriptlang.org/docs/>
- [4] Tailwind Labs. “TailwindCSS - Rapidly Build Modern Websites.” TailwindCSS Official Documentation, 2024. <https://tailwindcss.com/docs>
- [5] Radix UI. “Radix Primitives - Unstyled, Accessible Components for React.” Radix UI Documentation, 2024. <https://www.radix-ui.com/primitives>
- [6] shadcn. “shadcn/ui - Beautifully Designed Components.” shadcn/ui Documentation, 2024. <https://ui.shadcn.com/>
- [7] Oracle Corporation. “MySQL 8.0 Reference Manual.” MySQL Official Documentation, 2024. <https://dev.mysql.com/doc/refman/8.0/en/>
- [8] Node.js Foundation. “Node.js Documentation.” Node.js Official Documentation, 2024. <https://nodejs.org/en/docs/>
- [9] Socket.IO. “Socket.IO Documentation - Bidirectional and Low-Latency Communication.” Socket.IO Official Documentation, 2024. <https://socket.io/docs/>
- [10] Google Cloud. “Google Cloud Storage Documentation.” Google Cloud Platform Documentation, 2024. <https://cloud.google.com/storage/docs>
- [11] Google Cloud. “Transcoder API Documentation.” Google Cloud Platform Documentation, 2024. <https://cloud.google.com/transcoder/docs>
- [12] Google Cloud. “Google Cloud Platform Documentation.” Google Cloud Official Documentation, 2024. <https://cloud.google.com/docs>
- [13] Firebase. “Firebase Documentation - Build and Run Apps.” Firebase Official Documentation, 2024. <https://firebase.google.com/docs>

- [14] Firebase. “Firebase Cloud Messaging Documentation.” Firebase Official Documentation, 2024. <https://firebase.google.com/docs/cloud-messaging>
- [15] Google DeepMind. “Gemini API Documentation - Google’s Most Capable AI Model.” Google AI Documentation, 2024. <https://ai.google.dev/docs>
- [16] OpenAI. “Whisper - Robust Speech Recognition via Large-Scale Weak Supervision.” OpenAI Research, 2022. <https://openai.com/research/whisper>
- [17] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. “Robust Speech Recognition via Large-Scale Weak Supervision.” arXiv preprint arXiv:2212.04356, 2022.
- [18] Stripe, Inc. “Stripe API Documentation - Payment Processing for the Internet.” Stripe Official Documentation, 2024. <https://stripe.com/docs/api>
- [19] Stripe, Inc. “Stripe Checkout Documentation.” Stripe Official Documentation, 2024. <https://stripe.com/docs/payments/checkout>
- [20] WebRTC Project. “WebRTC - Real-Time Communication for the Web.” WebRTC Official Documentation, 2024. <https://webrtc.org/>
- [21] Bergkvist, A., Burnett, D. C., Jennings, C., Narayanan, A., & Aboba, B. “WebRTC 1.0: Real-Time Communication Between Browsers.” W3C Recommendation, 2021. <https://www.w3.org/TR/webrtc/>
- [22] Loreto, S., Romano, S. P., Castaldi, P., & Miniero, L. “Real-Time Communication with WebRTC: Peer-to-Peer in the Browser.” O’Reilly Media, 2014.
- [23] FFmpeg Developers. “FFmpeg Documentation - A Complete, Cross-Platform Solution to Record, Convert and Stream Audio and Video.” FFmpeg Official Documentation, 2024. <https://ffmpeg.org/documentation.html>
- [24] Lovell, L. “Sharp - High Performance Node.js Image Processing.” Sharp Documentation, 2024. <https://sharp.pixelplumbing.com/>
- [25] bcrypt Contributors. “bcrypt - A Library to Help You Hash Passwords.” npm bcrypt Documentation, 2024. <https://www.npmjs.com/package/bcrypt>
- [26] OWASP Foundation. “OWASP Top Ten Web Application Security Risks.” OWASP Official Documentation, 2021. <https://owasp.org/www-project-top-ten/>
- [27] Fielding, R., & Reschke, J. “Hypertext Transfer Protocol (HTTP/1.1): Authentication.” RFC 7235, Internet Engineering Task Force, 2014.

