



An-Najah National University

Faculty of Engineering & Information Technology

Presented in partial fulfillment of the requirements for Bachelor
degree in Computer Engineering

Graduation Project 1



Code Treasure

Students:

Noor al-huda Hamayel

Rand Hamayel

Supervisor:

Eng. Oula Mardawi

January 23, 2026

Acknowledgment

All thanks and gratitude to An-Najah University in general, and to the Computer Engineering Department and Eng. Oula Mardawi in particular. We also thank our families, friends, and everyone who supported us to accomplish this work

Disclaimer Statement

This report was written by the two students in the Department of Computer Engineering at the College of Engineering, Noor al-huda Hamayel and Rand Hamayel. It may contain linguistic errors or errors in the content because it has not been modified. All opinions, suggestions, and results in it are from the students themselves, and the university is not responsible for any use of this report other than the purpose for which it was presented.

Abstract

Code Treasure is a gamified educational platform available as a mobile application and a website, designed to make learning programming concepts more engaging and interactive for players. The platform combines learning with gameplay by presenting educational content in the form of levels, maps, and challenges, where players progress by answering questions and completing tasks in an entertaining environment.

The system consists of two main user roles: **Admin and Player**. The admin is responsible for fully managing and supervising the platform. This includes creating and organizing levels, categories, questions, avatars, and treasures, as well as defining rewards such as coins for each level. The admin also manages in-game coin packages, avatars (free or paid), references for each category, announcements, and notifications. In addition, the admin dashboard provides comprehensive statistics and analytics, such as the number of players, levels, categories, questions, active users, online players, and most played levels. The admin can also monitor player feedback through comments, manage transactions generated from real payment operations, and control admin accounts.

On the other hand, the player can explore levels displayed on an interactive map, answer questions, and complete levels to earn coins as rewards. Player performance is evaluated based on time, incorrect answers, and hint usage, while progress reflects the number of questions completed within each level. Coins earned by players can be used to purchase hints and avatars. Each avatar is associated with a specific treasure on the map, enhancing the sense of gameplay and immersion. Players can also rate levels, add comments, view other players' profiles, and communicate with each other through a built-in chat feature.

To enhance engagement, the platform includes an AI chatbot for assistance, a secure payment system for purchasing coin packages, password recovery functionality, and a welcoming onboarding experience that grants new players free coins to start playing. Through its interactive design and educational gameplay, Code Treasure aims to transform learning into an enjoyable and motivating experience, bridging the gap between education and entertainment.

Contents

List of Figures	6
1 Introduction	10
1.1 Statement of the problem	10
1.2 Objectives of the work	11
1.3 Scope of the work	11
1.4 Significance of our work	
1.5 Organization of the report	
2 Constraints, Standards/ Codes and Earlier course work	14
2.1 Constraints and limitation.....	14
2.2 Standards and Codes.....	15
2.2.1 React native Front-end Application.....	15
2.2.2 Database Back-end.....	16
2.2.3 Waterfall Model.....	16
2.3 Earlier coursework.....	18
3 Literature Review	19
4 Methodology	22
4.1 Tools, Methods and Programming Languages.....	22
4.1.1 Tools.....	22
4.1.2 programming language.....	23
4.1.3 Database.....	25
5 Results and Discussion	26
5.1 System Evaluation.....	26
5.2 Web Application.....	27
5.2.1 Admin:.....	28
5.2.2 Player.....	42

5.3	Mobile Application.....	65
5.3.1	Admin.....	65
5.3.2	Player.....	69
6	Conclusions and Recommendations	76
6.1	Summary.....	76
6.2	Future Works.....	77
	References	78

List of Figures

5.21 Landing page	27
5.22 Admin	28
5.22 Admin sidebar	29
5.22 Admin dashboard	30
5.22 Avatar management	30
5.38 Category management	32
5.39 Maps management	33
5.40 Levels management	35
5.41 Announcement management	37
5.43 Admins management	37
5.44 Coins and packages	38
5.45 Transactions	39
5.49 Admin edit profile	39
5.50 Tournaments	39
5.51 Player	42
5.52 Player Authentication	42
5.53 Player welcome page	45
5.54 Levels	46
5.55 Store	47
5.56 Dashboard	50
5.57 Billing and payment	51
5.58 Notification	52
5.59 Messages	53
5.60 Player profile	54
5.61 AI chatbot	55
5.62 Game Flow	56
5.63 Tournament	62

List of Tables

Chapter 1

Introduction

1.1 Statement of the problem

With the rapid growth of technology and digital platforms, traditional learning methods have become less engaging for many learners, especially children and beginners. Educational content is often presented in a static and theoretical way, which can reduce motivation and limit long-term retention of knowledge. In addition, many learning platforms lack interactive and game-like elements that encourage users to actively participate and progress through content in an enjoyable manner such as Mimo website.

Learning programming concepts in particular can be challenging for beginners, as it requires logical thinking, problem-solving skills, and continuous practice. Without an engaging environment, learners may lose interest quickly. Therefore, there is a need for an educational platform that transforms learning into an interactive and rewarding experience, combining education with gameplay to increase motivation, engagement, and learning efficiency.

1.2 Objectives of the work

The main objective of this project is to develop a mobile application and a website that provide an interactive and gamified learning experience. The platform aims to encourage users to learn through exploration, challenges,

and rewards rather than traditional learning methods.

The objectives of this work include:

- Designing an educational game-based platform that integrates learning with interactive gameplay.
- Motivating users to complete levels and challenges through rewards, progress tracking, and performance evaluation.
- Providing a structured system for managing levels, questions, categories, and educational content.
- Enhancing user engagement through avatars, maps, animations, and in-game interactions.

1.3 Scope of the work

This project focuses on building a complete and integrated educational gaming platform called **Code Treasure**. The platform includes two main user roles: Admin and Player. The admin is responsible for creating and managing levels, questions, categories, avatars, triggers, rewards, and references related to each category. The admin also monitors player performance, comments, ratings, transactions, and overall system statistics through a dedicated dashboard.

The player interacts with the system by selecting an avatar, exploring the map, answering questions, and completing levels to earn coins and rewards. The platform tracks the player's progress and performance based on time, incorrect answers, and hint usage. Players can also rate levels, leave comments, view other players' profiles, and communicate with each other.

The scope of the work includes both web and mobile environments, ensuring accessibility and consistency across platforms.

1.4 Significance of our work

Developing a gamified educational platform is an important step toward making learning more engaging and effective. By combining educational content with game mechanics such as maps, avatars, rewards, and challenges, the platform increases motivation and encourages continuous learning.

This project helps transform complex concepts into interactive experiences, making learning more enjoyable and less intimidating. It also promotes competition, self-improvement, and problem-solving skills. Through performance tracking and progress monitoring, learners can better understand their strengths and areas for improvement, while administrators gain valuable insights into user behavior and engagement.

1.5 Organization of the report

This report is organized into several chapters. Chapter One introduces the project, presents the problem statement, objectives, scope, and significance of the work. Chapter Two discusses the constraints, standards, and technologies used in the project. Chapter Three explains the system design and methodology adopted during development. Chapter Four presents the implementation details and system features. Chapter Five discusses the results, evaluation, and challenges encountered. Finally, the conclusion summarizes the project and provides recommendations for future improvements. Additional supporting materials are included in the appendices.

Chapter 2

Constraints, Standards/ Codes and Earlier course work

2.1 Constraints and limitation

1- Limited Technical Resources and Documentation:

The documentation available for some of the technologies used in this project, such as Node.js, React Native, and Stripe Payment integration, mainly covers basic use cases. Implementing advanced features like real-time notifications, AI chatbot integration, payment systems, and performance analytics required extensive research, experimentation, and reliance on community resources. Additionally, working within an Nx monorepo environment required a solid understanding of workspace configuration and dependency management.

2- Game Design and User Experience Challenges:

Designing an educational game that feels engaging rather than like a traditional application posed a significant challenge. Achieving a balance between gameplay, learning objectives, animations, and performance optimization required multiple iterations and testing across both web and mobile platforms.

3- Performance and Scalability Considerations:

Handling player progress, performance calculations, transactions, notifications, and real-time interactions introduced challenges related to data consistency and scalability. Ensuring smooth gameplay while maintaining accurate tracking of coins, hints, ratings, and progress required careful backend design.

2.2 Standards and Codes

To ensure reliability, maintainability, and scalability, the project adheres to modern software development standards and best practices. These include modular architecture, RESTful API design, secure authentication mechanisms, and responsive UI principles for cross-platform compatibility. Coding standards such as clean code practices, proper naming conventions, and separation of concerns were followed throughout the development process.

2.2.1 Front-end Application (React Native)

For the front-end development of Code Treasure, **React Native** was chosen as the primary framework. React Native enables the development of cross-platform applications for both web and mobile environments using a shared codebase. This approach significantly reduced development time while ensuring a consistent user experience.

React Native's component-based architecture allowed for the creation of reusable UI elements such as maps, avatars, dashboards, and overlays. The declarative programming model made it easier to manage application states and user interactions. Integration within an **Nx workspace** further enhanced code organization, scalability, and maintainability across multiple application

2.2.2 Database and Back-end

The backend of the project was developed using **Node.js**, an open-source, event-driven runtime environment built on Google Chrome's V8 engine. Node.js was selected for its high performance, scalability, and ability to handle real-time communication and API requests.

For data management, **MySQL** was used as the primary database and managed through **phpMyAdmin**. This relational database structure was suitable for organizing core system data such as users, levels, questions, categories, avatars, transactions, and performance records.

To support real-time features such as chat and instant updates, **Socket.IO** was integrated into the backend, enabling real-time bidirectional communication between players. Additionally, **Firebase** was used to handle notifications, ensuring that players receive instant alerts for announcements, new levels, and system events even when offline.

For media storage, **AWS S3 (iDrive e2)** was utilized to securely store and manage images such as avatars, maps, and other visual assets. This approach ensured scalability, reliability, and efficient access to media content.

Authentication and authorization were implemented using **JSON Web Tokens (JWT)**, providing a secure and stateless method for managing user sessions and protecting API endpoints.

2.2.3 Waterfall Model

The Waterfall model was adopted as the development methodology for this project, following a structured and sequential process.

- **First Phase – Requirement Analysis:**

The project began with identifying system requirements, user roles, game mechanics, constraints, and functional specifications.

- **Second Phase – System Design:**

This phase involved designing the system architecture, database schema, APIs, and user interfaces, including maps, dashboards,

avatars, and performance tracking.

- **Third Phase – Implementation:**

The development phase focused on implementing individual modules such as authentication, level management, performance evaluation, notifications, chat system, AI chatbot, and payment processing. Each module was tested independently.

- **Fourth Phase – Integration and Testing:**

All components were integrated into a single system, followed by comprehensive testing to detect and resolve functional, performance, and usability issues.

- **Final Phase – Deployment:**

The system was prepared for deployment on both web and mobile platforms, ensuring stability, security, and scalability.

2.3 Earlier coursework

During our studies in the Computer Engineering program, we gained foundational knowledge that directly supported the development of this project. This included web development, database systems, software engineering, algorithms, networking, and system design.

Additionally, we did some researches and self-learning related to React Native, Node.js, MySQL, phpMyAdmin, Firebase, Socket.IO, JWT authentication, cloud storage services, and version control using GitHub. This academic background played a significant role in successfully implementing and documenting the Code Treasure platform.

Chapter 3

Literature Review

Gamification in education has emerged as an effective approach to enhance learner engagement, motivation, and knowledge retention. Gamified learning environments integrate game elements such as points, rewards, levels, avatars, and challenges into educational contexts. This approach transforms traditional learning into an interactive experience, encouraging learners to actively participate rather than passively consume information. Research has shown that gamification can positively influence learners' performance, persistence, and overall learning satisfaction.

One of the key concepts related to gamified learning is **game-based learning**, where educational content is delivered through gameplay mechanics. This method allows learners to progress at their own pace, receive instant feedback, and develop problem-solving skills in a motivating environment. Additionally, game-based learning has been found to support cognitive development and enhance critical thinking by engaging learners in decision-making processes and goal-oriented tasks.

(2019) In the study "*The Effects of Gamification on Student Motivation and Engagement*" by Dichev and Dicheva, the authors explored how game elements such as points, badges, and leaderboards influence learner motivation. The study concluded that gamification significantly increases engagement and encourages sustained interaction with learning content. The findings emphasize the importance of well-designed reward systems and clear progression mechanisms, which align with the use of coins, levels, and

performance tracking in digital learning platforms.[1]

(2021) In the research conducted by Hamari et al. titled “*Gamification in Education: A Systematic Literature Review*”, the authors analyzed multiple studies on the application of gamification in educational systems. The results demonstrated that gamified environments improve learner motivation and engagement, particularly when combined with meaningful challenges and feedback. The study highlights that poorly designed gamification elements may reduce effectiveness, stressing the importance of balanced design and user-centered experiences.[2]

Another important aspect of modern educational platforms is **adaptive learning and performance evaluation**. Evaluating learners based on factors such as time, accuracy, and resource usage allows systems to provide personalized feedback and improve learning outcomes. Performance-based evaluation encourages learners to improve their skills while maintaining motivation through visible progress indicators.

(2020) In the study “*Performance-Based Assessment in Digital Learning Environments*” by Johnson and Brown, the authors emphasized the role of performance metrics in enhancing learner accountability and motivation. The study found that learners who received detailed performance feedback showed higher levels of improvement compared to those assessed using traditional scoring methods. This supports the integration of performance indicators such as time, correctness, and hint usage in educational games.[3]

Furthermore, **social interaction and collaboration** play a vital role in increasing engagement within digital learning platforms. Features such as comments, ratings, and peer communication allow learners to exchange knowledge, share experiences, and build a sense of community. These interactions can positively influence learning motivation and create a

supportive learning environment.

In conclusion, existing literature highlights the effectiveness of gamification, performance-based evaluation, and social interaction in enhancing digital learning experiences. By integrating game mechanics, adaptive performance metrics, and interactive features, educational platforms can bridge the gap between learning and entertainment. These findings provide a strong theoretical foundation for the design and development of **Code Treasure**, which aims to deliver programming education through an engaging, game-based learning environment.

Chapter 4

Methodology

4.1 Tools, Methods and Programming Languages

4.1.1 Tools

- **Visual Studio Code:**

Visual Studio Code was used as the primary Integrated Development Environment (IDE) for developing both the frontend and backend of the system. It provided powerful features for code editing, debugging, extensions, and project organization, which facilitated efficient development.

- **Android Studio:**

Android Studio was utilized mainly as an emulator to test the mobile version of the application on multiple virtual devices. This ensured compatibility, responsiveness, and smooth performance across different screen sizes and Android versions.

- **GitHub:**

GitHub was used for version control and collaborative development. It allowed team members to manage source code, track changes, resolve conflicts, and maintain a clear development history throughout the project lifecycle.

- **Firestore:**

Firestore was integrated to support push notifications and real-time features. Firestore Cloud Messaging (FCM) was used to deliver notifications to players when new levels or announcements were published by the admin, as well as to notify players about important in-game events.

- **Postman:**

Postman was employed as an API testing tool to design, test, and validate RESTful APIs. It enabled efficient testing of backend endpoints, ensuring correct data exchange between the frontend and backend.

- **AWS S3 (iDrive e2):**

AWS S3, through iDrive e2, was used as a cloud storage solution for storing images such as avatars, maps, and other game assets. This ensured secure, scalable, and reliable storage with fast access to media files.

- **Socket.IO:**

Socket.IO was integrated to enable real-time communication between players, particularly for the in-game chat functionality. It allowed instant message delivery and real-time interaction within the platform.

4.1.2 programming language

- JavaScript:

JavaScript was used as the primary programming language for backend development through Node.js. It enabled efficient handling of server-side logic, asynchronous operations, and API requests.

- React Native:

React Native was used to develop the frontend for both web and mobile platforms. Its component-based architecture allowed the creation of reusable and interactive UI components, such as maps, dashboards, avatars, and overlays.

- SQL:

Structured Query Language (SQL) was used to interact with the relational database, enabling efficient data storage, retrieval, and management through phpMyAdmin.

4.1.3 Database

For data storage, the project utilized a **MySQL relational database**, managed using **phpMyAdmin**. This database was used to store core system data, including user accounts, levels, questions, categories, avatars, triggers, transactions, player progress, performance metrics, comments, and ratings.

The relational structure of MySQL ensured data integrity and efficient querying, making it suitable for handling structured and interconnected data. phpMyAdmin provided a user-friendly interface for database management, allowing easy monitoring, modification, and maintenance of the database throughout the development process.

Chapter 5

Results and Discussion

In this chapter, the final outcomes of the **Code Treasure** project are presented and discussed. The chapter highlights the implemented functionalities, system performance, and user interaction through detailed descriptions of the platform's features. Screenshots are used to demonstrate the practical implementation of the system on both web and mobile platforms.

5.1 System Evaluation

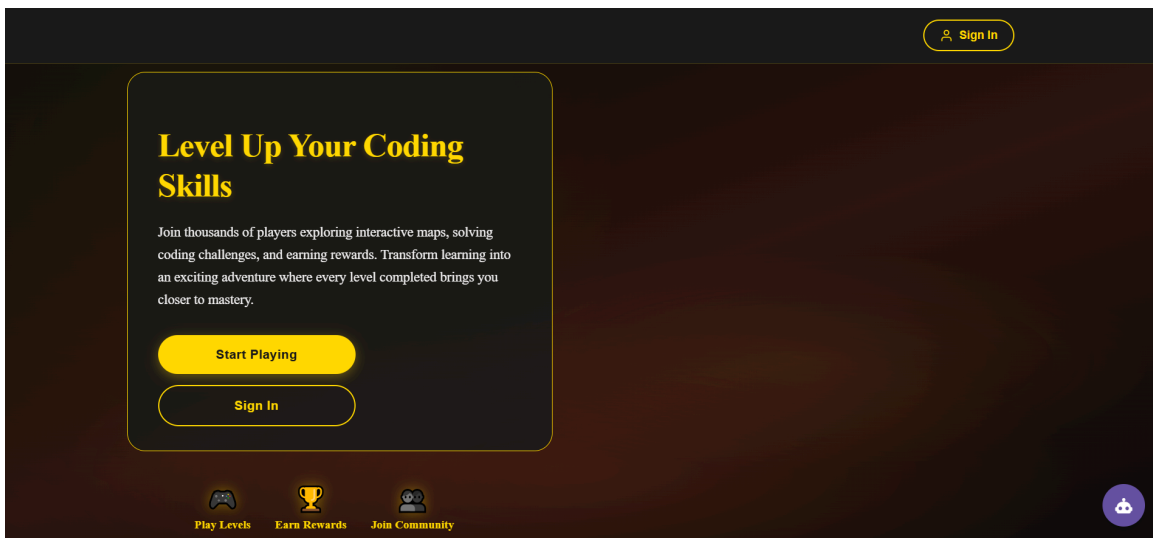
To evaluate the effectiveness of **Code Treasure**, the system was tested through real usage scenarios focusing on usability, performance, and engagement. The evaluation aimed to verify that the platform successfully combines learning programming concepts with an interactive game-based experience.

The results showed that presenting programming challenges through maps, avatars, and levels increased user engagement and motivation. The reward system, progress tracking, and performance evaluation encouraged players to complete levels and improve their skills.

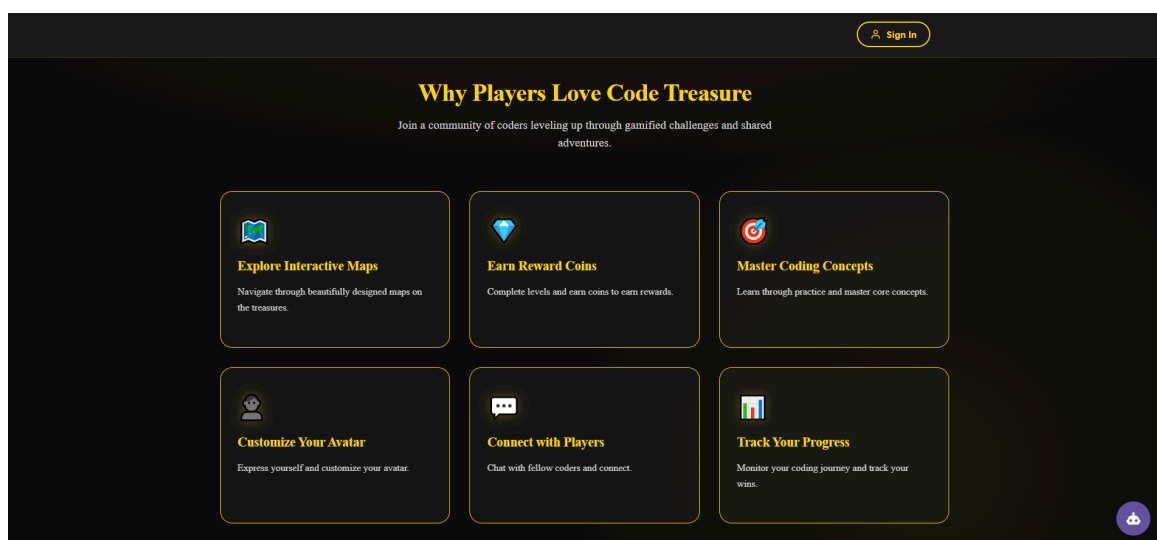
5.2 Web Application

5.2.1 Landing page:

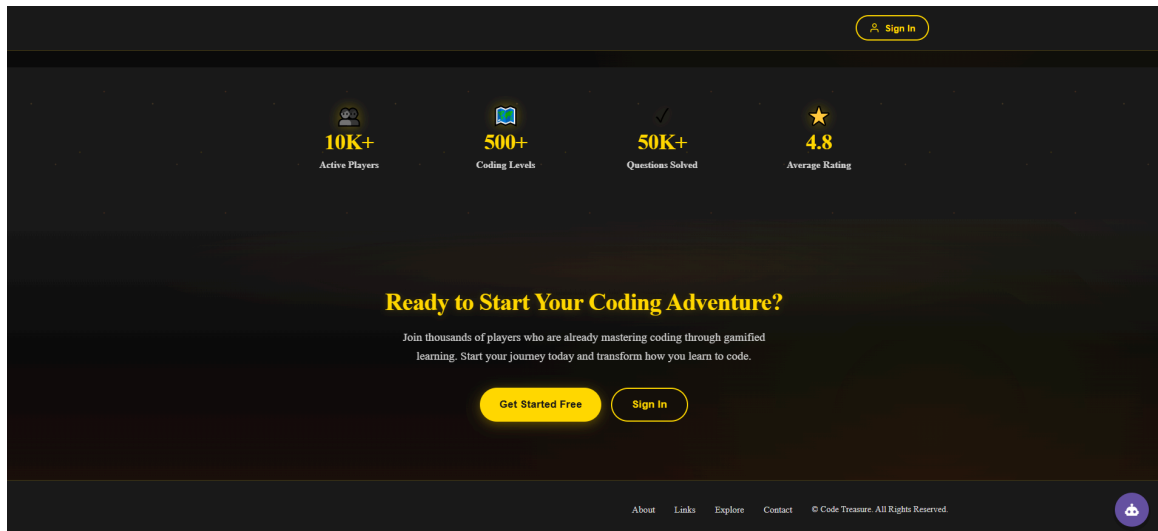
The landing page was added to introduce the Code Treasure project and present its main idea in a simple and attractive way. It explains the concept of the game-based learning system and highlights its key features, such as learning through levels, collecting coins, and using avatars. The landing page also includes clear navigation options such as Sign Up and Login to guide users into the platform. Its main goal is to create a strong first impression and encourage users to start using the application.



