بسم الله الرحمن الرحيم

An-Najah National University
Faculty of Engineering and Information Technology
Computer Engineering Department

Graduation Project 1

DevHub

Nasr Shaer & Kareem Yqoup

Supervisor: **Dr. Samer Arandi**

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

January 2025

Acknowledgment

We would first want to express our sincere appreciation to the Almighty. Without His blessings that led us every step of the way our route, leading to the successful completion of this project.

We are really grateful to our parents and other loved ones. We have been continuously inspired and uplifted by your constant encouragement along our journey. We have gained strength and stability from your kind and supportive remarks, which have created a solid foundation.

And to our supervisor, Dr. Samer Arandi, we express our gratitude for your guidance and your significant suggestions. The precise insights, insightful remarks, provoking debates, and useful feedback you have provided have greatly improved our project's capability.

We would especially want to thank An-Najah National University's Computer Engineering Department for their continuous support.

Disclaimer

The team is comprised of two computer engineering students, Nasr Shaer and Kareem Yaqop, currently pursuing their education at An-Najah University. Collaborated on the completion of this paper. The authors acknowledge full responsibility for any errors identified within. The ideas, suggestions, findings, and opinions expressed in this paper solely belong to the authors and do not reflect the viewpoint of An-Najah National University.

Table of Contents

Table of Figures

Abstract

Today developers face challenges related to code collaboration, testing, debugging and staying updated with evolving and new frameworks on daily bases. Developer Hub addresses these issues by providing a universal platform that combines most of devs needs like AI-assisted IDE, cloud-based sandboxes and a collaborative bug bounty system.

Developer Hub is a developer-centered platform designed to ease software development process by integrating a cross-language IDE, AI-powered assistance and cloud-based sandboxes. The platform encourages collaboration, knowledge sharing, and rapid testing of frameworks and programming languages. Additionally, it features a bug bounty system for debugging challenges and a GitHub code search tool to learn from real-world examples.

1 Introduction

The "Devhub" project is a developer-centered application that was created using NextJs and NestJs. The name of the project is appropriate given that it is the hub where developers exchange knowledge, find their needs and learn or write using new frameworks without any overhead needed.

This system was designed to ensure that developer has all of his needs inside a single platform, which save us the trouble of having to visit tons of platforms daily to use the tools that we need on daily basis.

Therefore, the platform offers an integrated cross-platform multilanguage IDE, this IDE offers a lot of features starting from lazy load, which saves the user the time of waiting all files to load, instead it loads the project metadata and prioritize the file contents based on the user interactions. Moreover, Devhub has a dev community which allows the developers to share ideas, knowledge or even their projects so others can collaborate and interact with. Other than that, the best way to learn things is by learn from real world examples so Devhub offers `code lens` which enables the developer to search across open-source projects in the Github with many customization.

Devhub provides more than just an environment to write the code, which is also provide a place to run the code on our sandboxes which are hosted on Amazon services, which enable developers to test the new evolving frameworks without any overhead.

2 Constraints, Standards and Earlier course work

One of the difficulties we encountered was learning a new languages and frameworks. We are aware that this will be difficult at first, but with practice, it will get easier. Since we had no prior experience making mobile apps or nextJS framework, learning was necessary to finish this project, and putting what we learned into practice was even more important. It was challenging, particularly because there wasn't enough time. The absence of courses and occasionally not having the necessary skills to study a certain subject, particularly how to work with commonly used tools like Android Studio and certain frameworks like nextjs, nestjs and react-native, as well as the usage of Node.js for backend and Mongoose for MongoDB.

Then we revisited the courses we studied in the previous years, such as database, web development, and critical thinking. Our strategy for developing in this the project started with gathering the most data and necessities needed to finish it. Next, we constructed an initial database, which would allow for further growth or modification as the design progressed according to our desired layout.

First, we began with the design of the backend services as amazon services, and how to handle running and syncing user project, also designing for interactive AI chat that is related to our project. Then we drew a blueprint for our initial database design for the system and we made sure that is flexible design to accommodate changes during this project phase.