- [28] W3C. “Web Content Accessibility Guidelines (WCAG) 2.1.” W3C Recommendation, 2018. <https://www.w3.org/TR/WCAG21/>
- [29] W3C. “WAI-ARIA Authoring Practices Guide.” W3C Working Draft, 2024. <https://www.w3.org/WAI/ARIA/apg/>
- [30] Siemens, G., & Long, P. “Penetrating the Fog: Analytics in Learning and Education.” *EDUCAUSE Review*, vol. 46, no. 5, 2011, pp. 30-40.
- [31] Baker, R. S., & Inventado, P. S. “Educational Data Mining and Learning Analytics.” In *Learning Analytics: From Research to Practice*, Springer, 2014, pp. 61-75.
- [32] Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. “Systematic Review of Research on Artificial Intelligence Applications in Higher Education.” *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, 2019, pp. 1-27.
- [33] Holmes, W., Bialik, M., & Fadel, C. “Artificial Intelligence in Education: Promises and Implications for Teaching and Learning.” Center for Curriculum Redesign, 2019.
- [34] Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. “Intelligence Unleashed: An Argument for AI in Education.” Pearson Education, 2016.
- [35] Moodle Pty Ltd. “Moodle - Open-Source Learning Platform.” Moodle Official Documentation, 2024. <https://docs.moodle.org/>
- [36] Instructure, Inc. “Canvas LMS Documentation.” Canvas Official Documentation, 2024. <https://community.canvaslms.com/>
- [37] Blackboard Inc. “Blackboard Learn Documentation.” Blackboard Official Documentation, 2024. <https://help.blackboard.com/>
- [38] Coursera, Inc. “Coursera - Build Skills with Online Courses.” Coursera Platform, 2024. <https://www.coursera.org/>
- [39] Udemy, Inc. “Udemy - Online Learning Platform.” Udemy Platform, 2024. <https://www.udemy.com/>
- [40] edX Inc. “edX - Free Online Courses by Harvard, MIT, & More.” edX Platform, 2024. <https://www.edx.org/>
- [41] Google. “Flutter Documentation - Build Apps for Any Screen.” Flutter Official Documentation, 2024. <https://docs.flutter.dev/>
- [42] Google. “Firebase for Flutter.” Firebase Official Documentation, 2024. <https://firebase.google.com/docs/flutter/setup>

- [43] Martin, R. C. “Clean Code: A Handbook of Agile Software Craftsmanship.” Prentice Hall, 2008.
- [44] Fowler, M. “Patterns of Enterprise Application Architecture.” Addison-Wesley Professional, 2002.
- [45] Newman, S. “Building Microservices: Designing Fine-Grained Systems.” O’Reilly Media, 2015.
- [46] Fielding, R. T. “Architectural Styles and the Design of Network-Based Software Architectures.” Doctoral Dissertation, University of California, Irvine, 2000.
- [47] Masse, M. “REST API Design Rulebook.” O’Reilly Media, 2011.
- [48] Date, C. J. “An Introduction to Database Systems.” Pearson Education, 8th Edition, 2003.
- [49] Elmasri, R., & Navathe, S. B. “Fundamentals of Database Systems.” Pearson Education, 7th Edition, 2015.
- [50] Norman, D. A. “The Design of Everyday Things: Revised and Expanded Edition.” Basic Books, 2013.
- [51] Krug, S. “Don’t Make Me Think, Revisited: A Common Sense Approach to Web Usability.” New Riders, 3rd Edition, 2014.
- [52] Nielsen, J., & Budiu, R. “Mobile Usability.” New Riders, 2012.
- [53] MDN Web Docs. “Web APIs - MDN Web Docs.” Mozilla Developer Network, 2024. <https://developer.mozilla.org/en-US/docs/Web/API>
- [54] npm, Inc. “npm Documentation - Package Manager for JavaScript.” npm Official Documentation, 2024. <https://docs.npmjs.com/>
- [55] Git. “Git Documentation - Version Control System.” Git Official Documentation, 2024. <https://git-scm.com/doc>
- [56] GitHub, Inc. “GitHub Documentation.” GitHub Official Documentation, 2024. <https://docs.github.com/>
- [57] ECMA International. “ECMAScript 2024 Language Specification.” ECMA-262, 15th Edition, 2024.
- [58] W3C. “HTML Living Standard.” WHATWG, 2024. <https://html.spec.whatwg.org/>

- [59] W3C. “CSS Specifications.” W3C Official Documentation, 2024. <https://www.w3.org/Style/CSS/specs.en.html>
- [60] IETF. “The OAuth 2.0 Authorization Framework.” RFC 6749, Internet Engineering Task Force, 2012.