(a) Landing page1



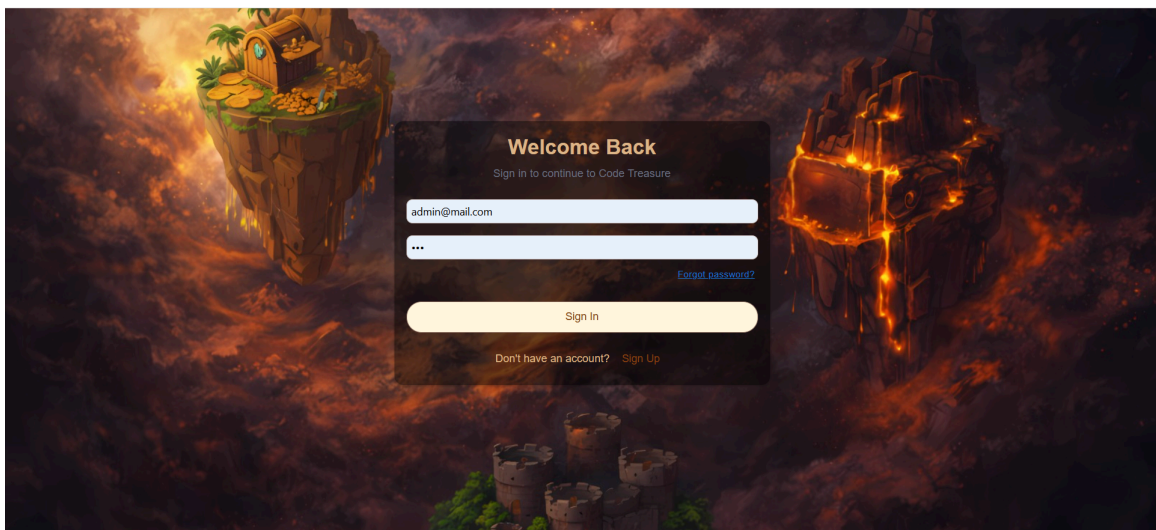
(b) Landing page2



(c) Landing page3

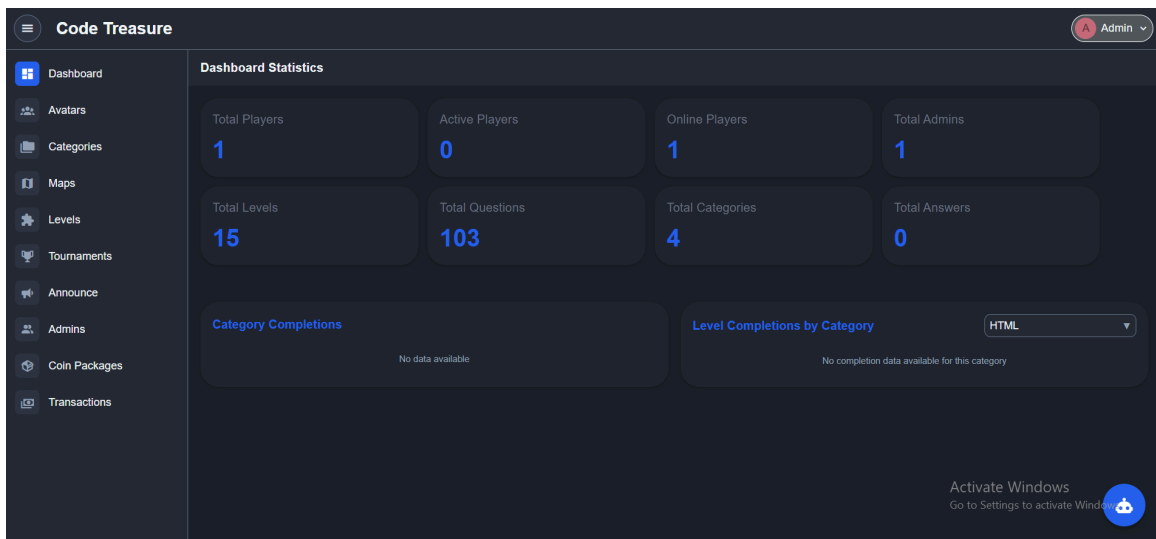
5.2.2 Admin

1. **Sign in:** The admin must authenticate using email and password to access the system. Authentication is secured using JWT (JSON Web Token) to protect sensitive operations.



(a) sign in

2. Admin sidebar



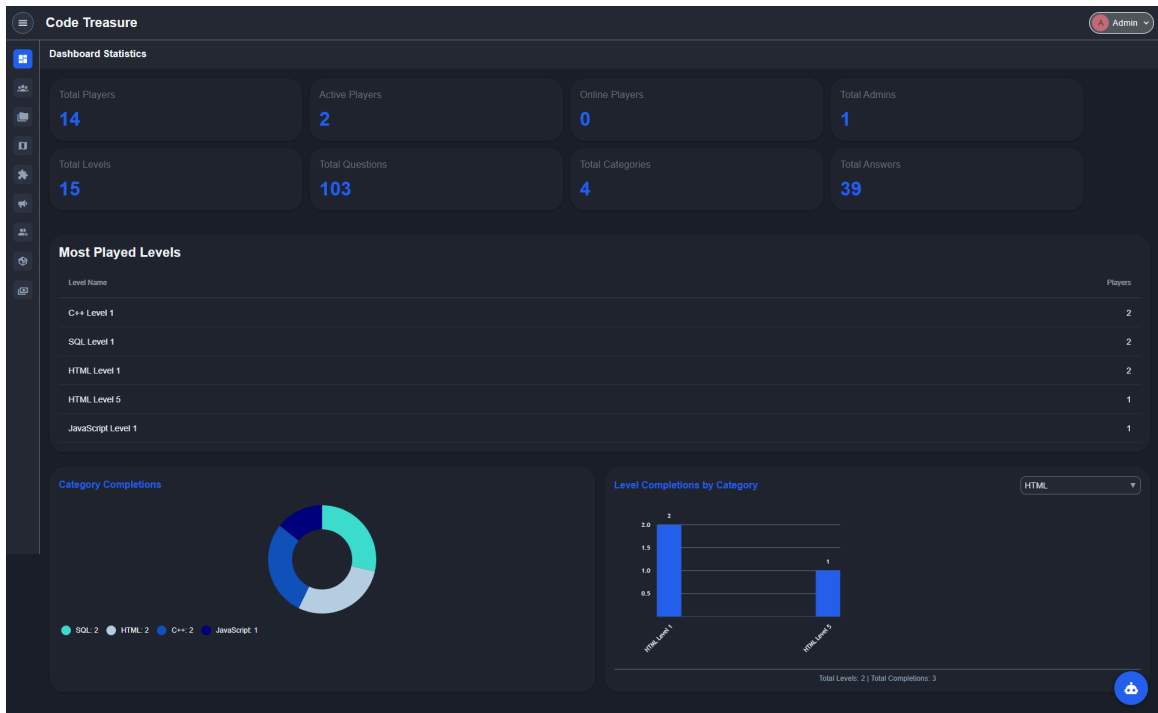
(b) Admin sidebar

3. Admin dashboard: The dashboard provides a comprehensive overview of the platform, displaying key statistics such as total players, total levels, active players, online players, total questions, total categories, total admins, and total answers. It also highlights the most played levels.

Dashboard Charts

The Admin Dashboard includes two main analytical charts to monitor player progress and engagement:

- Category Completions (Pie Chart):**
 This chart represents the distribution of completed levels across all categories. Each slice shows the number of levels completed in a specific category, giving the admin a quick overview of which categories are most actively completed by players.
- Level Completions by Category (Bar Chart):**
 This chart allows the admin to select a specific category. The X-axis displays the levels belonging to the selected category, while the Y-axis represents the number of players who have completed each level. This provides detailed insight into player engagement at the level level within each category.



(c) Admin dashboard

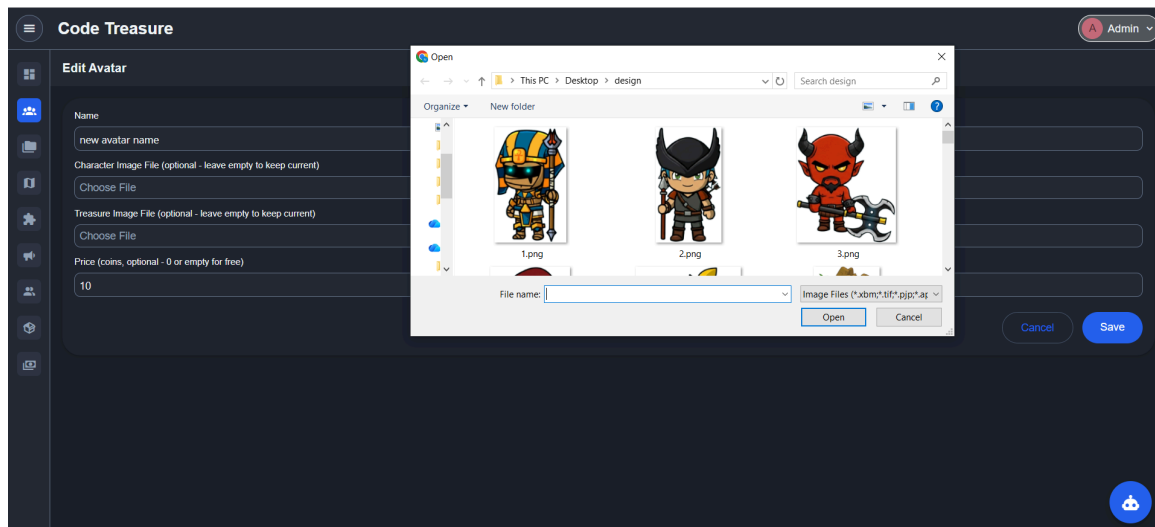
4. Avatars: avatar and treasure management the admin can view all available avatars and treasures also he can edit each and create a new avatar.

The screenshot displays the 'Code Treasure' Admin dashboard, specifically the 'Avatars' management section. The top navigation bar includes a menu icon, the title 'Code Treasure', and a user profile dropdown for 'Admin'. The section title 'Avatars' is followed by a 'Create Avatar' button. Below this is a table listing available avatars:

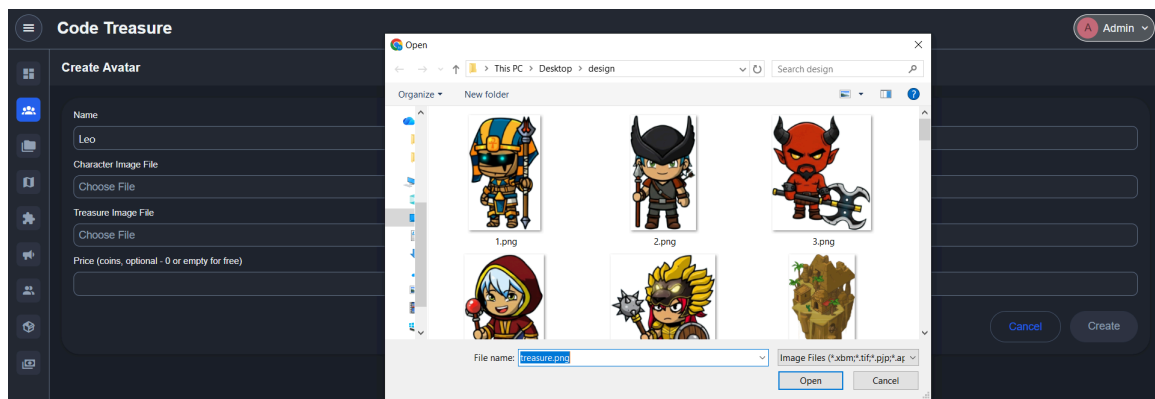
ID	Name	Character Image	Treasure Image	Actions
1	Archer			Edit
2	Vellum			Edit
3	Brute			Edit
4	Leo			Edit
5	Anath			Edit

At the bottom right, there is a status bar indicating 'Activate Windows' and a small blue icon.

(a) Listing avatars

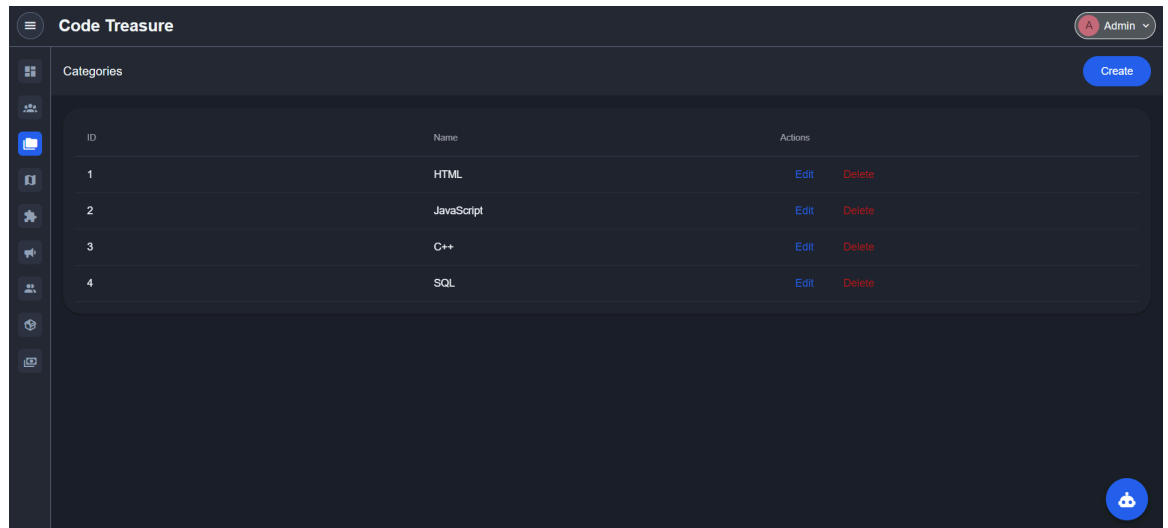


(b) Editing avatar



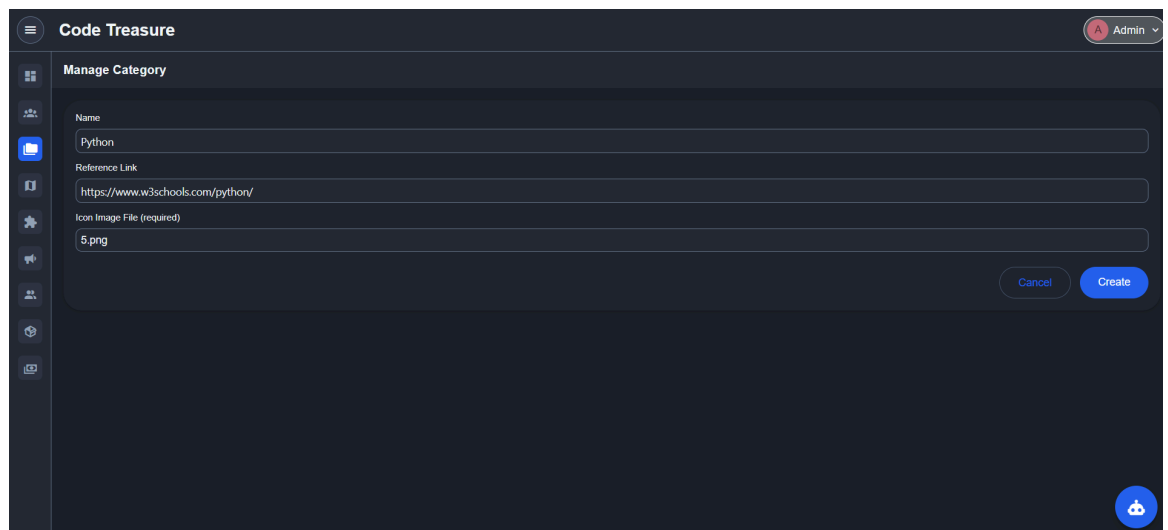
(c) Creating avatar

5. Categories: Categories such as JavaScript, CSS, HTML, React, Node.js, Python, SQL, and Git are created and managed by the admin to organize learning content.



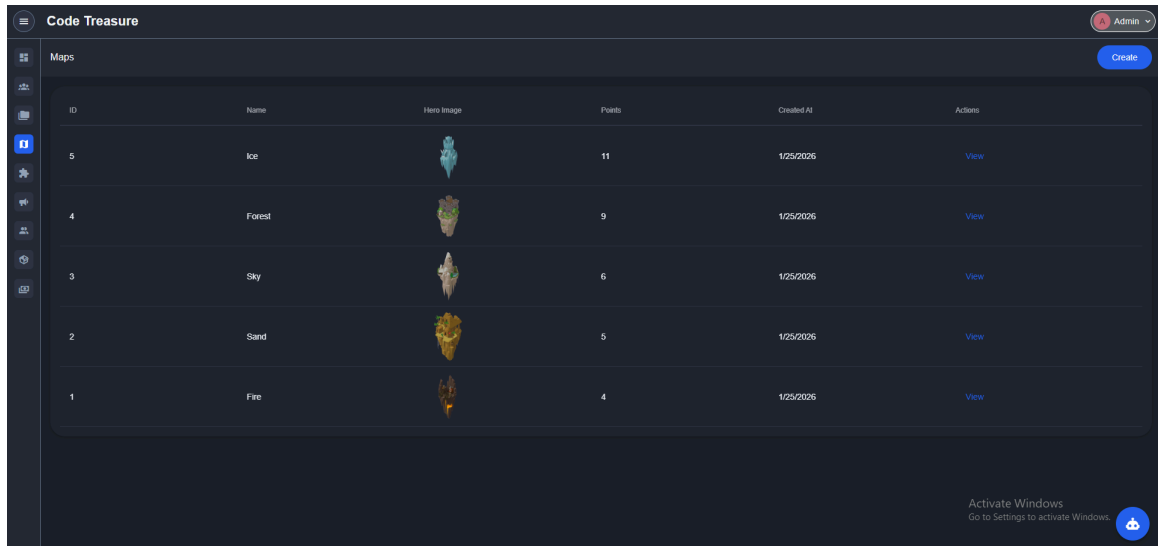
(a) listing categories

Within the category creation , admin can add a reference so that the player can learn about the category.



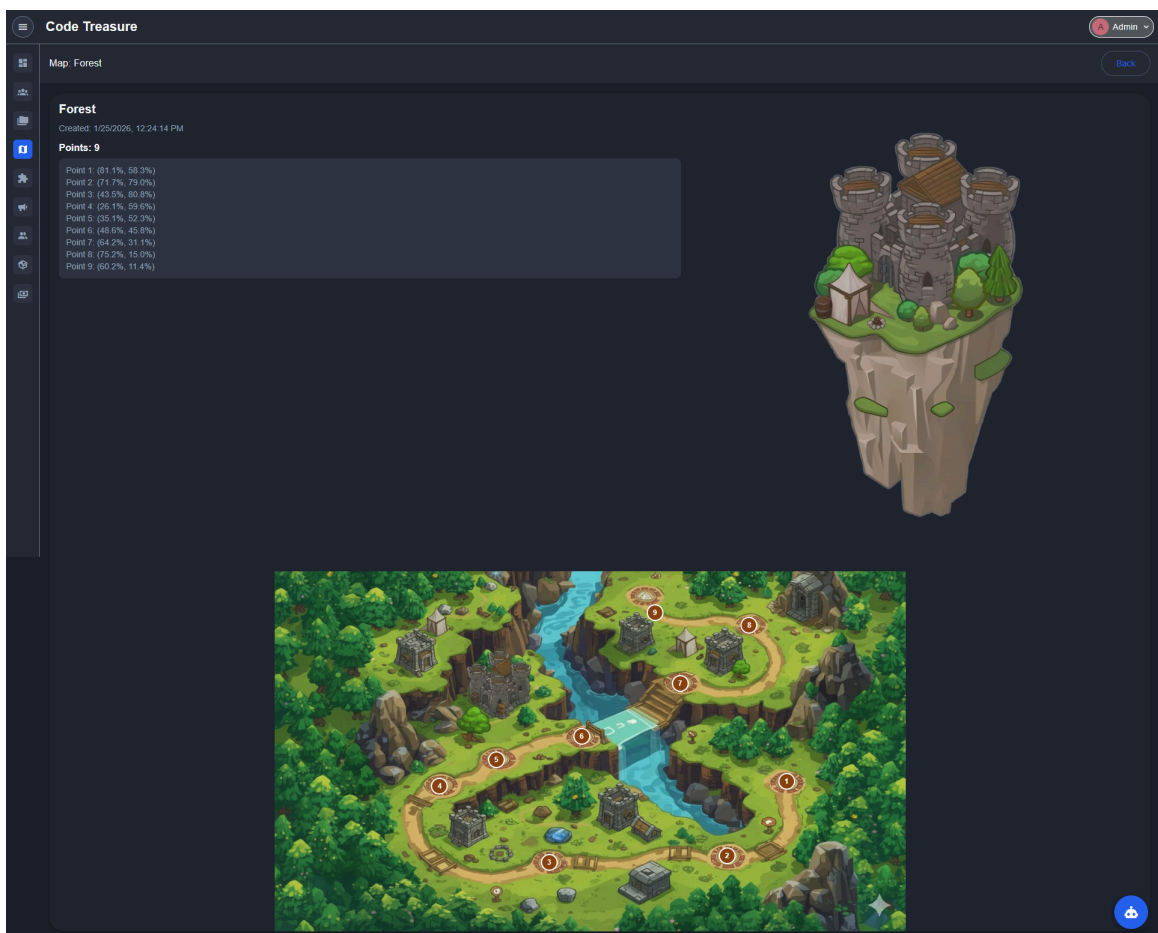
(b) Creating category

6. Maps :maps are themes designed to be used in multiple levels , in the creation process admin select a specific points on the map to determine the path for the avatar to reach the treasure ,these points represents the number of question for this level.