3 literature Review

With the evolve of AI models like GPT-4 have demonstrated strong capabilities in code generation, refactoring and test creation (Chen et al., 2021; Bubeck et al., 2023). While AI-generated code can improve productivity sometime human oversight remains essential due to errors and hallucinations (Ji et al., 2023). GPT-4 enhances code quality by reducing syntax errors and improving maintainability (Fowler, 2019), but its ability to generate reliable test cases is still limited, often requiring manual validation (Wei et al., 2023). The role of AI in coding continues to grow, but ensuring accuracy and reliability remains a key challenge. [1] [2]

Cloud computing has transformed software development by offering scalable, cost-effective solutions (Erl et al., 2013; Rittinghouse & Ransome, 2016). Service-oriented architectures, particularly SaaS, enhance software accessibility while reducing maintenance efforts (Palos-Sanchez et al., 2017). The rise of online IDEs, such as AWS Cloud9 and Eclipse Orion, enables browser-based coding with real-time collaboration and automated debugging (Yulianto et al., 2017; Ghorashi & Jensen, 2016). However, challenges like security risks and resource limitations persist (Fan et al., 2019; Goldman et al., 2011). Further research is needed to optimize online IDEs for large-scale development. [3]

Cloud IDE is transforming software development by enabling real-time collaboration and remote access. While these platforms offer ease of use and flexibility, challenges like execution speed and security remain. JavaScript is the most widely used language in online IDEs, and technologies like microservices and Docker help optimize performance. By leveraging these advancements, cloud-based development can become more efficient and secure. [4] [5]

4 Methodology

4.1 System Design:

To create web and the mobile app, a variety of tools and programming languages are needed for mobile application development. This chapter will include a list of the tools, programming languages, and technologies that were used to build our project.

4.1.1 Database:

We choose to use a NoSQL database, MongoDB, for storing and managing our application data because of its many benefits that meet our needs. MongoDB is a popular document-oriented database that stores information in BSON (Binary JSON), a robust format like JSON. Because it enables us to store data with different forms, this structure is very useful for our application because it is perfect for handling a variety of information.

The dynamic and changing data requirements of our application are well suited for MongoDB because of its prowess in managing substantial amounts of unstructured or semi-structured data. The essential flexibility of the data model in our application is reflected in its schema-less architecture, which makes it simple to react to changes in data structure. The main benefit of MongoDB is its intuitive handling of relationships and nested structures. This is important because in our application, entities may have intricate interactions, and we need to be able to express these relationships effectively. MongoDB is a document-based database. MongoDB's support for horizontal scalability is essential for accommodating the expected growth in data size with an increasing number of users. It employs sharding, enabling seamless distribution of data across multiple servers, ensuring high availability and performance even as the dataset expands.

In terms of analytics and reporting, MongoDB offers powerful aggregation framework and indexing capabilities. These features empower us to perform complex queries, aggregations, and analyses efficiently, supporting our application's requirement to provide the hotel owner with insightful business analytics.

In summary, MongoDB, with its flexible schema design, scalability, and powerful querying capabilities, emerged as the optimal choice for our application. As our user base grows and data complexity increases, MongoDB provides a robust foundation to meet the evolving demands of our application.