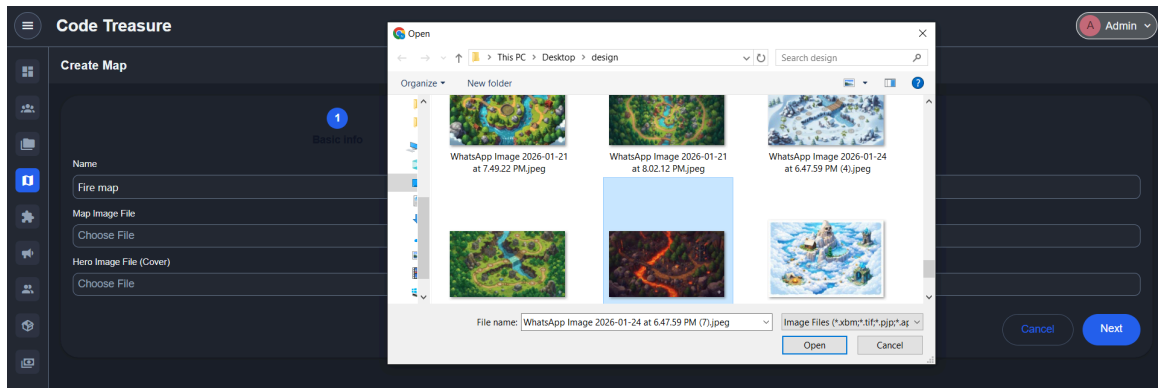


ID	Name	Hero Image	Points	Created At	Actions
5	Ice		11	1/25/2026	View
4	Forest		9	1/25/2026	View
3	Sky		6	1/25/2026	View
2	Sand		5	1/25/2026	View
1	Fire		4	1/25/2026	View

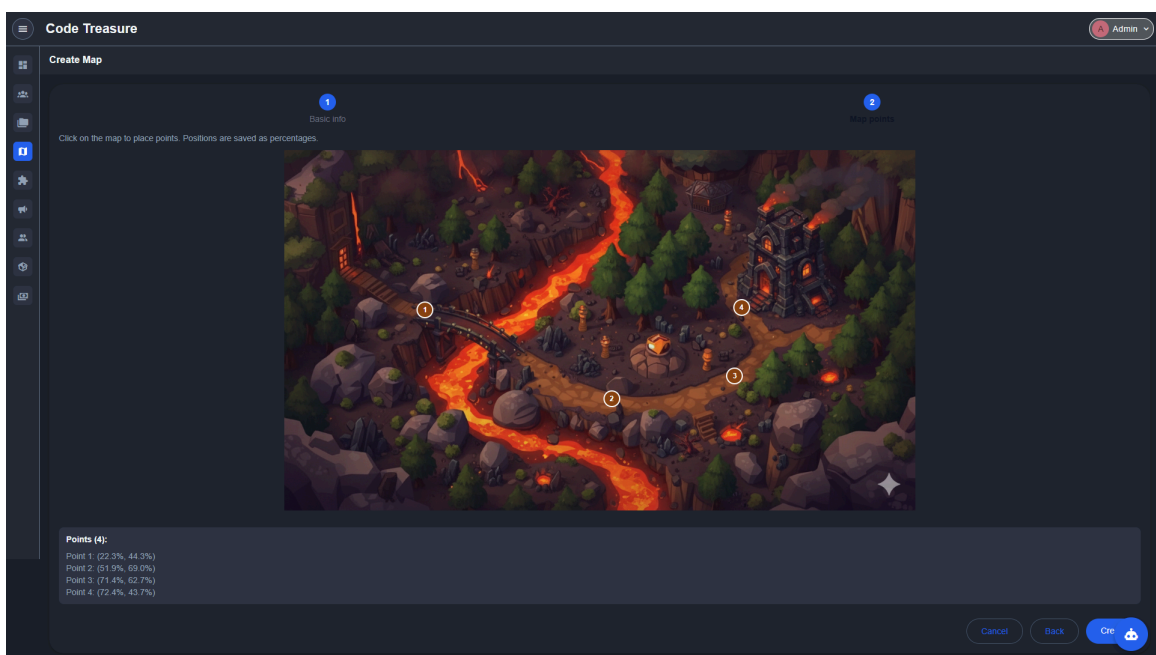
(a) Listing maps



(b) View map details



(c) Creating map



(d) points selection

7. Levels: Level Management

The admin can view level details and players comments on this level, create, edit, and delete levels. While creating a level, the admin defines the category, difficulty (easy, medium, hard), coins reward, select map and create questions.

Code Treasure

Admin

Levels

Create

ID	Name	Map	Category	Reward Coins	Difficulty	Actions		
1	HTML Level 1	Fire	HTML	10	easy	View	Edit	Delete
2	HTML Level 2	Sand	HTML	15	easy	View	Edit	Delete
3	HTML Level 3	Sky	HTML	20	medium	View	Edit	Delete
4	HTML Level 4	Forest	HTML	25	medium	View	Edit	Delete
5	HTML Level 5	Ice	HTML	30	hard	View	Edit	Delete
6	JavaScript Level 1	Fire	JavaScript	10	easy	View	Edit	Delete
7	JavaScript Level 2	Sand	JavaScript	15	easy	View	Edit	Delete
8	JavaScript Level 3	Sky	JavaScript	20	medium	View	Edit	Delete
9	JavaScript Level 4	Ice	JavaScript	25	hard	View	Edit	Delete
10	C++ Level 1	Fire	C++	10	easy	View	Edit	Delete

(a) Listing levels

Code Treasure

Admin

HTML Level 3

Edit

Back to Levels

Category: HTML

Difficulty: Medium

Reward Coins: 20

Questions Count: 3

Level Rating: 4.5

Reference

Map



Comments (0)

No comments yet.

(b) View level

Code Treasure

Create Level

Basic info + Map Questions

Name
HTML level 5

Reward Coins
30

Category
HTML

Difficulty
medium

Map
Fire

Selected map has 4 point(s). 4 question(s) will be created.

(c) Creating level 1

- Creating the questions of this level

Code Treasure

Create Level

Basic info + Map Questions

Fill in the details for each question. One question per map point (1:1 relationship).

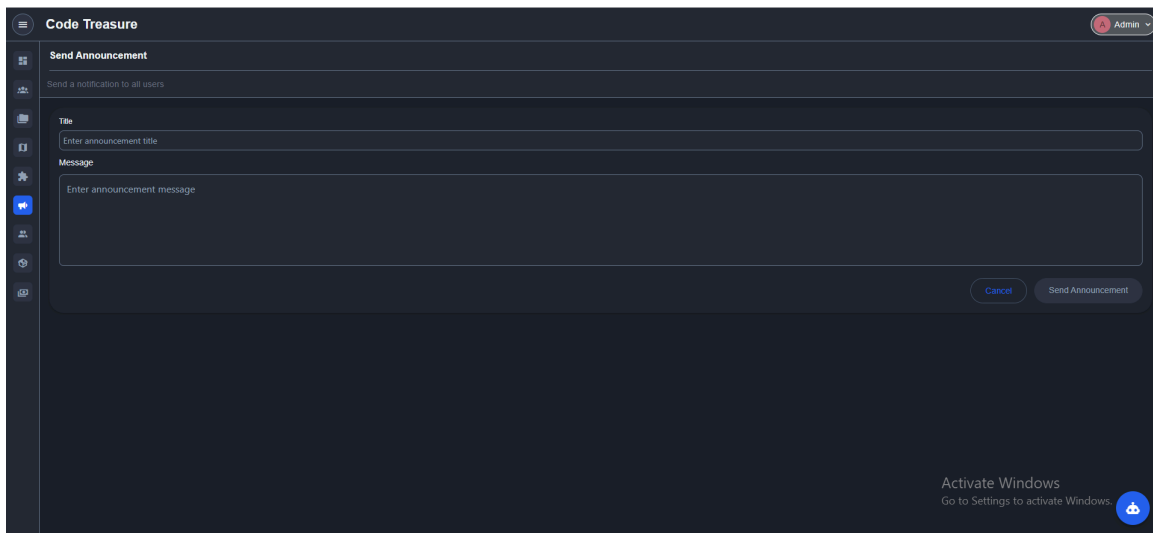
Question 1 (Point ID: 1)
Position: 20.33%, 44.23%
Description
Hint
Option label Remove Add Option

Question 2 (Point ID: 2)
Position: 37.48%, 57.4%
Description
Hint
Option label Remove Add Option

Question 3 (Point ID: 3)
Position: 60.71%, 69.9%
Description

(d) Create questions

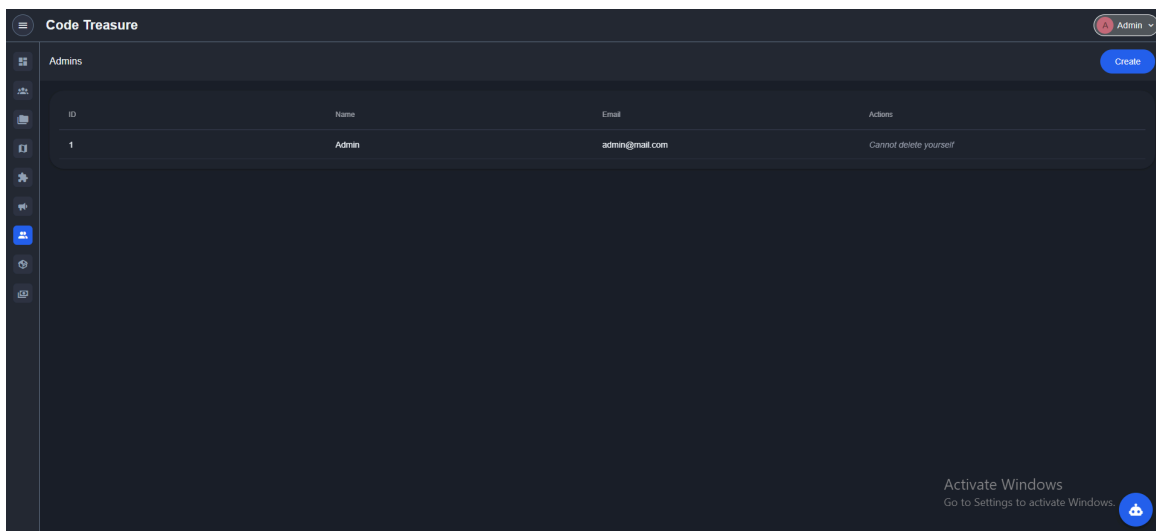
8. Announcements: The admin can send announcements to all players.



The screenshot shows the 'Send Announcement' form in the Code Treasure application. The form has a dark theme and includes a sidebar with navigation icons. The main content area contains a 'Title' field with the placeholder 'Enter announcement title' and a 'Message' field with the placeholder 'Enter announcement message'. At the bottom right of the form are two buttons: 'Cancel' and 'Send Announcement'. The top right corner of the application shows the user 'Admin' with a dropdown arrow. An 'Activate Windows' watermark is visible in the bottom right corner.

(a) Announcements

9. Admins: The system allows the main admin to add or remove other admin accounts.

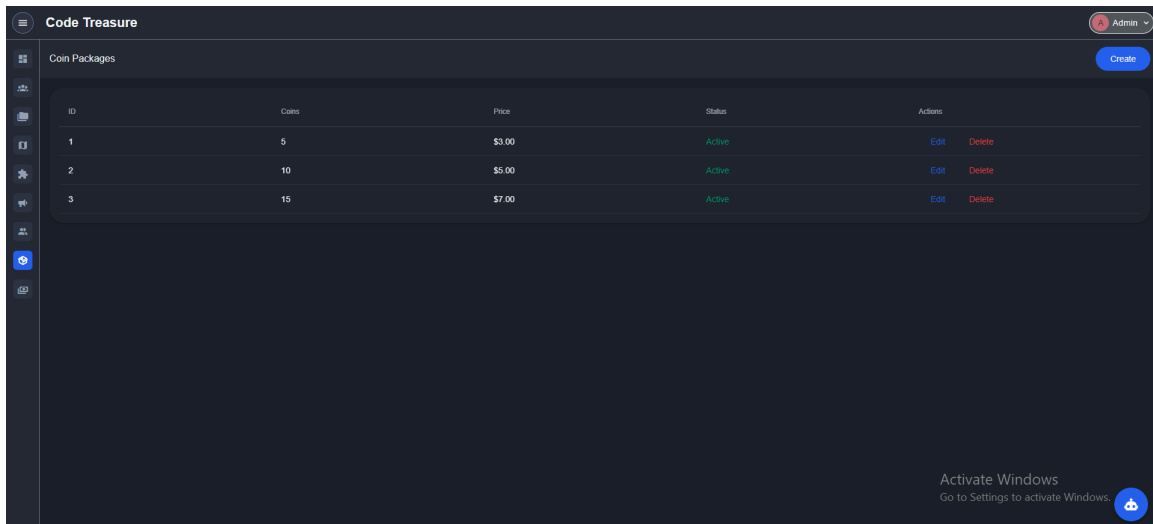


The screenshot shows the 'Admins' table in the Code Treasure application. The table has a dark theme and includes a sidebar with navigation icons. The table has four columns: 'ID', 'Name', 'Email', and 'Actions'. There is one row with the following data: ID 1, Name Admin, Email admin@gmail.com, and Actions Cannot delete yourself. At the top right of the table is a 'Create' button. The top right corner of the application shows the user 'Admin' with a dropdown arrow. An 'Activate Windows' watermark is visible in the bottom right corner.

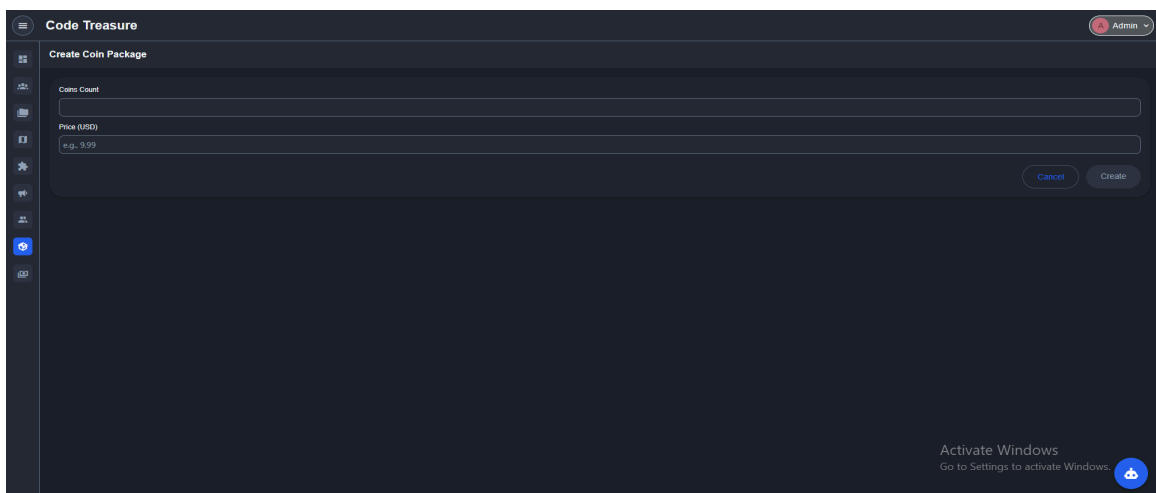
ID	Name	Email	Actions
1	Admin	admin@gmail.com	Cannot delete yourself

(a) Admins

10. Coins and Packages: The admin defines coin packages and sets their prices. Players can purchase these packages using a real payment system integrated with Stripe.

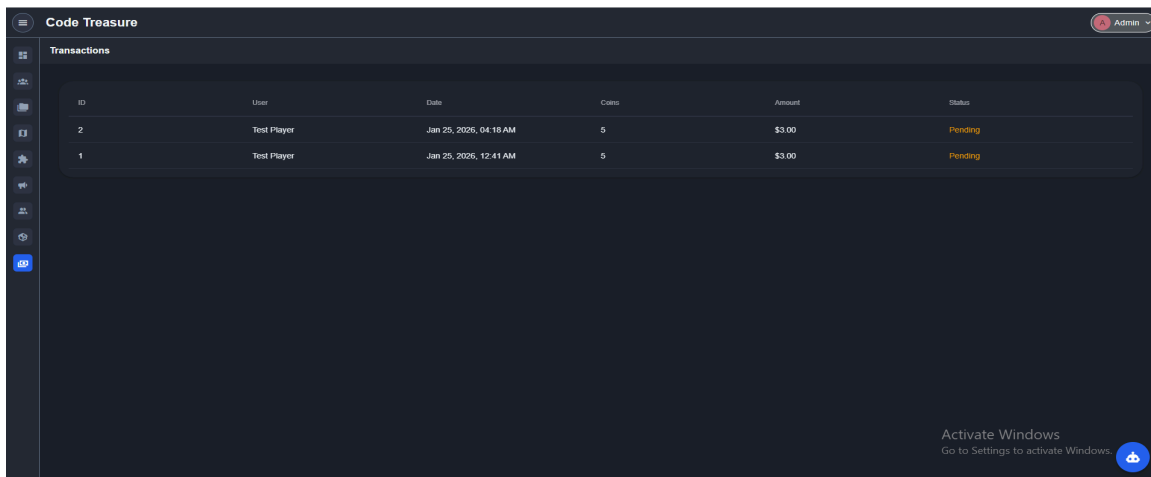


(a) Listing coin packages



(b) Creating package

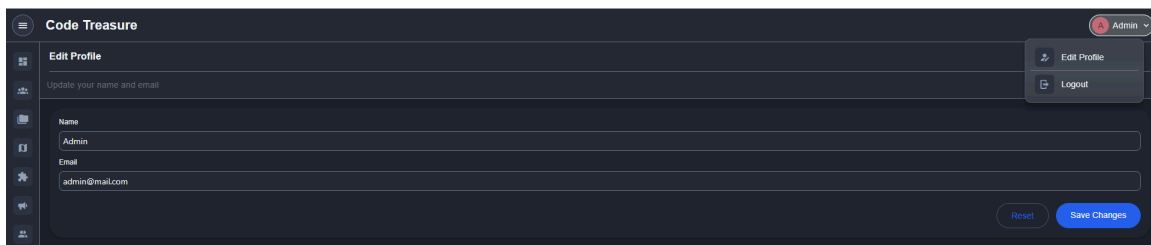
11. Transactions: The admin can view all financial transactions, including purchased coin packages and total revenue generated from players.



ID	User	Date	Coins	Amount	Status
2	Test Player	Jan 25, 2026, 04:18 AM	5	\$3.00	Pending
1	Test Player	Jan 25, 2026, 12:41 AM	5	\$3.00	Pending

(a) List transactions

12. The admin can edit his profile



Code Treasure

Edit Profile

Update your name and email

Name
Admin

Email
admin@mail.com

Reset Save Changes

(a) Edit admin profile

13. Tournaments: The project includes a Tournaments system designed to increase competition and player engagement. In this feature, the admin creates tournaments by selecting a specific set of levels on which players will compete.

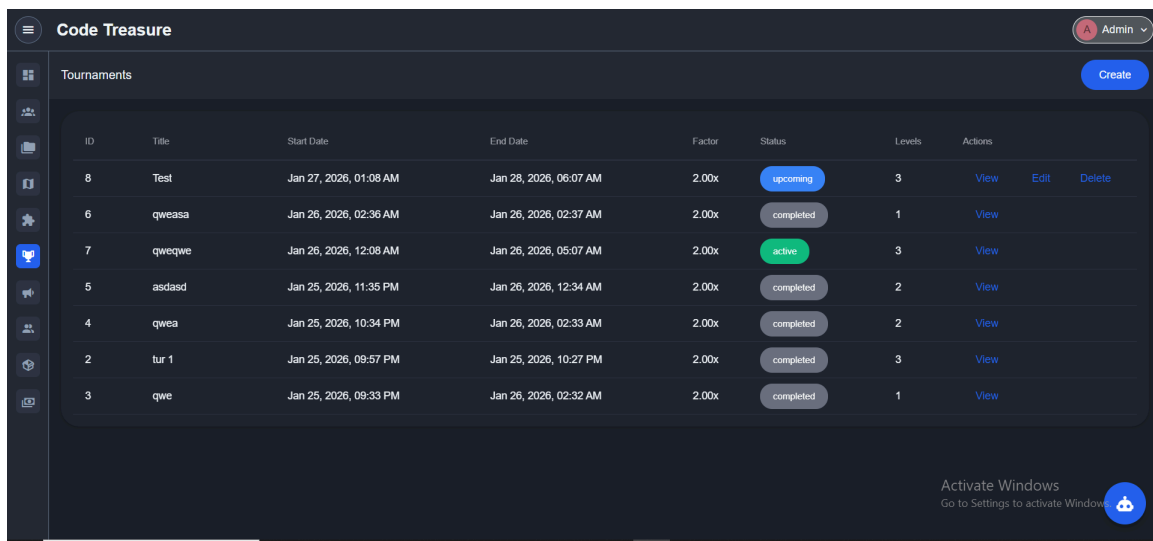
Each tournament has a defined start time and end time, and based on its timeline, it can be in one of three states:

- **Upcoming:** The tournament is scheduled but has not started yet.
- **Active:** The tournament is currently running and players can participate.
- **Completed:** The tournament has ended.

The admin can edit or delete only the Upcoming tournaments.

During an active tournament, players compete by completing the selected levels. If a player completes a level with all questions answered correctly, they receive a reward with multiplied coins instead of the original coin value. The coin multiplier is configurable by the admin, allowing flexibility in defining rewards.

This system encourages fair competition, improves motivation, and adds a dynamic challenge-based learning experience to the game.



ID	Title	Start Date	End Date	Factor	Status	Levels	Actions
8	Test	Jan 27, 2026, 01:08 AM	Jan 28, 2026, 06:07 AM	2.00x	upcoming	3	View Edit Delete
6	qweasa	Jan 26, 2026, 02:36 AM	Jan 26, 2026, 02:37 AM	2.00x	completed	1	View
7	qweqwe	Jan 26, 2026, 12:08 AM	Jan 26, 2026, 05:07 AM	2.00x	active	3	View
5	asdasd	Jan 25, 2026, 11:35 PM	Jan 26, 2026, 12:34 AM	2.00x	completed	2	View
4	qwea	Jan 25, 2026, 10:34 PM	Jan 26, 2026, 02:33 AM	2.00x	completed	2	View
2	tur 1	Jan 25, 2026, 09:57 PM	Jan 25, 2026, 10:27 PM	2.00x	completed	3	View
3	qwe	Jan 25, 2026, 09:33 PM	Jan 26, 2026, 02:32 AM	2.00x	completed	1	View

(a) Listing Tournaments

- Create a Tournament:

The screenshot displays the 'Code Treasure' web application interface for creating a tournament. The header shows the application name 'Code Treasure' and a user profile 'Admin'. The main section is titled 'Create Tournament'. It contains several input fields: 'Title' (filled with 'Tournaments 1'), 'Start Date/Time' (filled with '2026-01-27T03:18'), 'End Date/Time' (filled with '2026-01-28T08:17'), and 'Coins Factor' (filled with '2.0'). Below these is a 'Select Levels' section with a search bar and a list of levels. The list shows 'HTML Level 1' through 'HTML Level 5' and 'JavaScript Level 1'. The first three levels are selected, indicated by checkboxes. A 'Select All' link is visible. At the bottom right, there are 'Cancel' and 'Create Tournam' buttons.