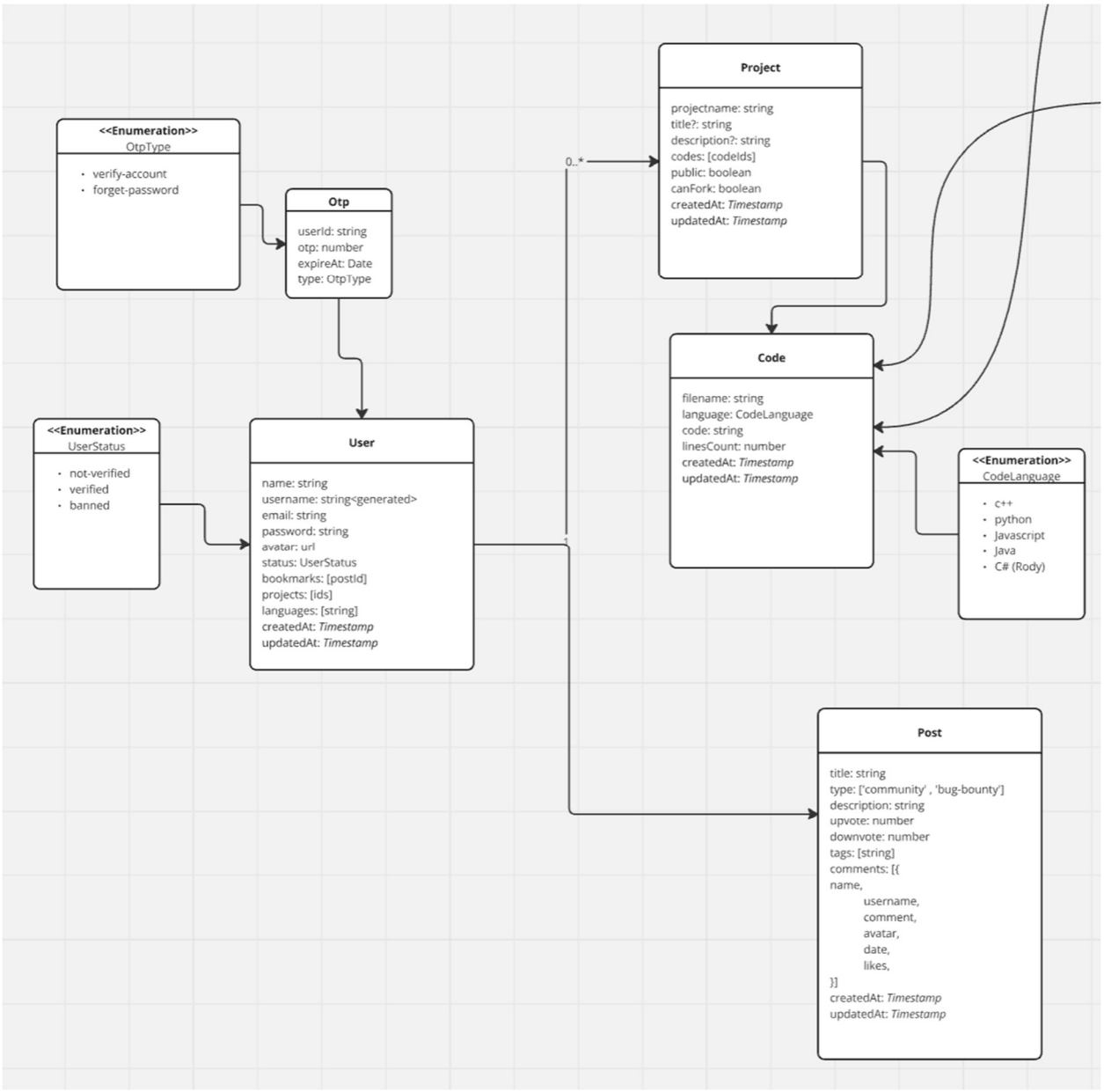

Figure 1 Database Design

4.1.2 Server side:

We decided to use the robust and NextJs for our application's server-side development. NestJs is built on top of ExpressJs which is built using Node.js which is an event-driven, non-blocking environment that makes it ideal for developing scalable network applications. In this case, the server-side functionality was handled by Node.js, which allowed for easy HTTP request connection with our MongoDB database. Given its effectiveness in managing asynchronous processes and real-time applications, Node.js is a great fit for our application. Node.js satisfies these needs precisely because our server-side processes entail interactions with a NoSQL database like MongoDB, which frequently depends on asynchronous operations.

Node.js's capacity to manage several concurrent connections with minimal overhead is one of its main features. Ensuring maximum performance and responsiveness is vital for our program, especially as our user base grows.

The dynamic community and package manager npm (Node Package Manager), which gives users access to a vast array of libraries and modules, are other advantages of Node.js. We can effectively manage our codebase and use different function modules for data representation thanks to the broad library support, which guarantees flexibility and extensibility in our server-side implementation. We can also use JavaScript on the client and server sides with Node.js, which helps to maintain consistency throughout our software. This single language strategy improves code maintainability, lowers the overhead of context change, and streamlines development.

Nest is a framework for building efficient, scalable Node.js server-side applications. It uses modern JavaScript, is built with TypeScript (preserves compatibility with pure JavaScript) and combines elements of OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming).

Under the hood, Nest makes use of Express, but also provides compatibility with a wide range of other libraries, like Fastify, allowing for easy use of the myriad of third-party plugins which are available.