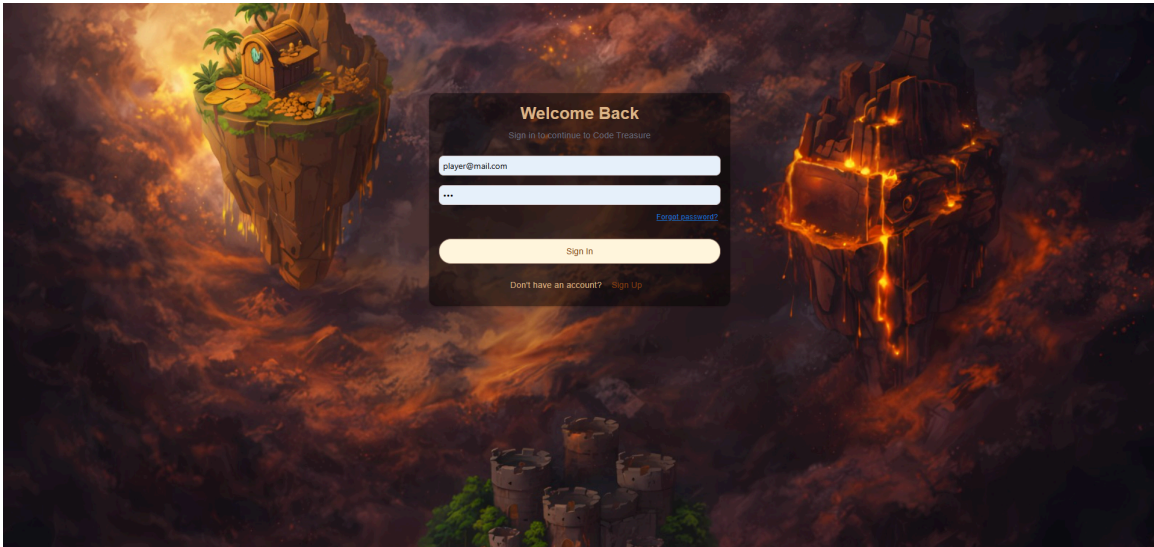
Level	Selected
HTML Level 1	☒
HTML Level 2	☒
HTML Level 3	☒
HTML Level 4	☐
HTML Level 5	☐
JavaScript Level 1	☐

(b) Create Tournament

5.2.3 Player

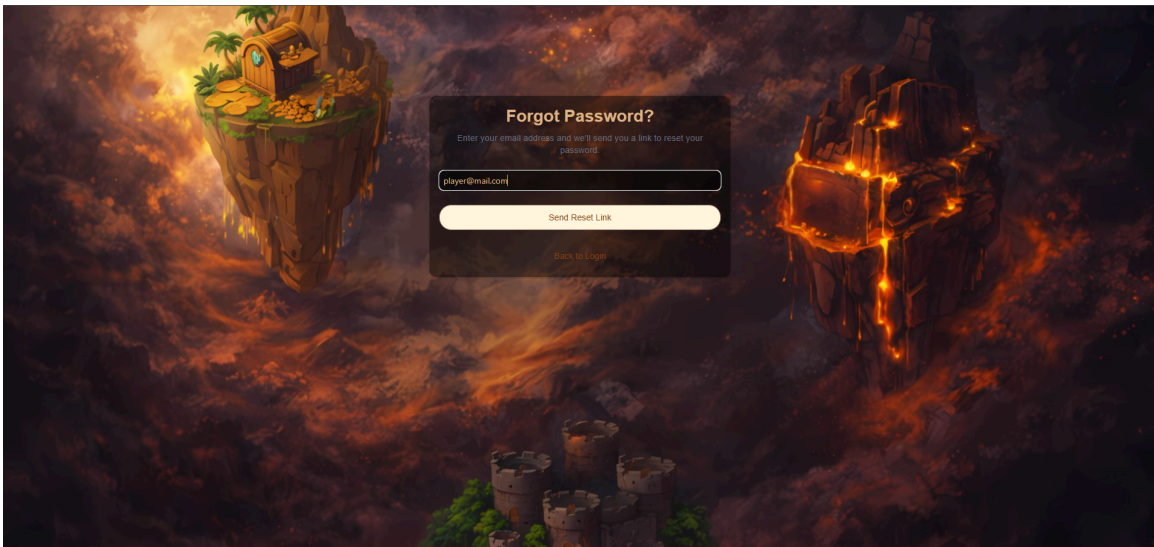
1. **Player authentication:** Players can register and log in using secure authentication by using JWT (JSON Web Token) to protect sensitive operations.

- Sign in form



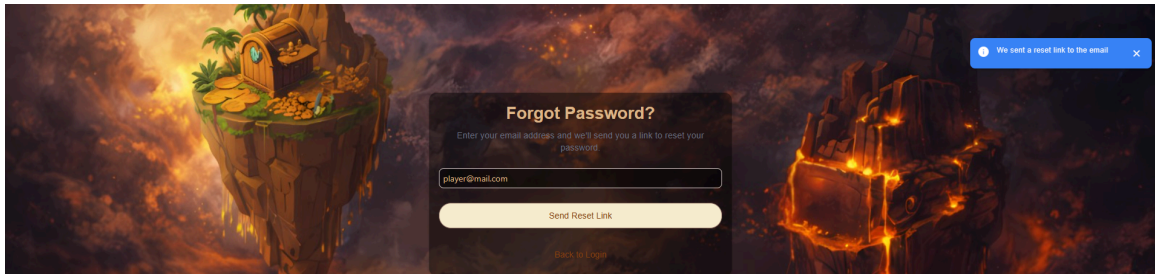
(a) Player sign in

- Forgot password form



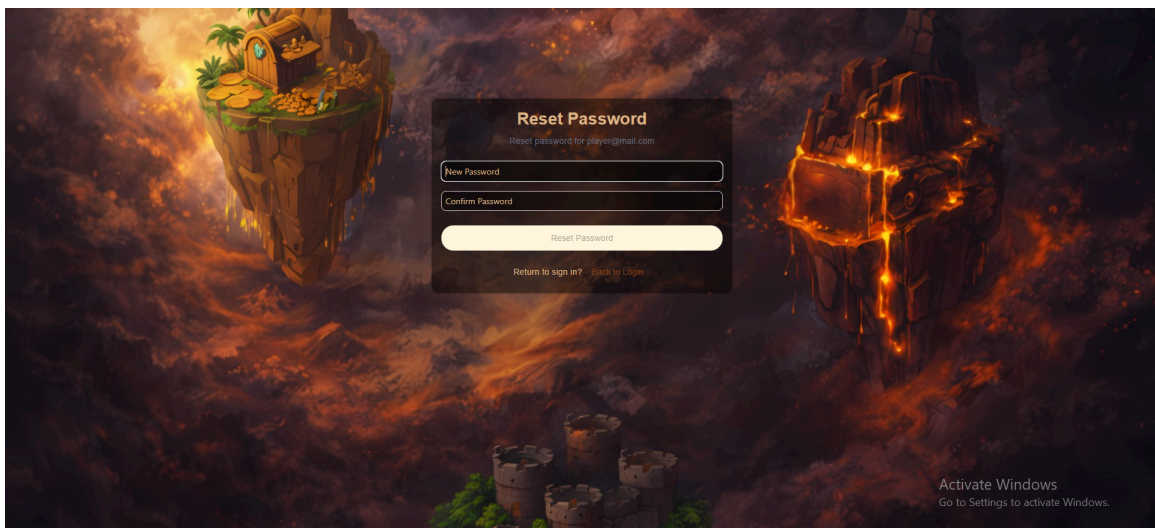
(b) Forgot password

- Real time notification (we sent a rest link to the email)



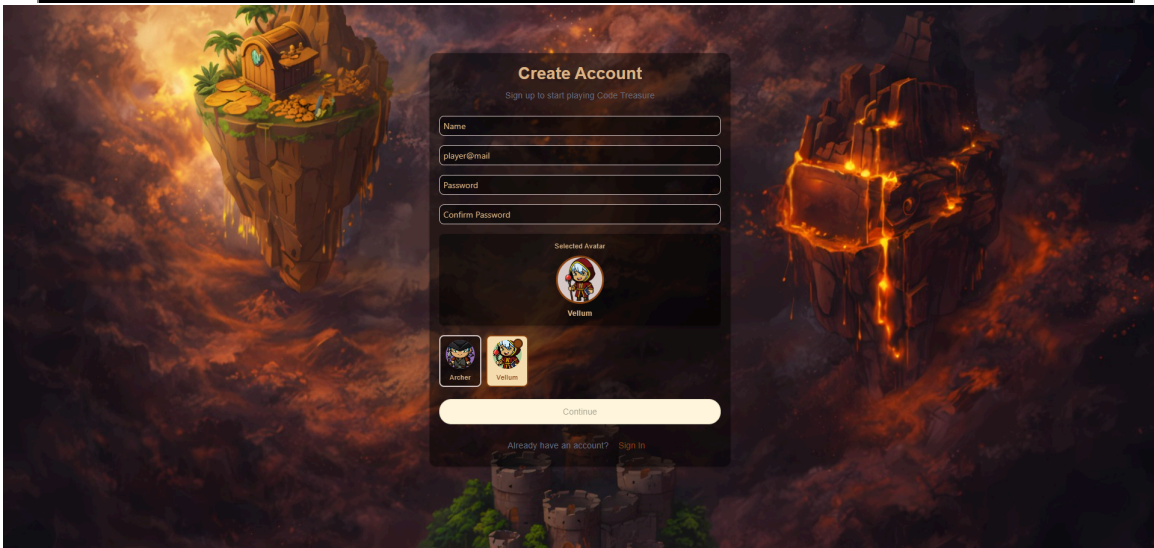
(c) Real time notification

- Reset password form



(d) Reset pass

- Sing up form



(e) Sign up

- Verify your email : it sends a code to your email

Verify Your Email Address

Hello roaa@mail.com,

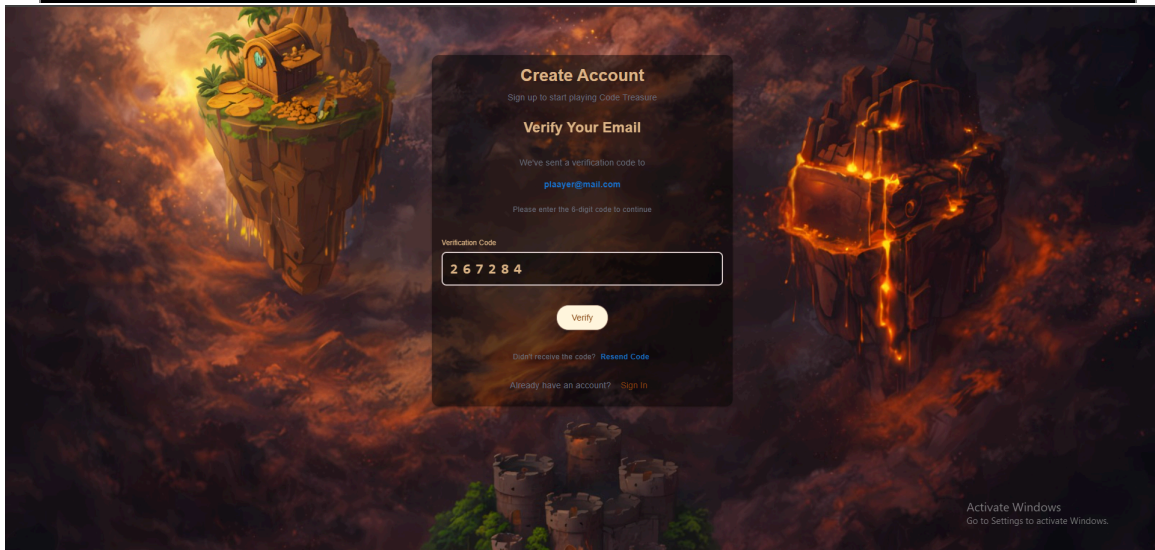
We received a request to create your account for your Code Treasure account. Please use the verification code below to complete this action:

2 6 7 2 8 4

Enter this code in the verification form to continue.

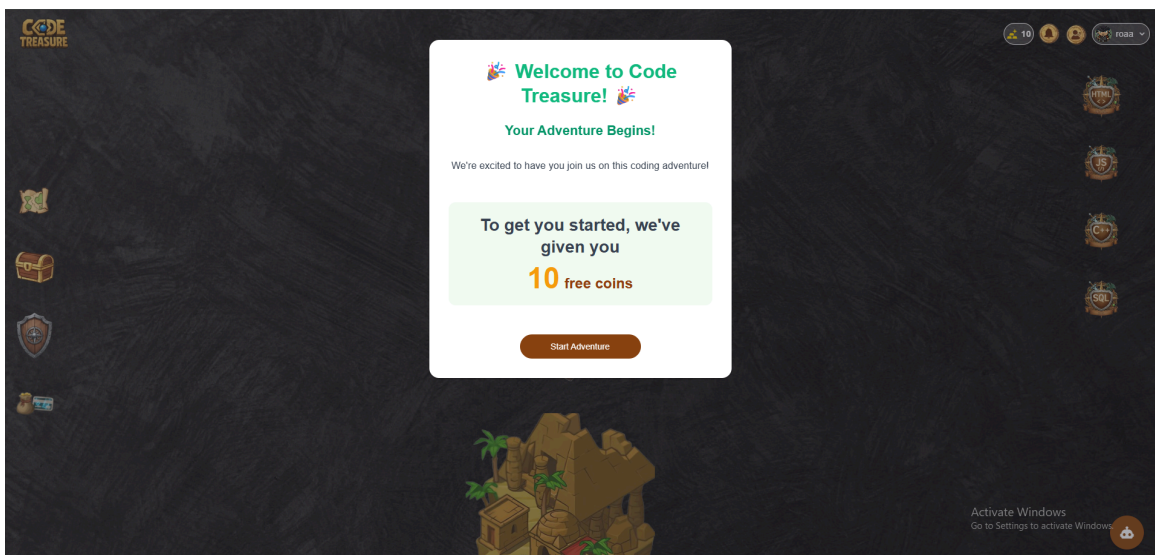
Security Notice:

- This code will expire in 15 minutes



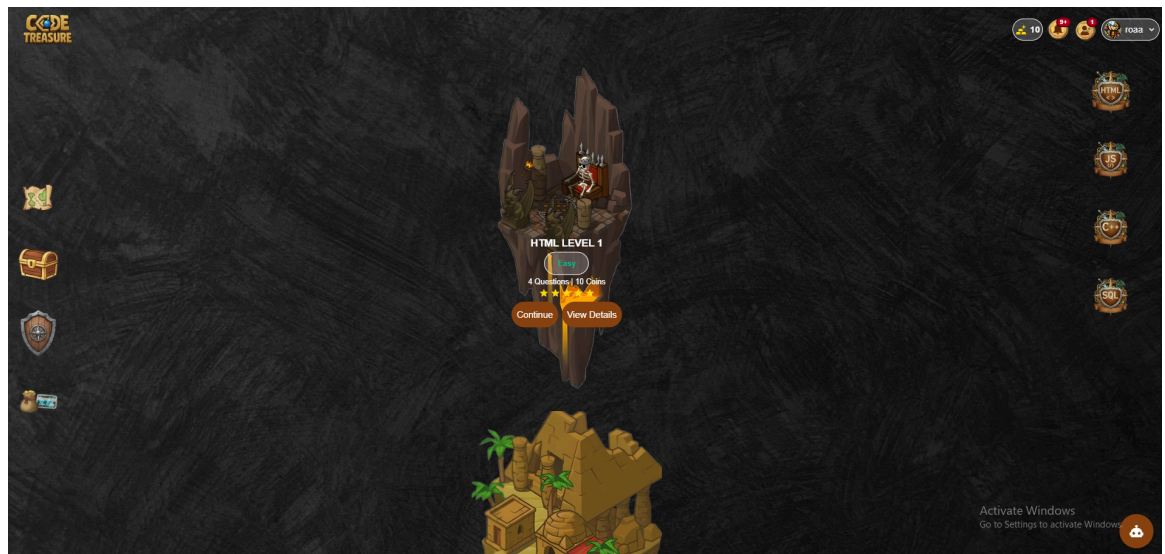
(f) Verify email

2. Welcome popup

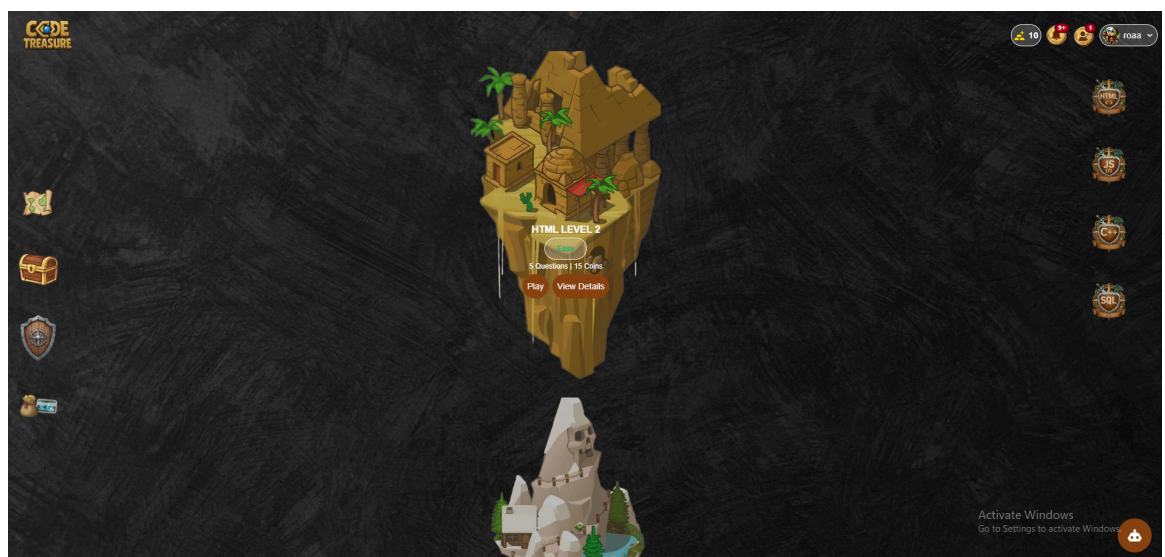


(a) Welcome popup

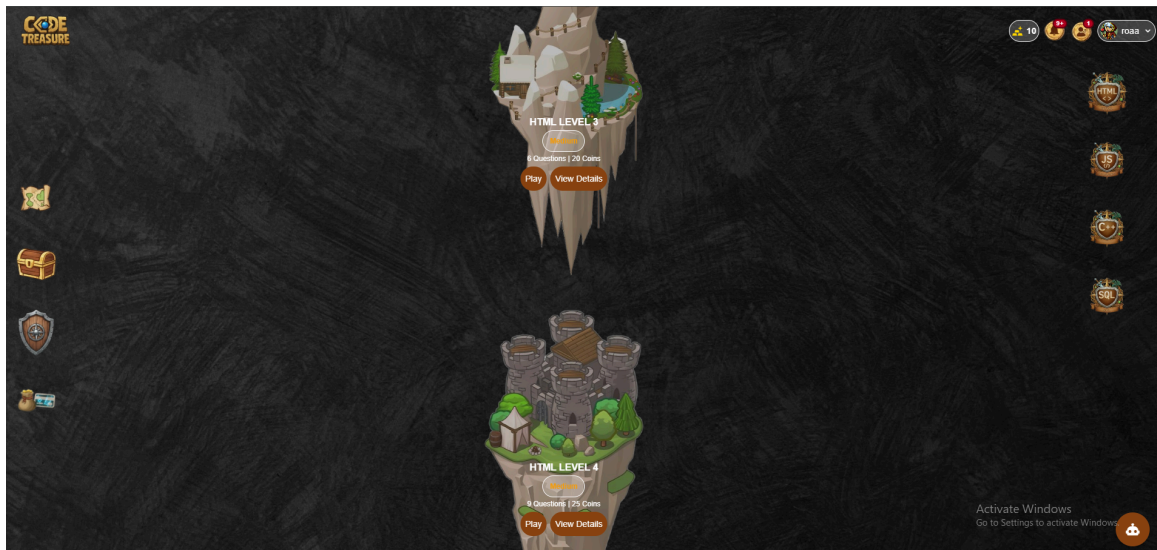
3. Levels: players can choose the category (HTML, CSS, C++, JS)
Then choose the level that he/she wants to play.