4.1.3 AWS Architecture for Integrated IDE with Sandbox

The architecture of our integrated IDE with a sandbox environment is designed to handle and execute user-submitted different projects within a secure and scalable AWS infrastructure. The process begins when a user clicks the "Run" button on the frontend which sends a request to our NestJS backend hosted on Render. The backend first retrieves project metadata, including the container ID, project framework, and container image from MongoDB Atlas. This metadata is crucial for determining the appropriate execution environment for the user's project and the.

Once the metadata is retrieved, the backend requests an EC2-based container in the ECS (Elastic Container Service) cluster. Then, the container is initialized using a pre-configured image stored in Amazon ECR (Elastic Container Registry). After that, the container is launched, it mounts the user's project files from Amazon S3 allowing the containerized application to access and execute the project's source code based on the specified language and framework.

During execution, all logs and output generated by the project are captured and stored in Amazon CloudWatch Logs. CloudWatch then streams the execution output back to the backend via CloudWatch Log Subscriptions, enabling real-time monitoring and debugging. Finally, the backend forwards the output to the frontend which allows the user to view the execution results directly in the IDE interface.

This architecture leverages AWS services to ensure scalability security and reliability. Amazon ECS provides a flexible and containerized execution environment while the Amazon S3 efficiently manages and stores project files. CloudWatch Logs ensures smooth log management and streaming, enhancing real-time feedback for users. By integrating these AWS components, our IDE sandbox offers a seamless and efficient experience for executing React projects in an isolated and scalable environment.

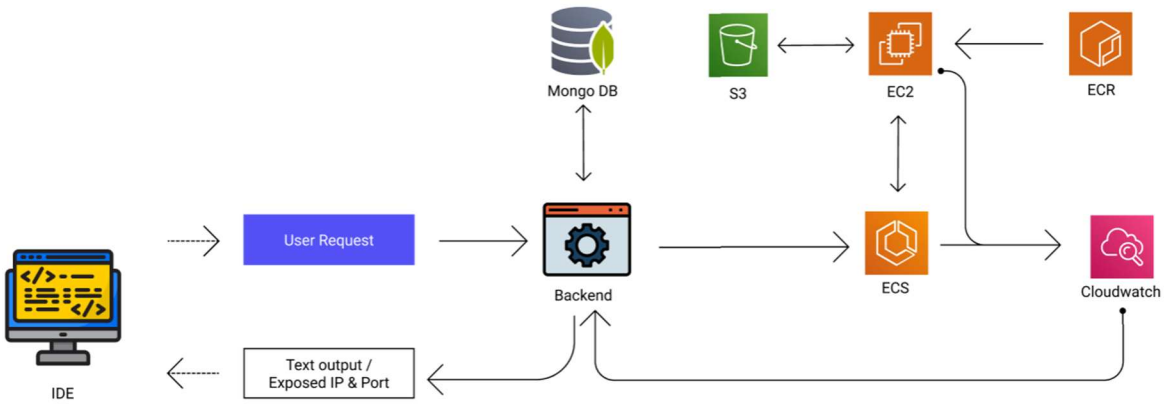

Figure 2 AWS Architecture

4.1.4 UI design:

In order to create a modern sleek design, we have used multiple libraries and frameworks, starting with tailwind which is an open-source CSS framework. Unlike other frameworks, like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables. Instead, it creates a list of "utility" CSS classes that can be used to style each element by mixing and matching.

Moreover, we have used different libraries built on top of tailwind like shadcn, magic-ui and aceternity ui. All of these to provide modern look, flexible and responsive design for all our users.

For the mobile application, it was built using expo which is an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React. Also, using native wind which is a tailwind version for react-native.

We've used multiple designing tools also like figma, miro and excalidraw to draw the basic wireframes of our UI interface for both mobile and web version of DevHub.

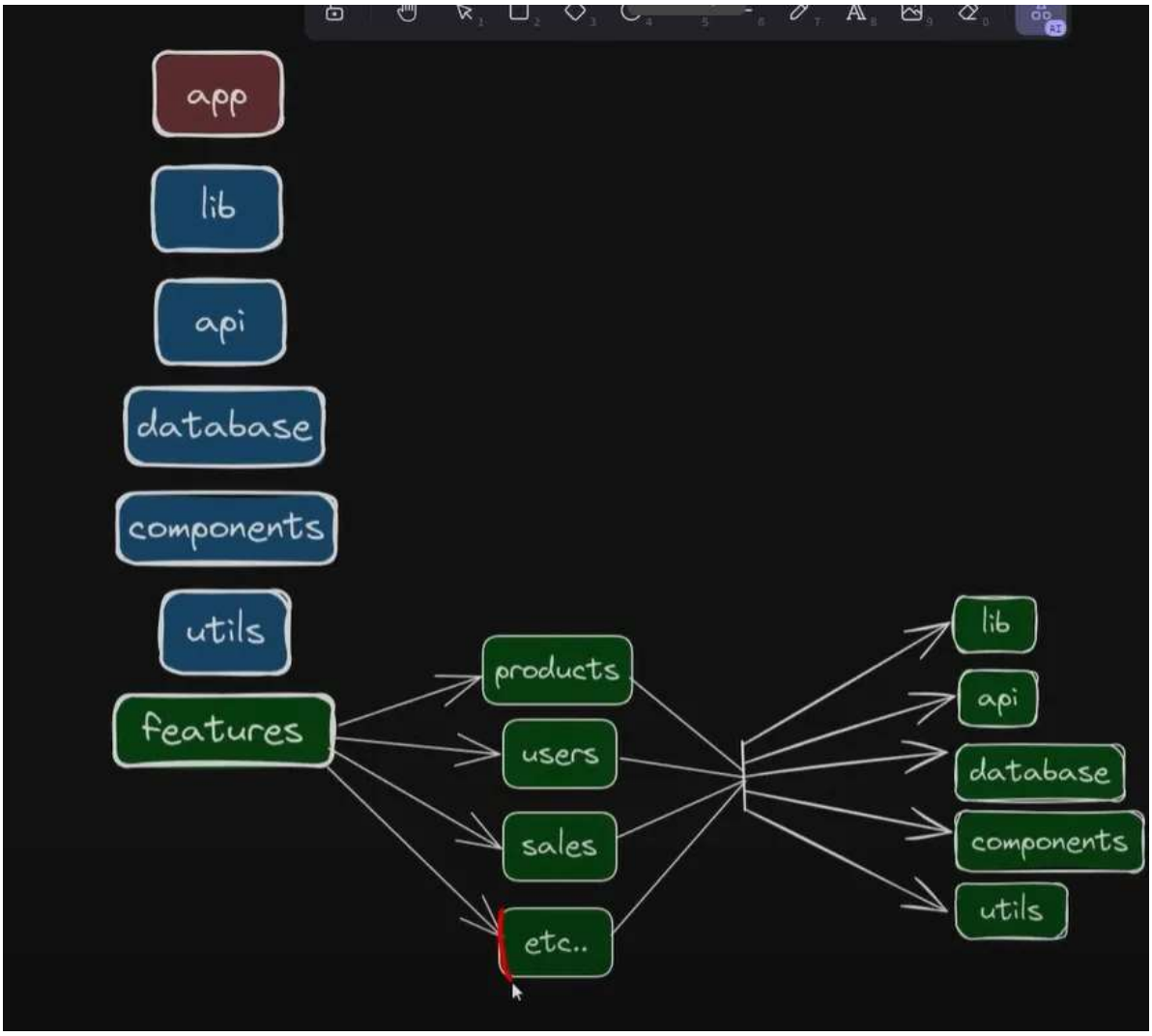

Figure 3:Front-end Architecture

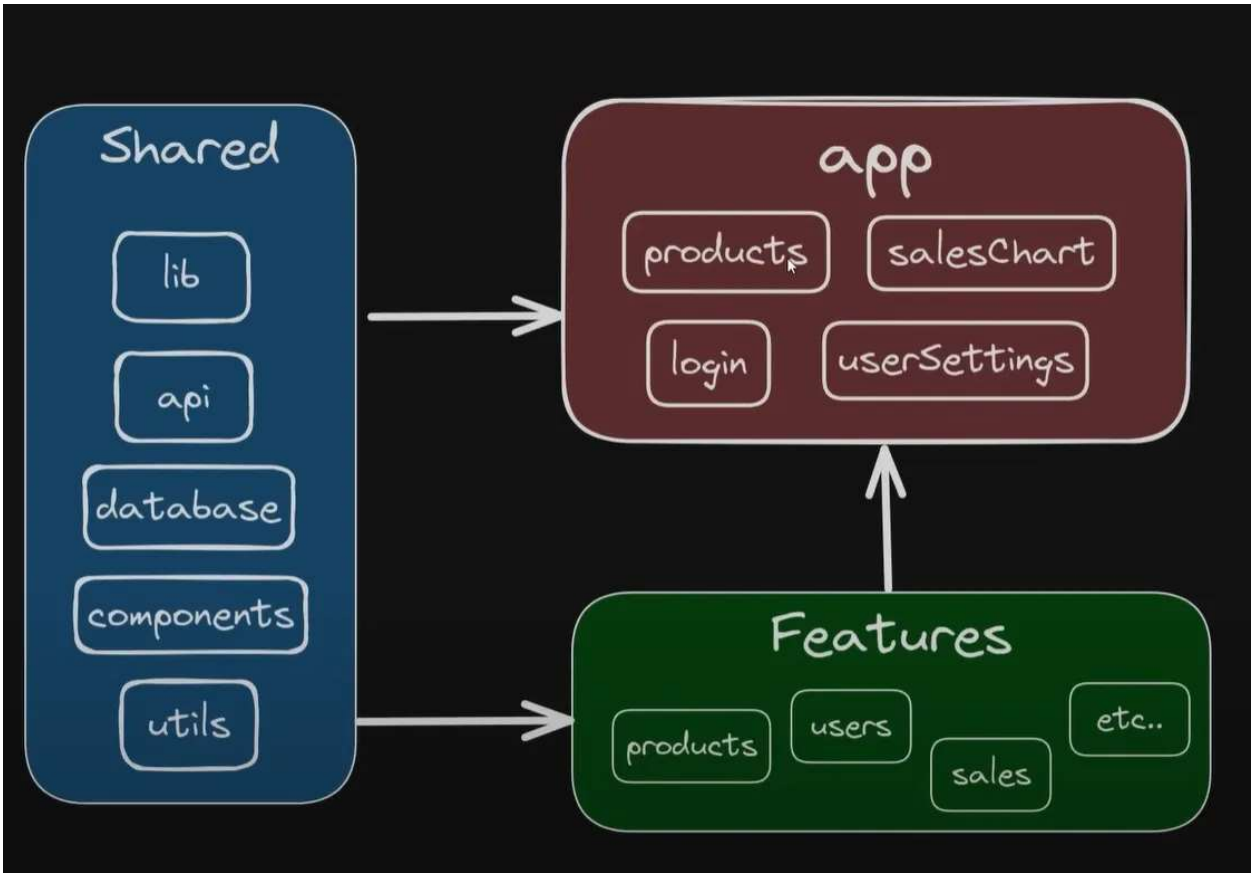

Figure 4 Frontend Architecture

4.1.5 Frontend Application Architecture:

In our frontend side of the project, we adopted feature-sliced architectural, is an architectural methodology for structuring frontend applications, especially large-scale React apps. It focuses on scalability, maintainability, and clear separation of concerns by organizing the project into feature-oriented layers.

By slicing each feature into separated layers as shown the figure 2 and figure 1. This provide separation of concerns also explicit dependencies which enforces that lower layers shouldn't depend on higher layers also it provide isolation and encapsulation to each feature components and related logic.

4.1.6 Backend Application Architecture

For our backend architecture, we've adopted NestJs Architecture which is modular architecture which make our business logic highly scalable, maintainable, and testable. It is built on top of Node.js and uses TypeScript, heavily inspired by Angular's dependency injection and modular system.

The modular approach is based on three main core principles which are:

1. Modular Design: Applications are split into modules.
2. Dependency Injection (DI): Promotes loose coupling.
3. Layered Architecture: Separates concerns into controllers, services, and repositories.

All of these principles played a role on how the architecture was split into 3 parts.

NestJs split the modular approach into three main layers which are:

1. Controller Layer (Presentation Layer): This layer handle HTTP Requests (GET, POST, PUT, DELETE). It also calls the service layer to process the business logic and it send back the response to the user.
2. Service Layer (Business Logic Layer): It contains the core business logic, it comes after the controller layer and it calls the Repository Layer (if used) to fetch or update data.
3. Repository Layer (Data Access Layer): This layer handles the database operations (CRUD) also it allow the usage of ORM as in our case Mongoose and it keeps the data access separated from business logic.