(a) Levels view 1

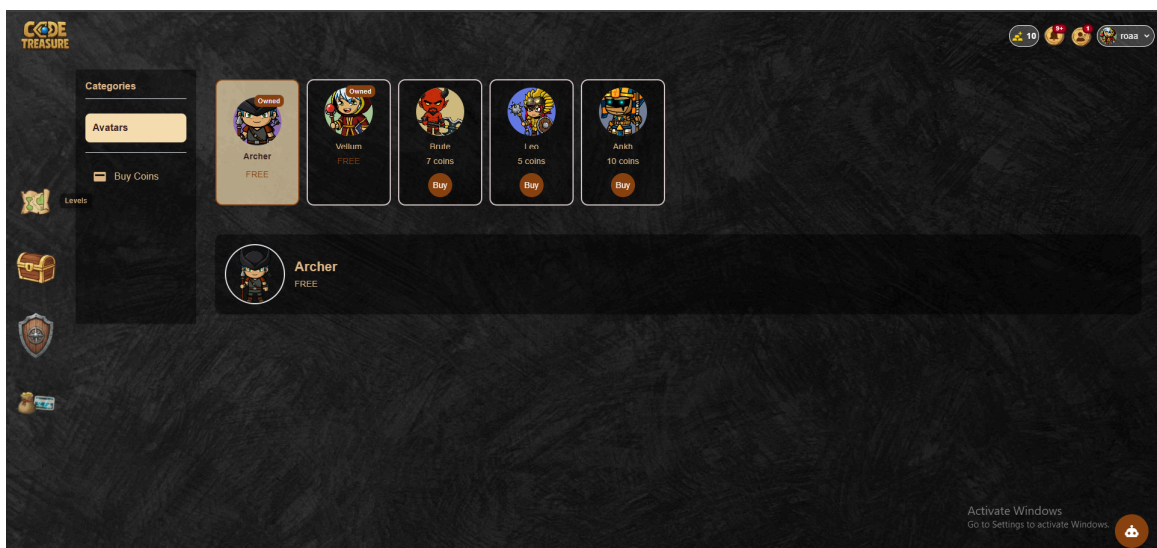


(b) Levels view 2

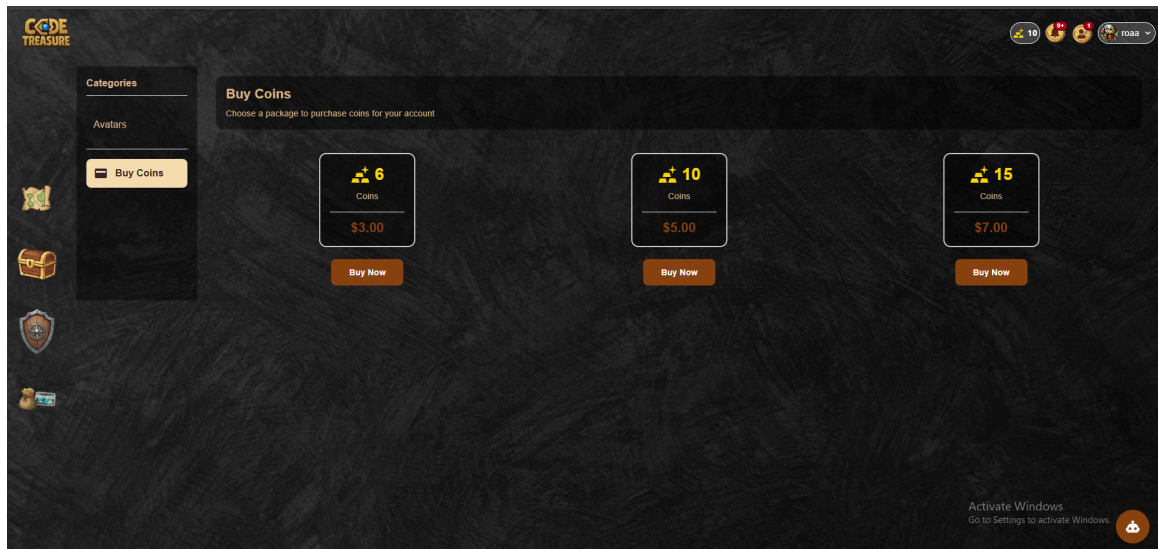


(c) Levels view 3

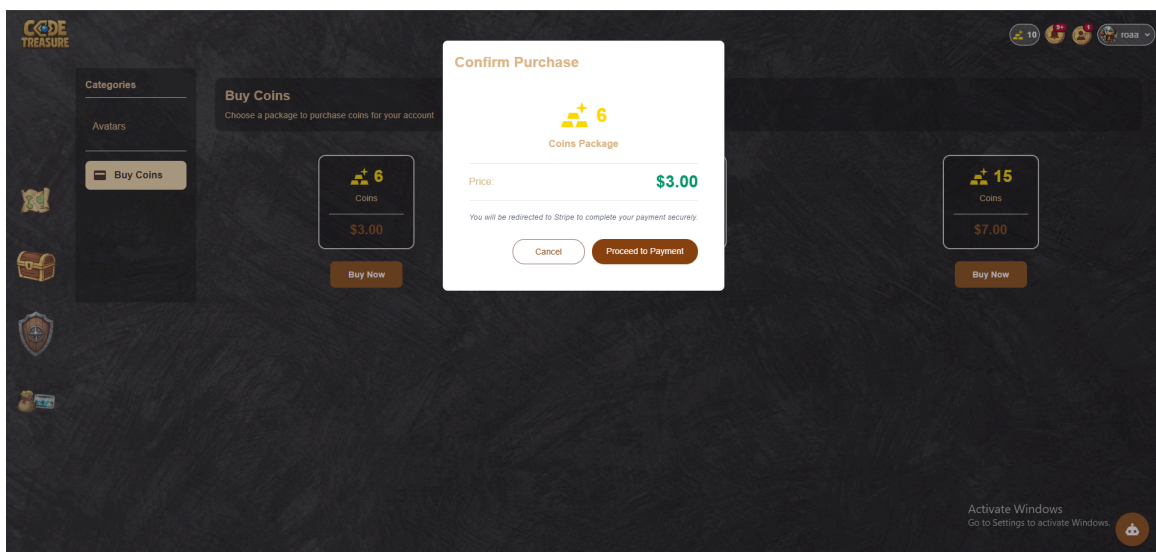
4. Store: The application includes an in-game store where players can purchase avatars using coins. Coins can be earned by completing levels or purchased through **real-money transactions** using the **Stripe** payment system. Stripe is integrated to provide a secure and reliable payment process, allowing players to buy coin packages defined by the admin. Once the payment is completed successfully, the coins are instantly added to the player's balance and can be used inside the store.



(a) Buying avatars



(b) Buying coin package



(c) Confirm Purchase

- Stripe payment

New business sandbox

Choose a currency:

\$9.78 \$3.00

1 USD = 3.6568 SAR

6 Coins Package
Purchase 6 coins for your account

\$9.78

ادفع باستخدام

OR

Pay with card

Email
roaa@mail.com
Did you mean @gmail.com?

Payment method

Card information

4242 4242 4242 4242

01 / 26 123

Cardholder name
Roalhamayel

Country or region
Palestinian Territories

☐ Save my information for faster checkout
Pay securely at new business sandbox and everywhere link is accepted.

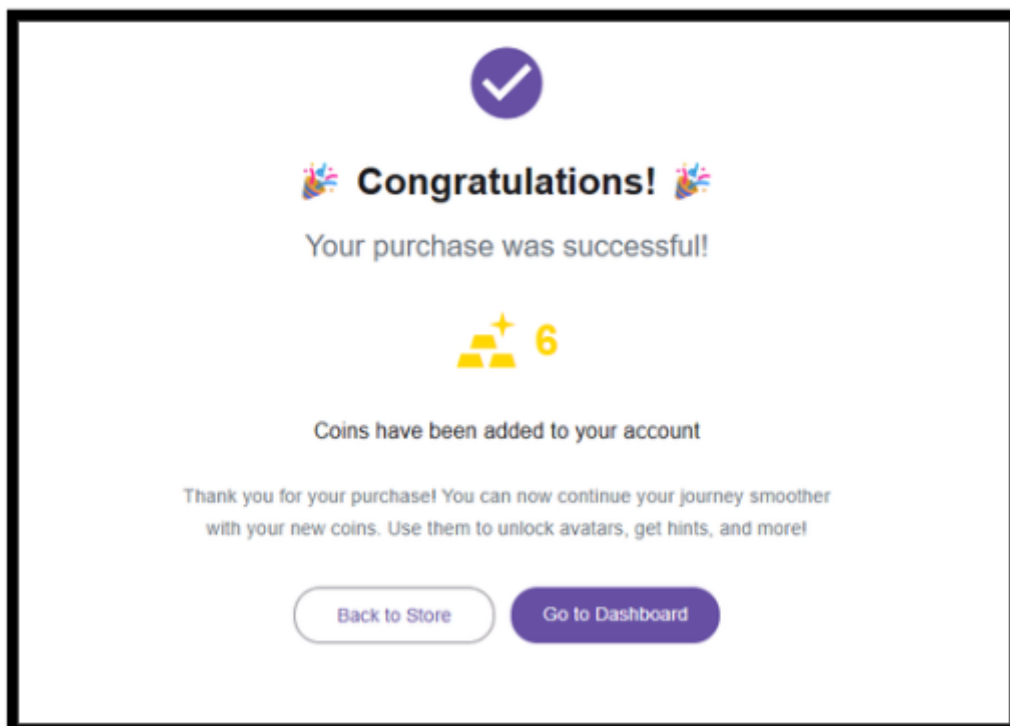
Pay

Powered by Terms Privacy

Activate Windows
Go to Settings to activate Windows.

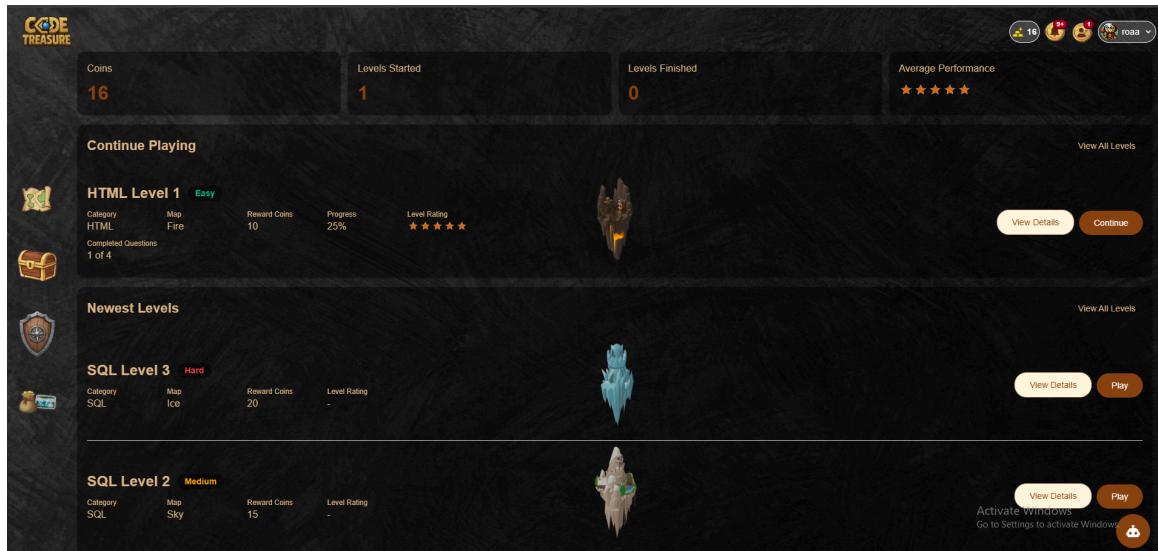
(d) stripe payment

- Successful payment



(e) Successful payment

5. Dashboard: The dashboard displays player statistics, progress, performance, available levels.



(a) Player dashboard

6. Billing and payment: When a player makes a payment using Stripe, the transaction details are recorded in the Billing and Payments section of the system. These details include information such as the purchased package, amount, payment method, and transaction status.

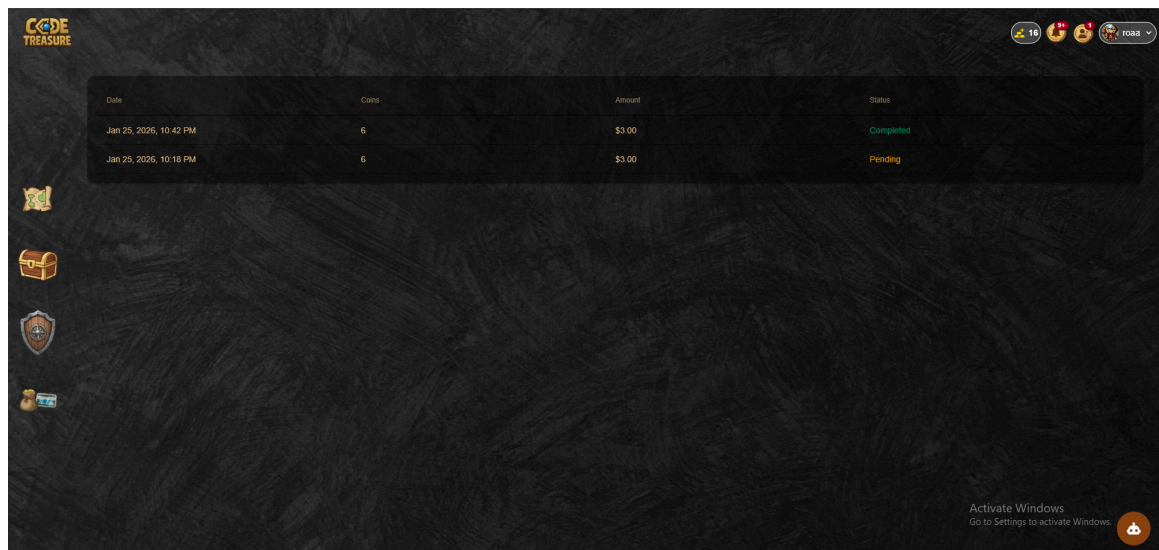
Initially, the payment status is set to Pending. This is because the system relies on Stripe Webhooks to confirm the final result of the transaction. Webhooks are secure event notifications sent from Stripe to our backend to inform us whether the payment has been successfully completed or failed.

Once the webhook event is received and verified, the system updates the payment status accordingly:

Completed if the payment is successful and confirmed by Stripe.

Failed if the payment is declined or not completed.

This approach ensures accurate payment tracking, prevents fraud or duplicate transactions, and guarantees that coins are only added to the player's account after Stripe confirms the payment.



Date	Coins	Amount	Status
Jan 25, 2026, 10:42 PM	6	\$3.00	Completed
Jan 25, 2026, 10:18 PM	6	\$3.00	Pending

(a) Billing operations view

7. Notifications: The notification system in our project is designed to keep users informed about important events in real time. It is implemented using Firebase and Socket.IO to ensure reliability and responsiveness.

Firebase is used to handle push notifications, allowing players to receive notifications even when the application is not actively open. This is mainly used for important events such as new level announcements, admin announcements, and system updates.

Socket.IO is used for real-time in-app notifications, enabling instant communication between the server and connected users without the need to refresh the page. It is used for events such as payment confirmations, new messages, level completion updates, and other live interactions.

By combining Firebase and Socket.IO, the system ensures that notifications are delivered efficiently, whether the user is online inside the application or offline, providing a smooth and engaging user experience.

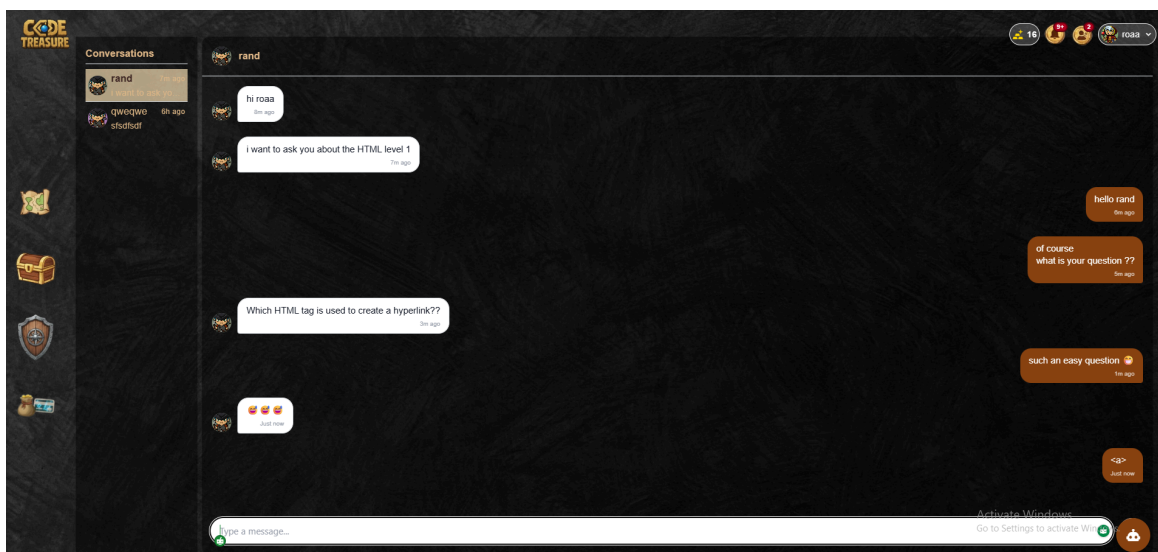


(a) Notification popup

8. Messages: The messaging system enables real-time communication between players within the platform. It is implemented using Socket.IO to allow instant message exchange without reloading the application. Messages are securely stored in Firebase Realtime Database, which ensures message persistence and allows users to view previous conversations. Access to the messaging system is restricted to authenticated users to maintain privacy and security.

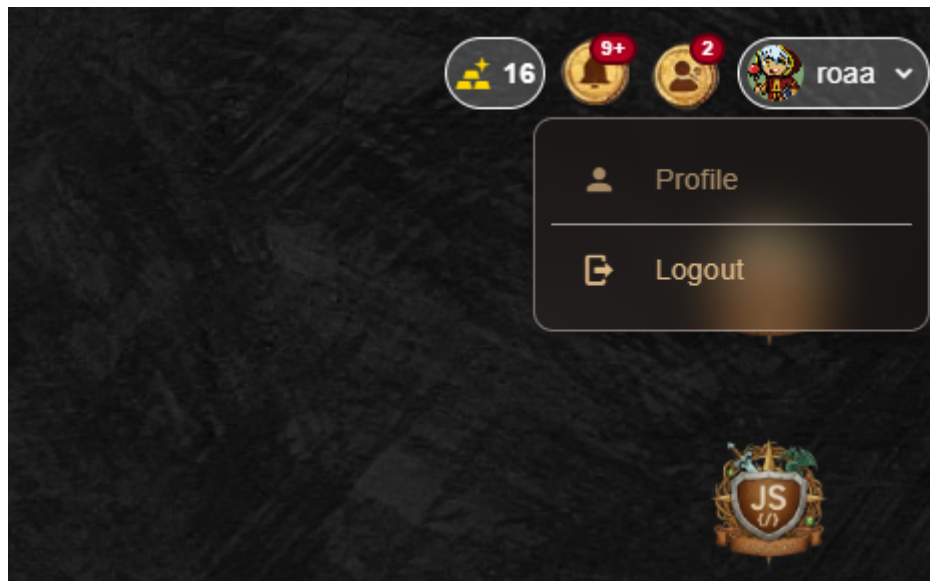


(a) Messages popup

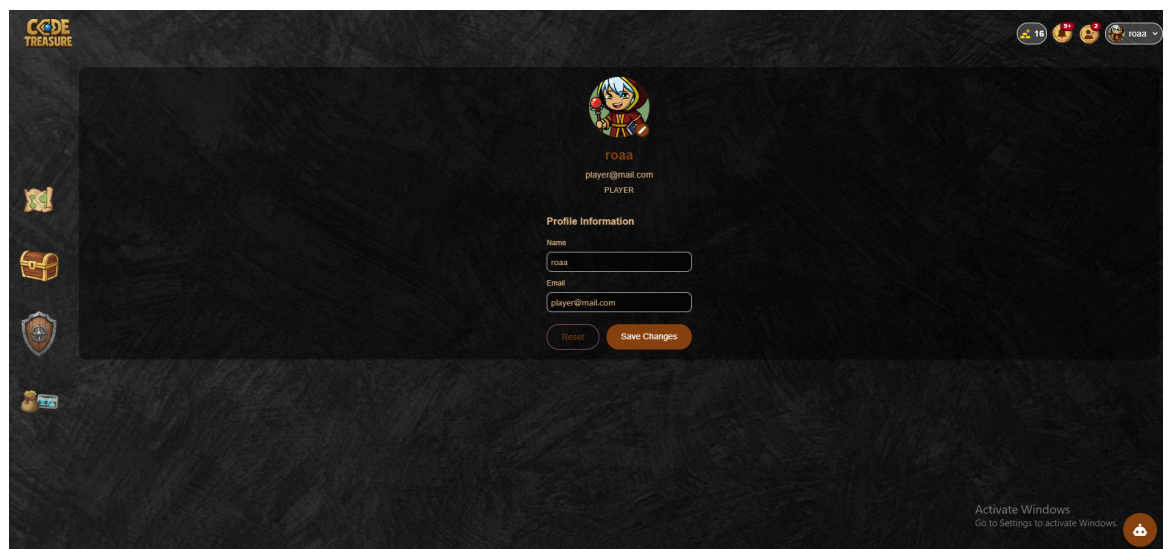


(b) chat

9. Player profile: players can edit their profile data and avatar.

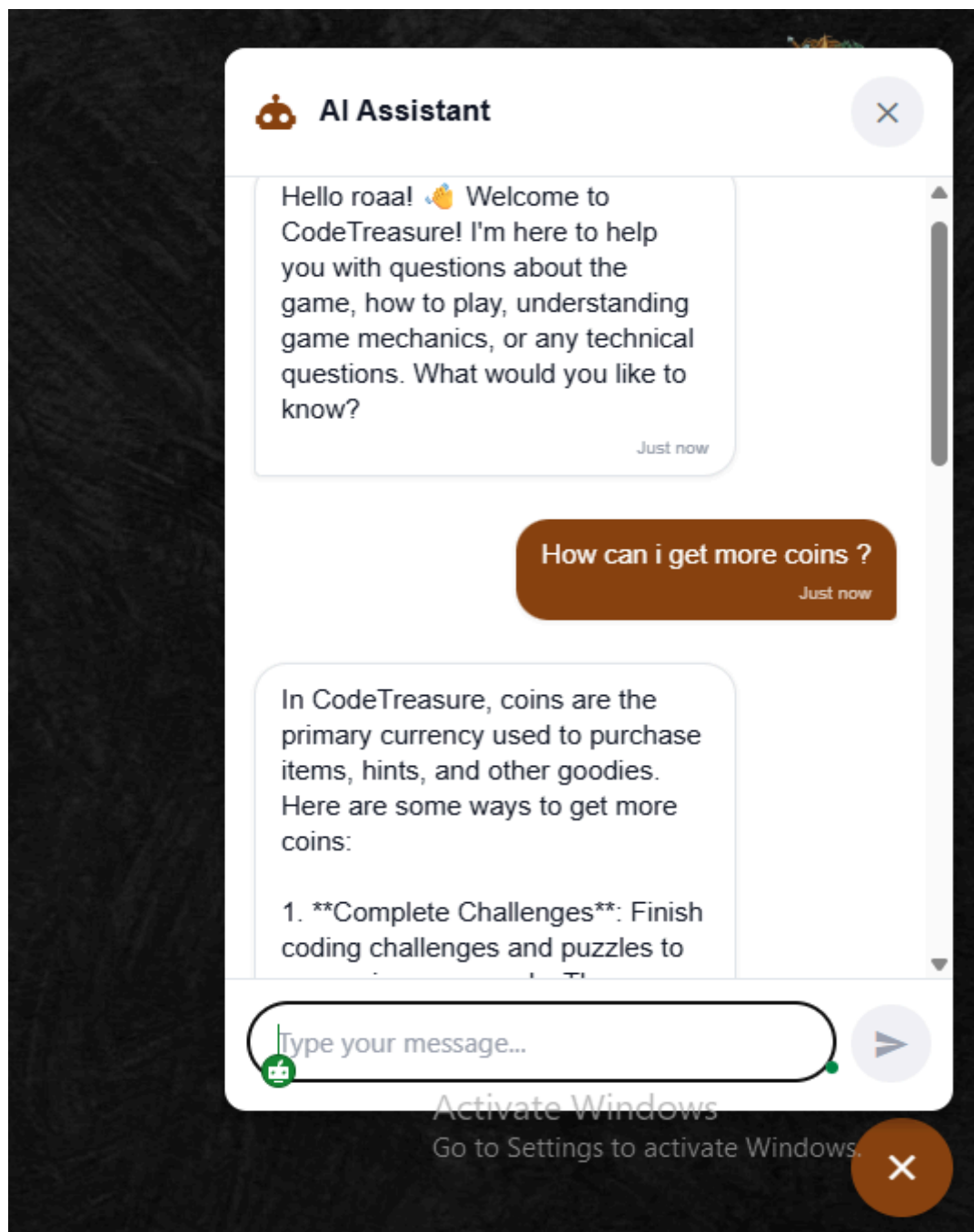


(a) Profile popup



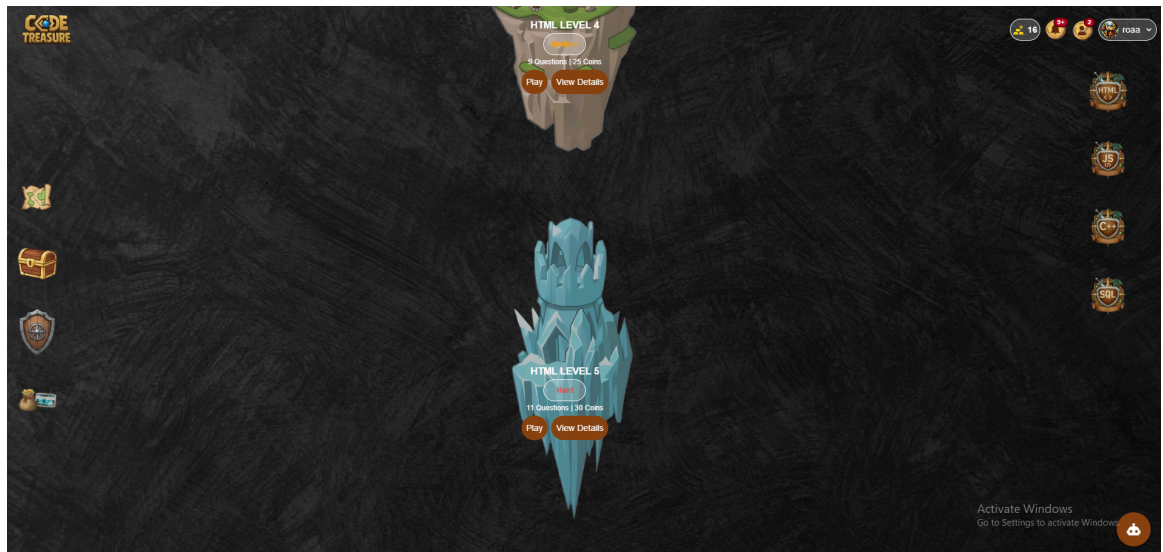
(b) Edit profile

10. AI chatbot: The AI chatbot was integrated into the system to assist players during gameplay and learning. It was implemented using the Groq library, which provides fast and efficient access to large language models. The chatbot was given a predefined context that explains the game mechanics, rules, levels, and learning objectives, allowing it to respond accurately and consistently to player questions. This context-based approach ensures that the chatbot provides relevant guidance, hints, and explanations related specifically to the game, enhancing the overall learning experience without revealing direct answers.



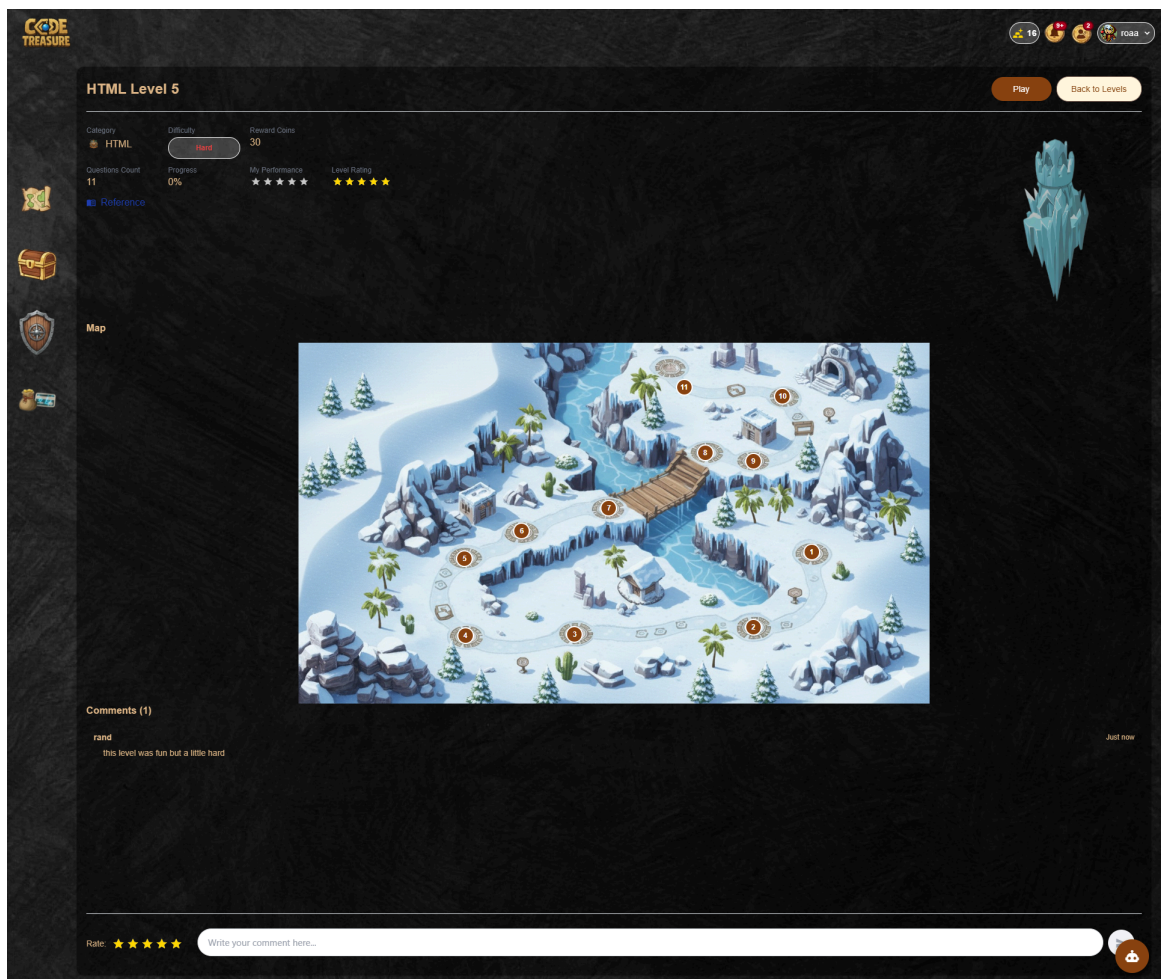
(a) AI chatbot

11. Game flow: level playing flow goes like this



(a) Level selection

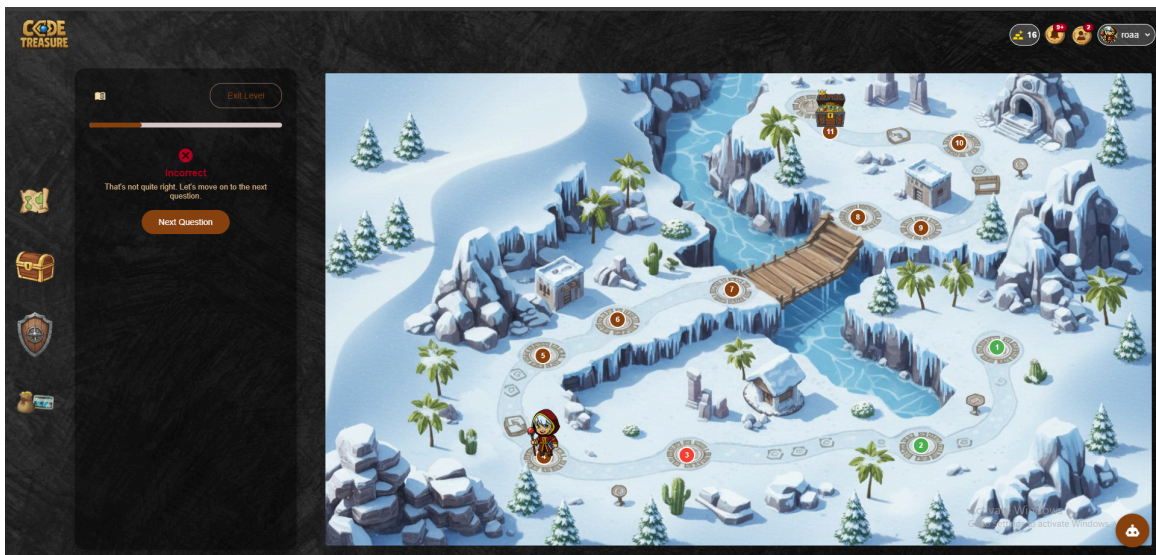
- Players can write comments and rate the level



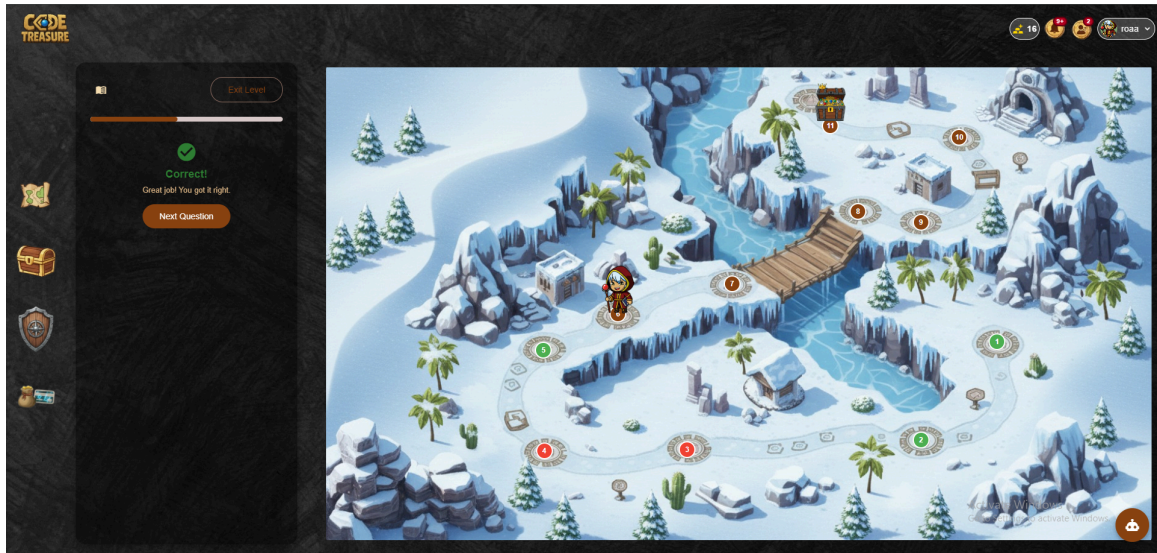
(b) View level details



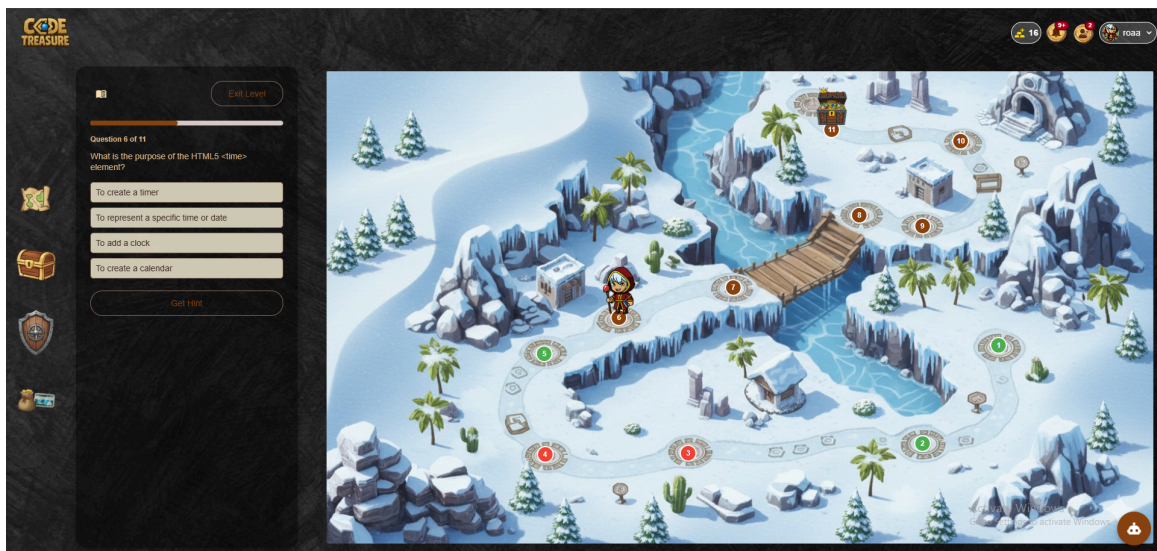
(c) Playing the level



(d) Wrong answer result

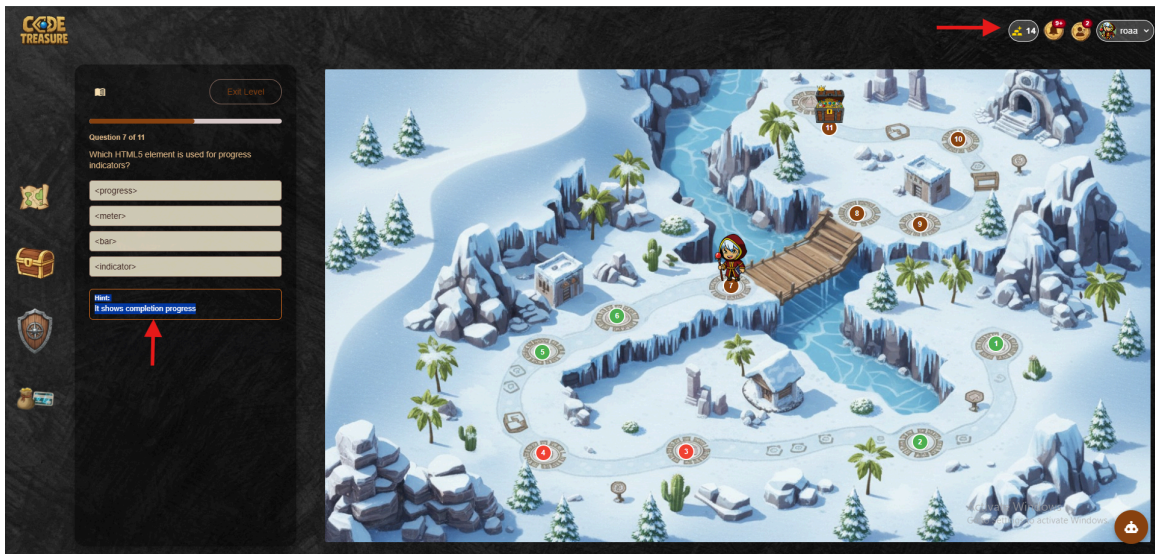


(e) Correct answer result



(f) Question

- When the player uses a hint it costs him 2 coins



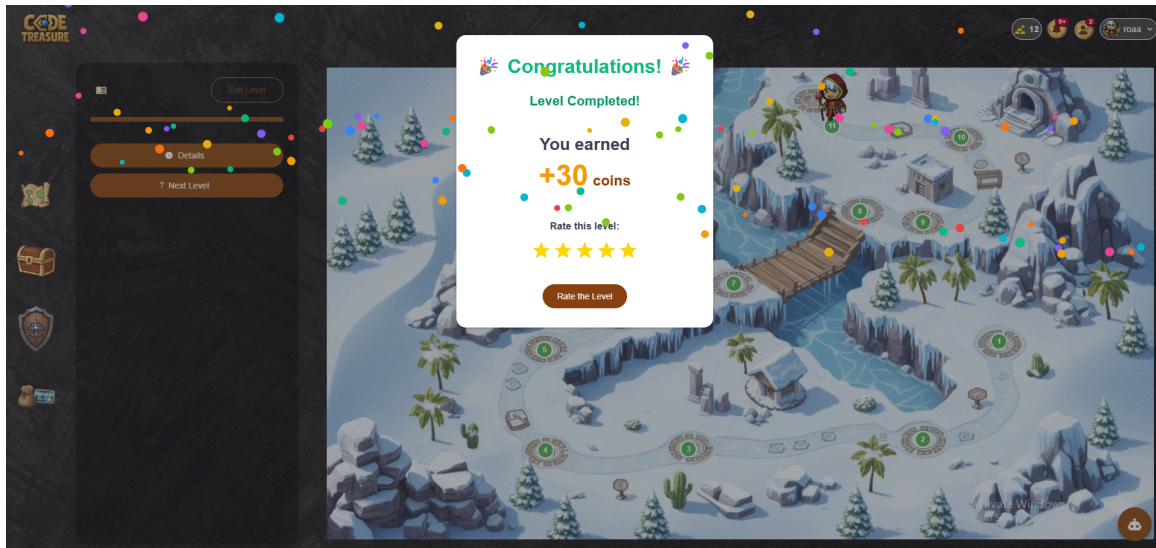
(g) Question hint

- In the end of the level the player can see his status from the colors code
 - ◆ Green circle : correct answer
 - ◆ Red circle : wrong answer
 - ◆ Brown circle : not answered yet
- If the player did not answer all the questions correctly he won't get the treasure and he won't take the level's coins



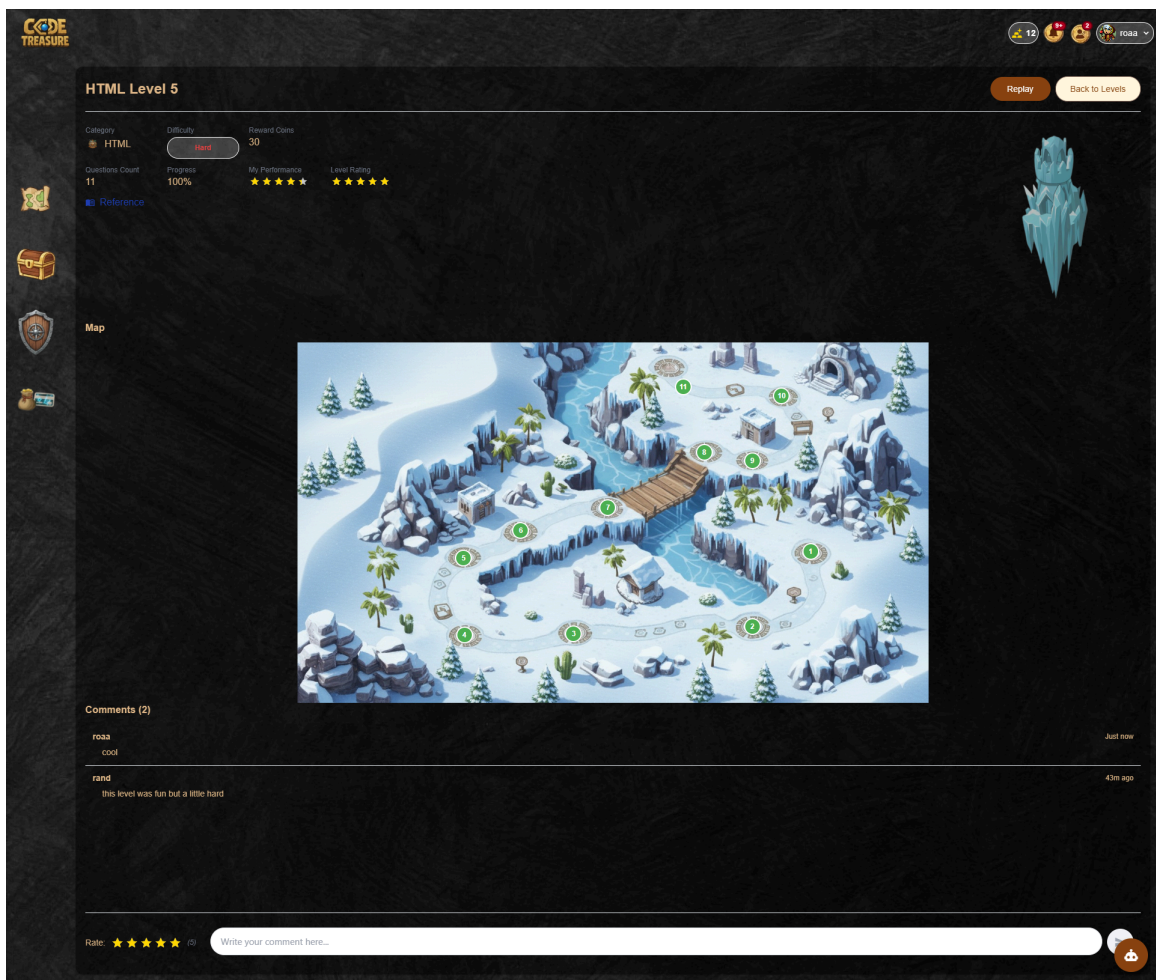
(h) Level result

- When the player answers all the questions correctly he earns the level coins and the treasure



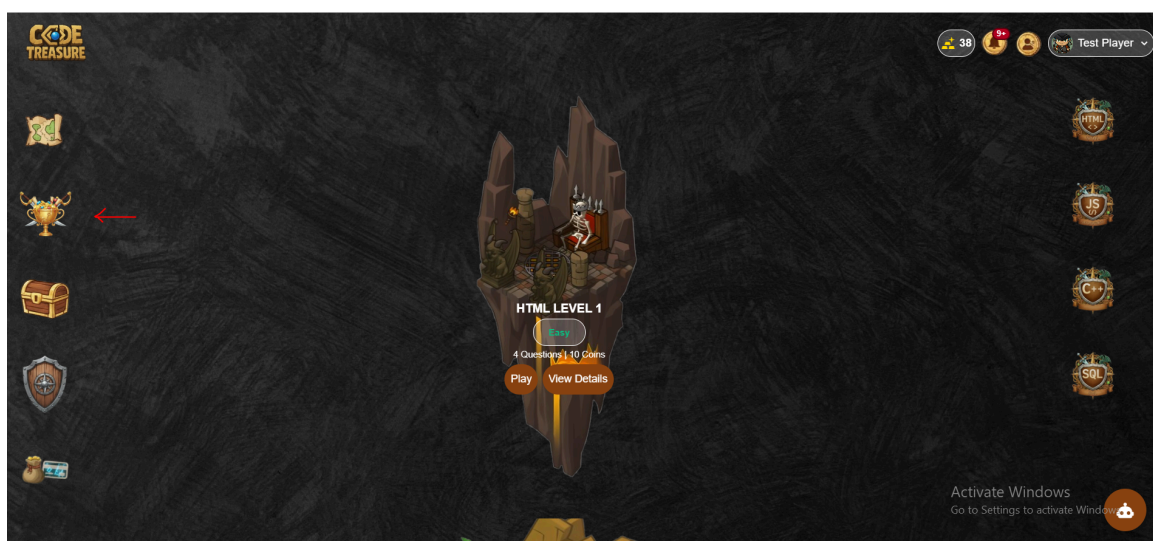
(i) congratulations and rating popup

- **Progress Tracking:** Progress represents the number of questions completed within a level.
- **Performance Evaluation (My Performance):**
Player performance is affected by incorrect answers, time taken, and hint usage. This provides a realistic and motivating evaluation mechanism.
- **Ratings and Comments:**
After completing a level, players can rate it and add comments. All comments are visible to other players.
- **Coins System:**
Players receive the full number of coins defined by the admin upon completing a level. Coins can be used to purchase hints or avatars.

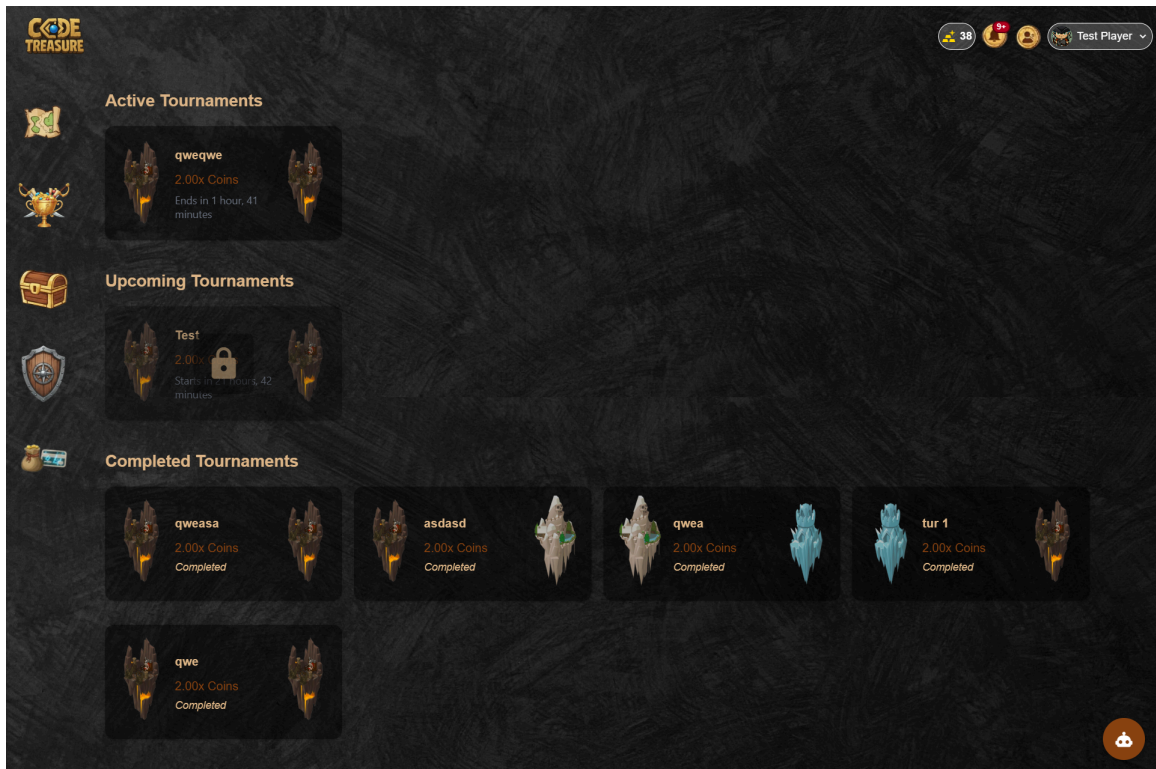


(g) Level details after answering all questions

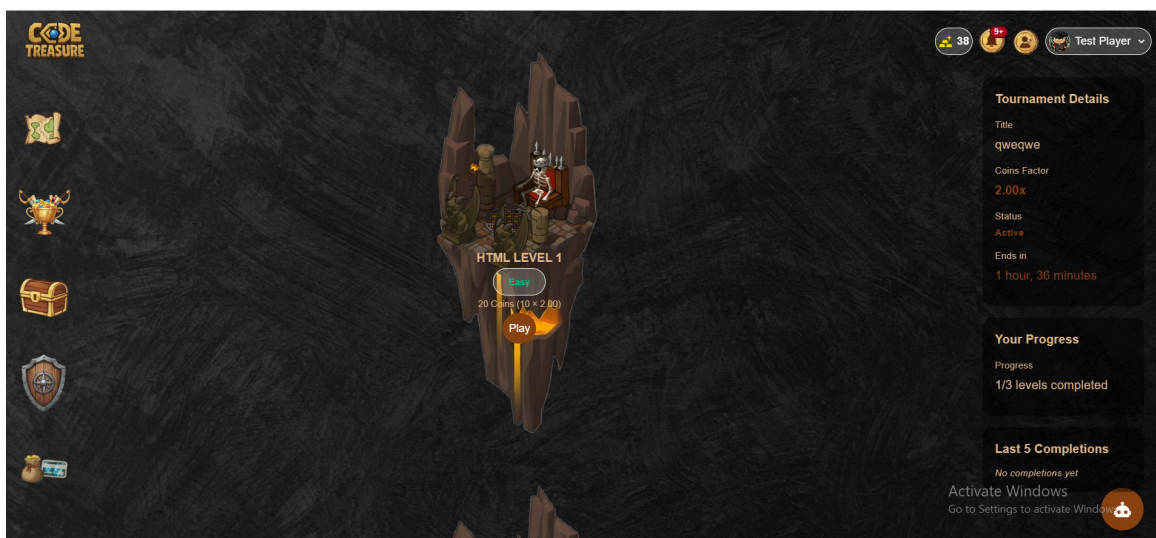
12. Tournament feature:



(a) Tournament feature

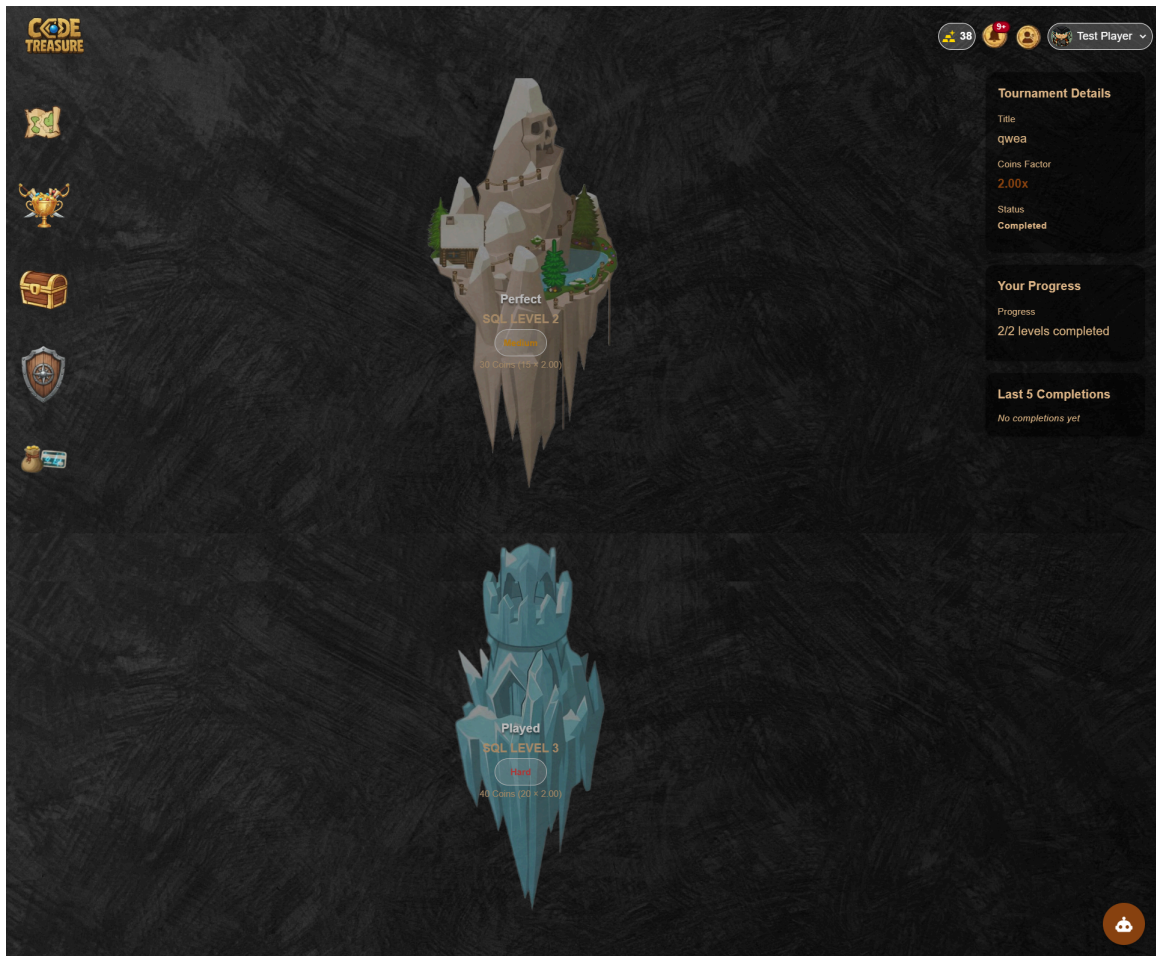


(b) Tournament status



(c) Active Tournament details

- Last 5 completions: means the last 5 players that completed the tournament level successfully.



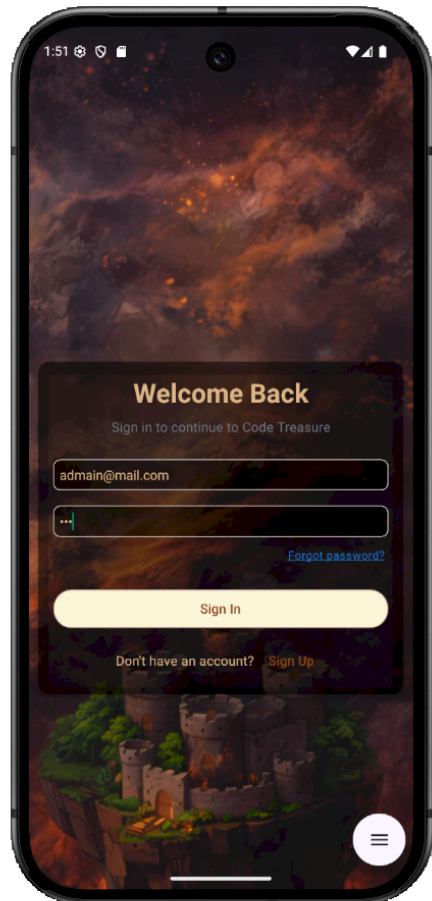
(d) Completed Tournaments

- When the tournaments level is completed the player can not play it .
- Perfect : the player had successfully completed the tournament level.
- Played : the player has played the level but did not answer all questions successfully.

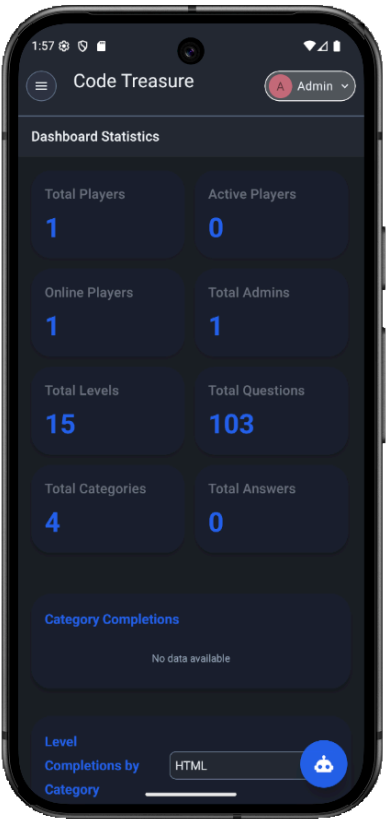
5.3 MOBILE Application

5.3.1 Admin

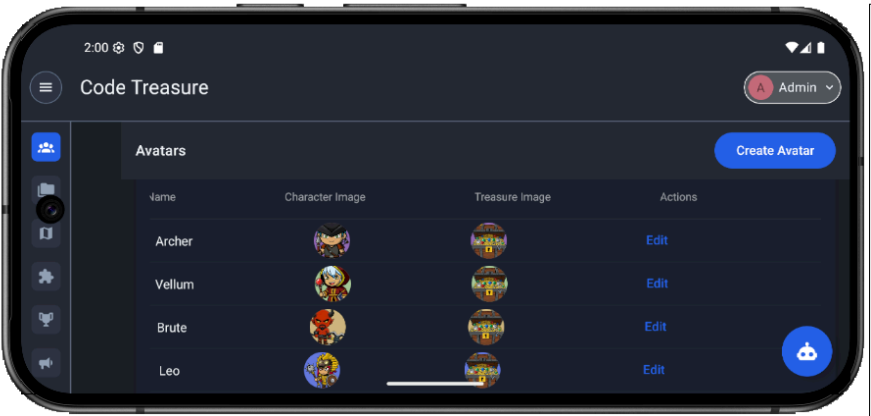
1. Sign in:



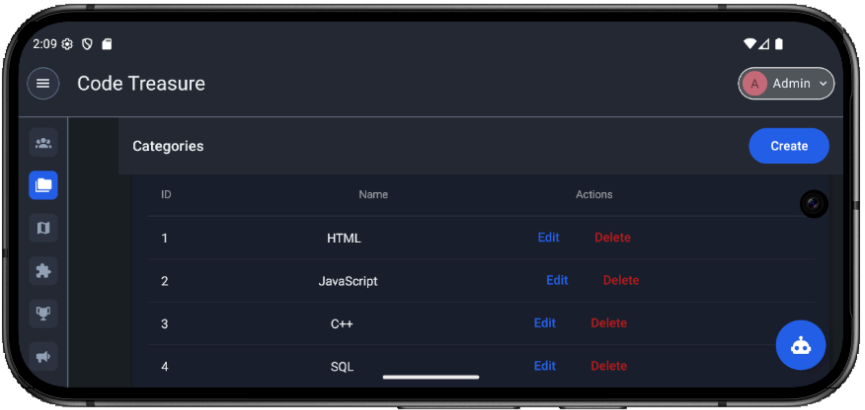
(a) Admin sign in



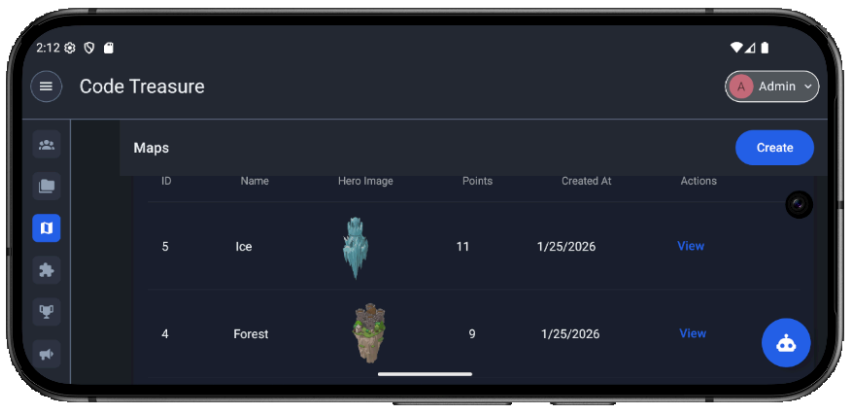
(b)Admin dashboard



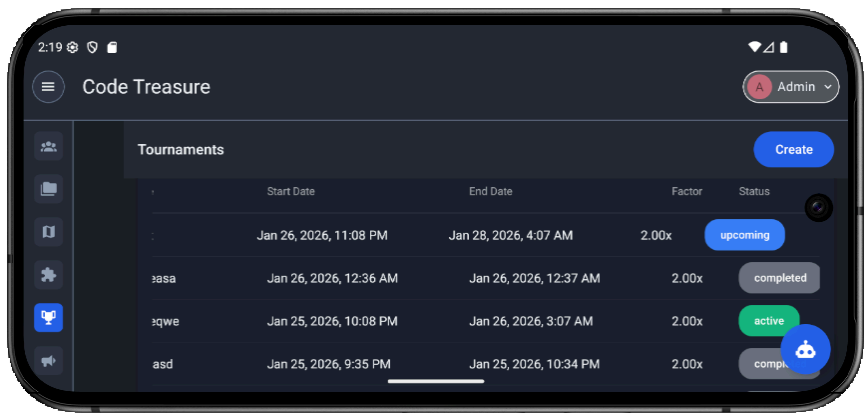
(a) Listing avatar



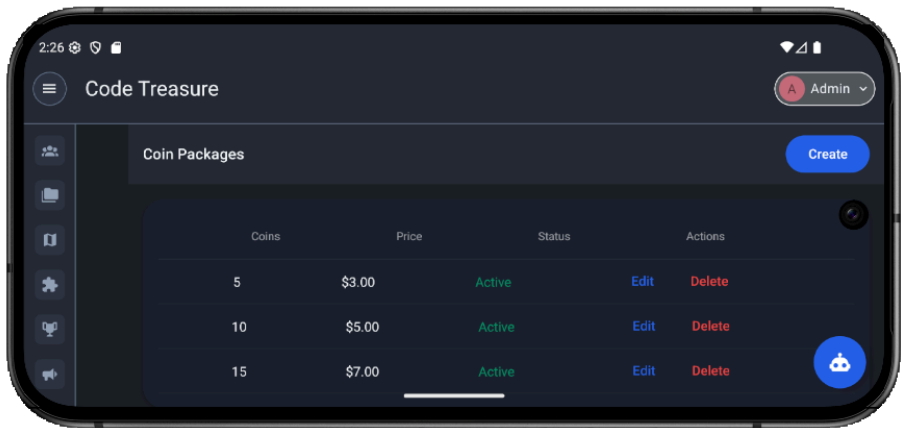
(c) Mobile category management



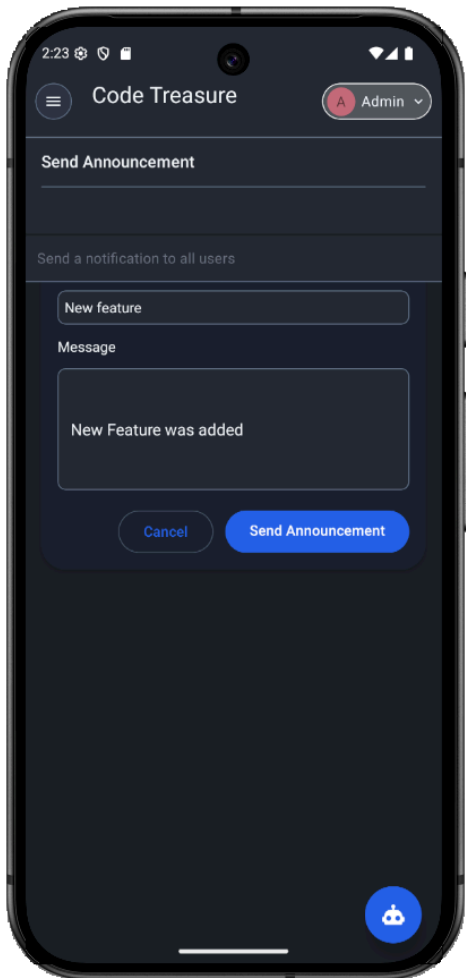
(d) Mobile maps management



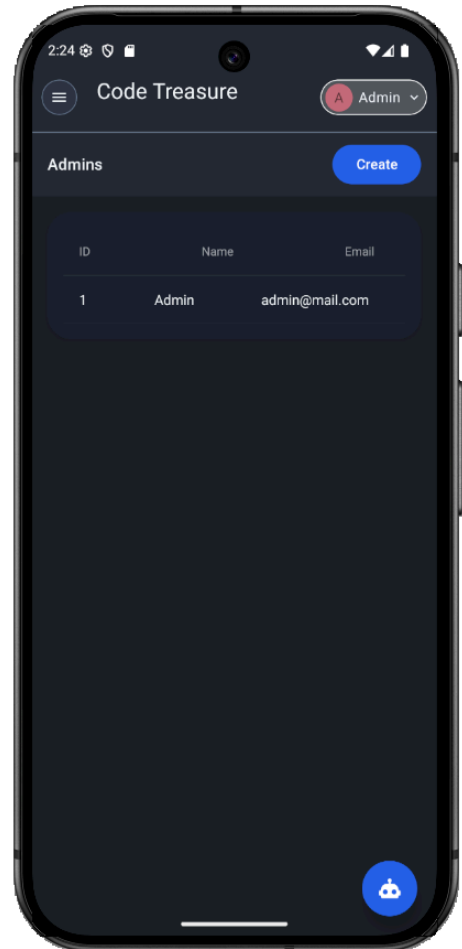
(e) Mobile tournaments management



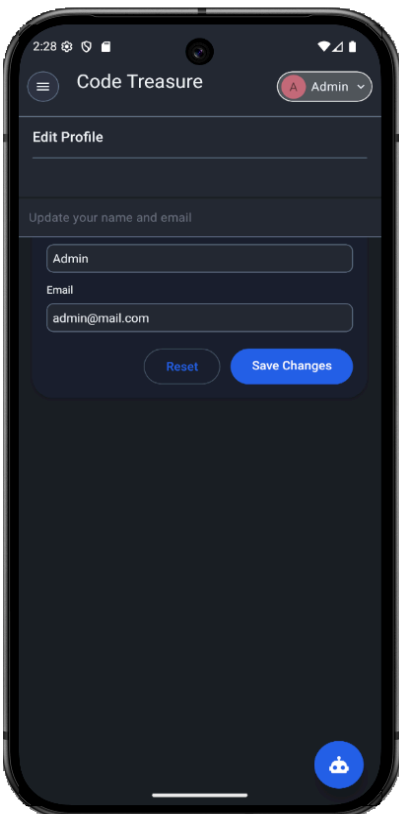
(f) Mobile coin packages



(g) Mobile announcement management



(h) Mobile admin management

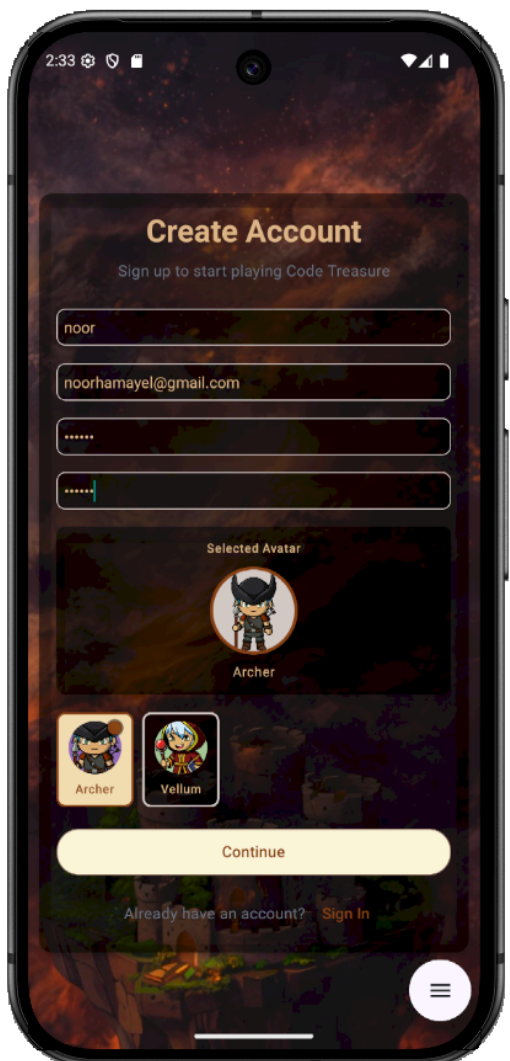


(i) Mobile edit profile

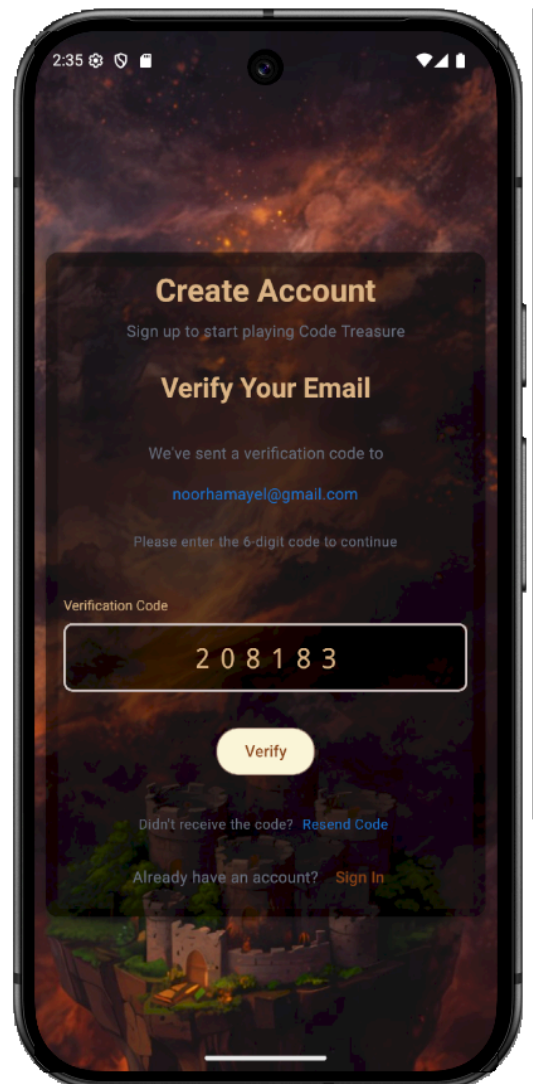
5.3.2 Player

1. Authentication:

(a) Sign up

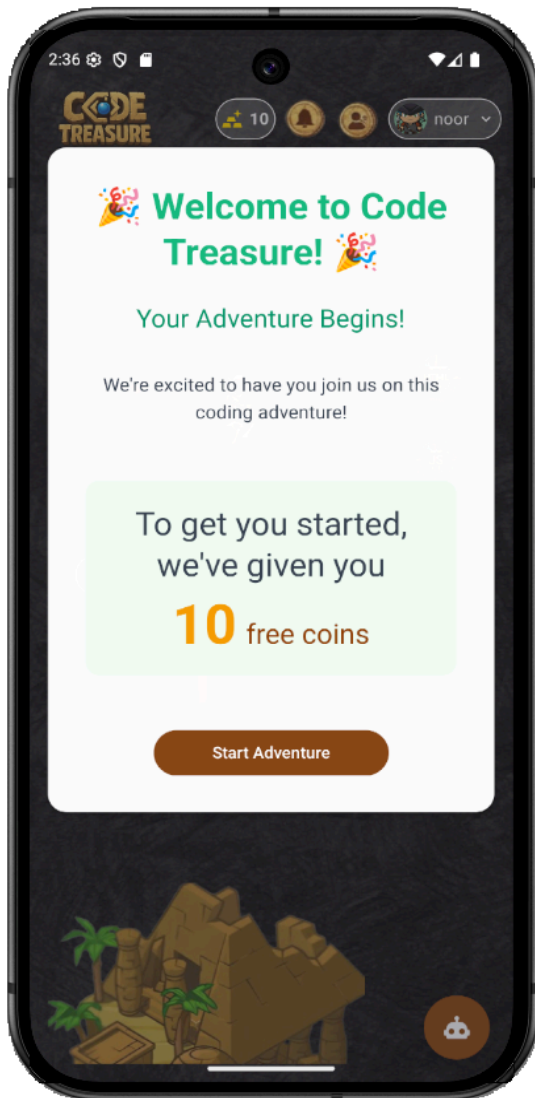


(b) verify your email



2. Levels

(a) Welcome popup



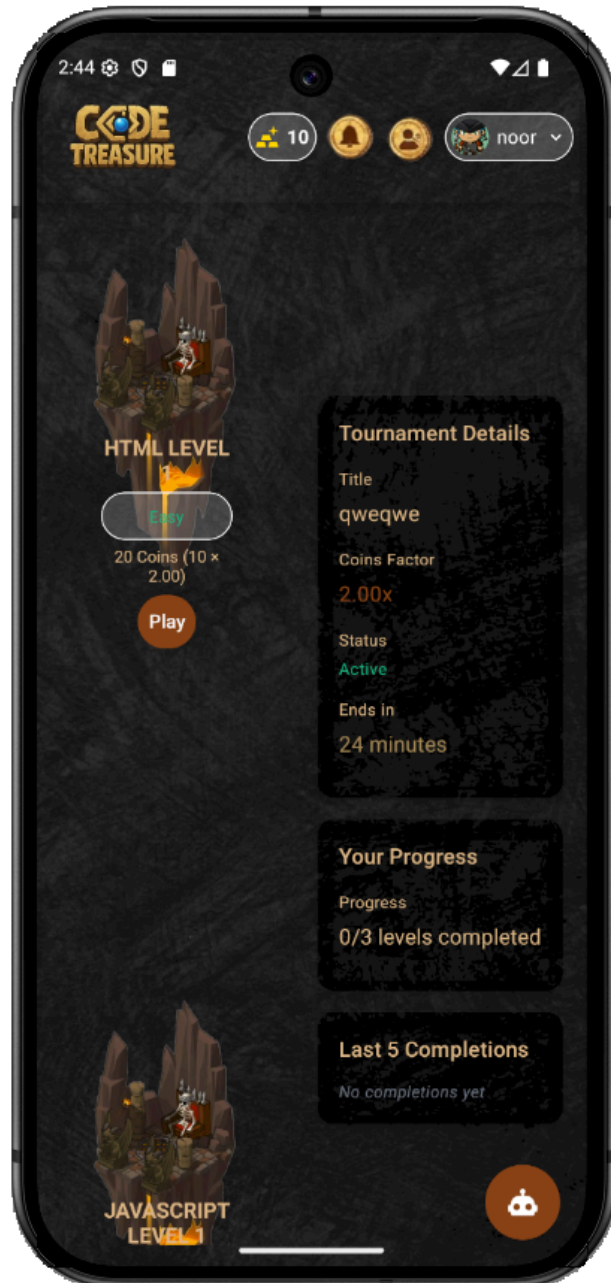
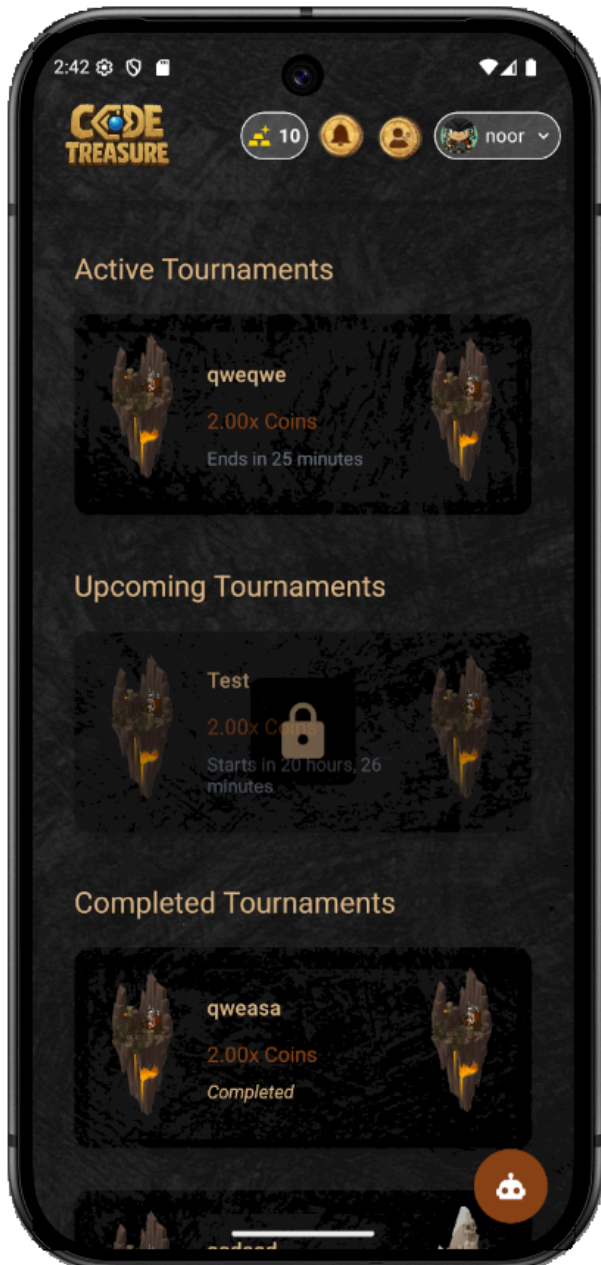
(b) Levels screen



3. Tournament

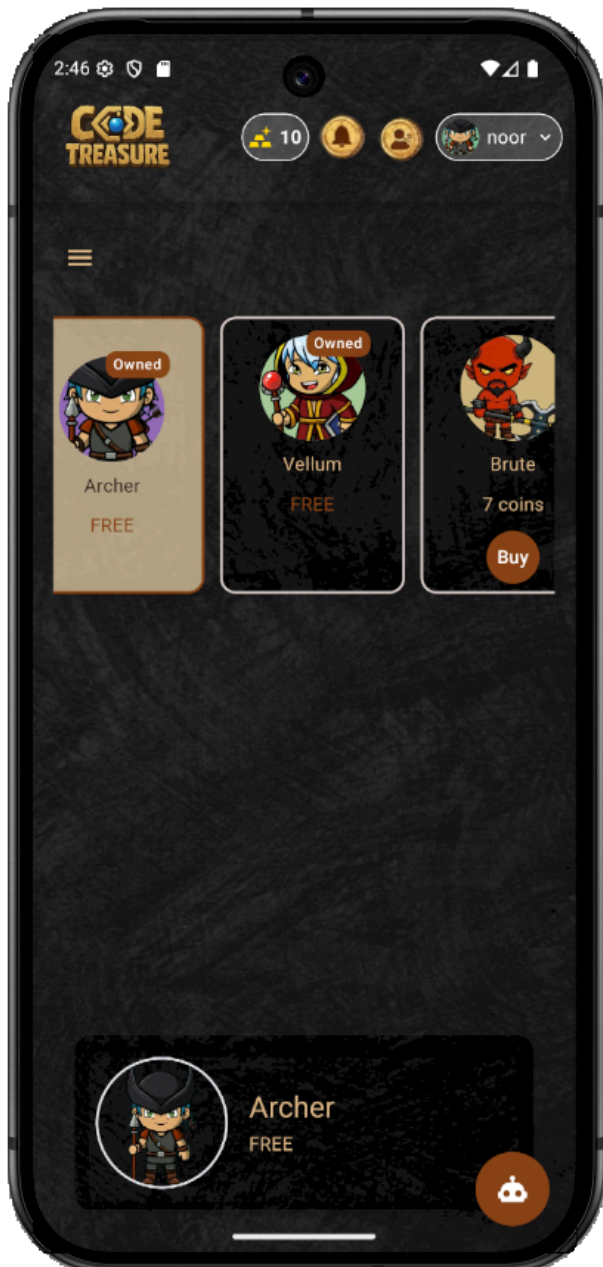
(a) Tournament screen

(b) Active Tournament screen

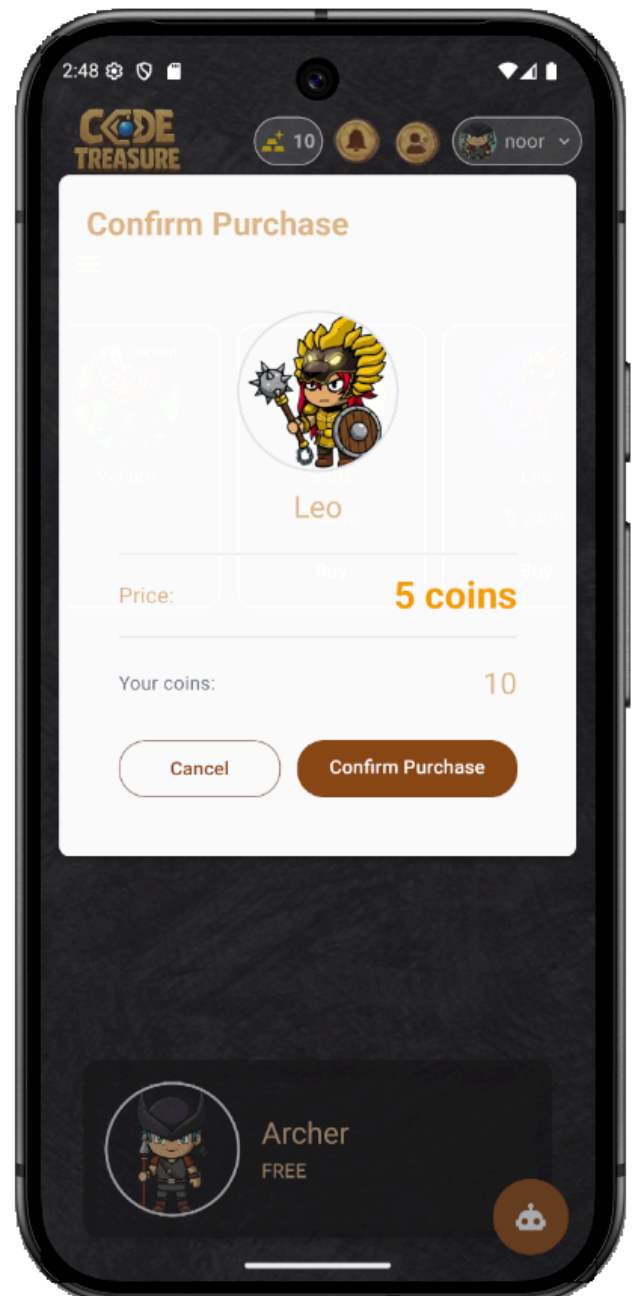


4. Store

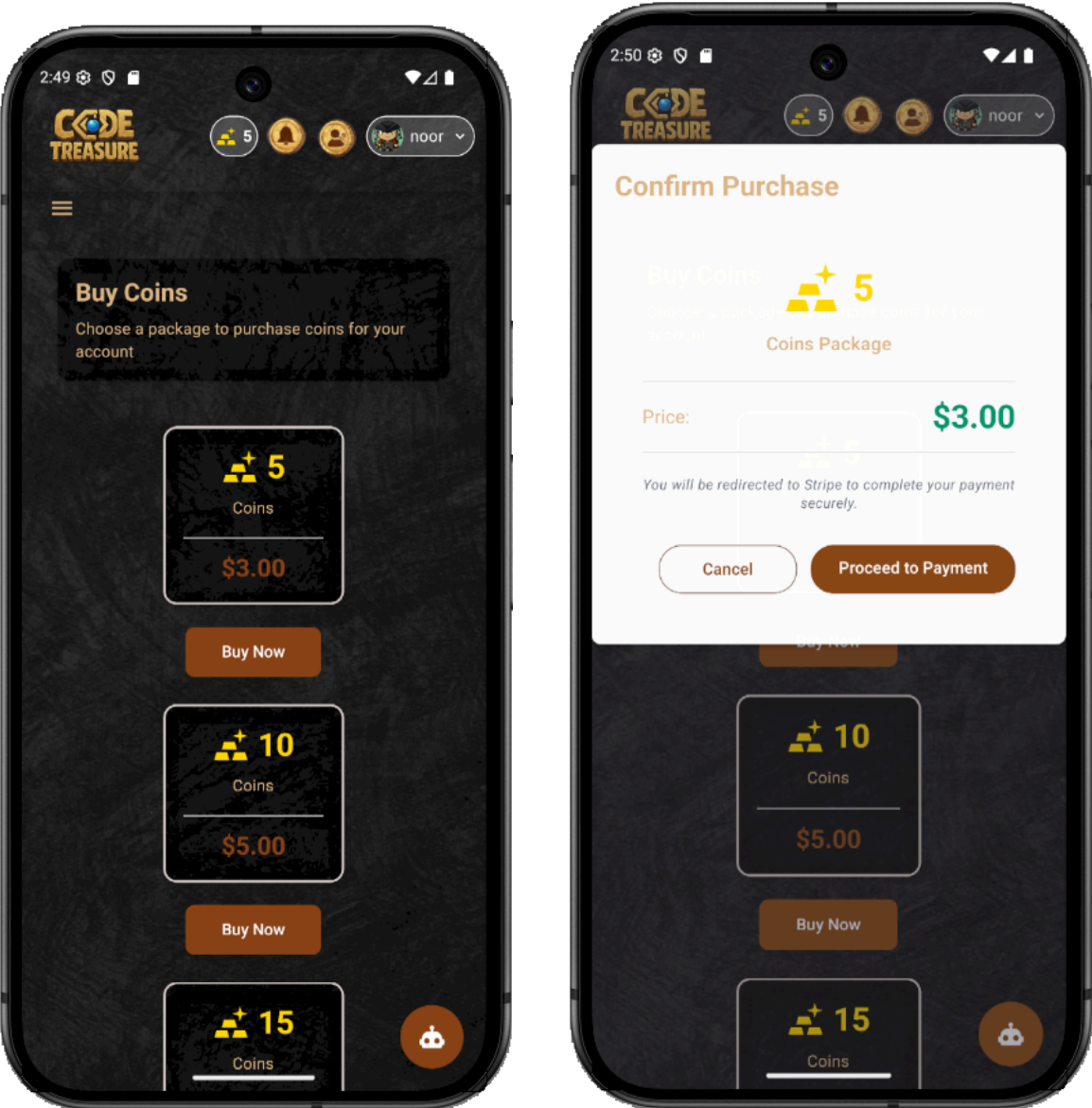
(a) Store screen



(b) confirm purchase screen

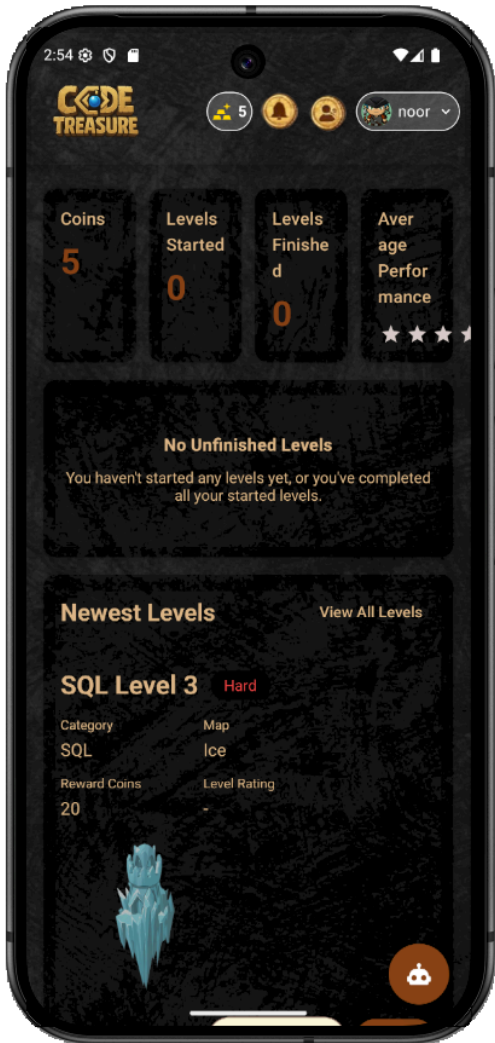


(c) Buy coins



5. Player dashboard

(a) Mobile dashboard



6. Level playing:



Chapter 6

Conclusions and Recommendations

In this chapter, the overall conclusions of the **Code Treasure** project are presented, followed by recommendations and proposed future work to further enhance the system.

6.1 Summary

Code Treasure is a game-based educational platform designed to make learning programming concepts more engaging, interactive, and enjoyable. The project combines learning with gameplay elements such as maps, avatars, levels, rewards, and challenges to motivate players and improve their problem-solving and logical thinking skills.

The platform allows players to learn various programming topics through structured levels and questions while tracking their progress and performance. Real-time features such as chat and notifications enhance user interaction and foster a collaborative learning environment. The admin dashboard provides full control over content management, user monitoring, statistics, and system configuration.

The results of the project demonstrate that integrating educational content with game mechanics can significantly increase learner

motivation and engagement. By transforming traditional learning methods into an interactive experience, Code Treasure successfully achieves its goal of making programming education more accessible and enjoyable.

6.2 Future Works

1. Expanding the in-game store to allow players to purchase full game themes that change the overall appearance of the game, such as backgrounds and colors.
2. Implementing a level progression system where players must complete a level before unlocking the next one.
3. Enhancing the assessment system by introducing different question types, such as code writing, fill-in-the-blank questions, and drag-and-drop activities.
4. Developing an intelligent recommendation system that suggests levels, challenges, or content based on the player's performance, progress, and interests, helping to personalize the learning experience and increase engagement.

References

1. Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., & Edwards, T. (2016). *Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning*. *Computers in Human Behavior*, 54, 170–179.
URL: <https://www.sciencedirect.com/science/article/pii/S074756321500519X>
2. Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*. MIT Press.
3. Kafai, Y. B., & Burke, Q. (2015). *Constructionist Gaming: Making Games for Learning*. *Educational Psychologist*, 50(4), 313–334.
URL: <https://www.tandfonline.com/doi/full/10.1080/00461520.2015.1128332>
4. Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). *Exploring children's learning experience in constructionism-based coding activities through design-based research*. *Computers in Human Behavior*, 99, 415–427.
URL: <https://www.sciencedirect.com/science/article/pii/S074756321930187X>