In the conclusion, we followed the best architecture that fit with our technology use and also the system of devhub. And the modular approach is

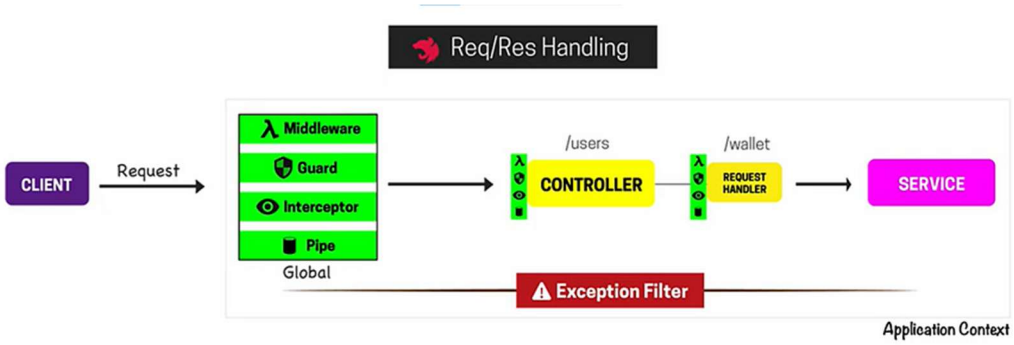

Figure 5 Backend Architecture

4.1.7 Agile Methodology:

We used the Agile model technique to execute our software system, strategically segmenting the process into five separate stages:

1. Requirement Analysis (Stage 1):

held lengthy brainstorming meetings to determine and list the core features that were intended for the program.

2. Planning (Stage 2):

Created a thorough schedule for completing the tasks, giving each activity a specified deadline.

conducted routine evaluations and conversations with the project supervisor once each assignment was finished.

3. Designing the Requirements (Stage 3):

sorted project work into priority categories, starting with the most important ones.

Regularly convened to assess advancements, deliberate on forthcoming tasks, and tackle obstacles, guaranteeing prompt resolution.

User stories were created for each project component, with the purpose, target audience, and technical details of each intended feature planned.

4. Development (Stage 4):

began coding each page's front end and back end respectively.

5. Testing (Stage 5):

Postman testing was used to verify the back-end’s functioning and make sure everything was correct.

6. Deploying (Stage 6):

We've used Vercel to upload our frontend part and Render to upload our Backend side.

Ensured that each page functions properly by establishing and testing the link between the front end and back end.

We were able to create and improve our software system iteratively thanks to the Agile model approach, which also ensured alignment with changing priorities and needs.

Figure 6: Agile Methodology

4.1.8 Testing:

With its many features, Postman is a very feature-rich and easy-to-use application that makes testing backend programs much easier. Because of its user-friendly interface, developers may create and execute a variety of HTTP requests, including GET, POST, PUT, and DELETE, with ease, facilitating extensive testing of backend features. Request setup is where Postman shines; it makes it simple to create parameters, headers, and body content, simulating real-world circumstances for reliable testing.

One of Postman's main advantages is its organization, which enables users to organize requests into collections and offers an organized method for managing and running test cases. Developers may move between several testing environments with ease since this arrangement includes environment variables. With the addition of automation processes, Postman goes beyond manual testing and makes it possible to write JavaScript test scripts for intricate testing scenarios and repetitive activities. Because of this automation, testing is far more efficient and produces repeated, reliable findings.

Postman also offers a thorough view of API answers, including headers, content, and status codes, through its deep response inspection feature. By allowing team members to share collections and requests, the technology makes cooperation easier. Testing is easily included into the software development lifecycle because to its connection with monitoring tools, continuous integration platforms, and version control systems. By enabling developers to record API endpoints, request forms, and anticipated answers, Postman's documentation features further increase its usefulness and make it an invaluable tool for both development and testing teams. In conclusion, Postman proves to be a priceless tool for developers, providing a comprehensive solution for backend testing through its collaboration features, automation capabilities, and user-friendly design.

Moreover, we have utilized swagger for better documentation and testing. Which has a lot of integration with NestJs so that it would be easier to test through UI with expected result and an interface to be able to use authentication barrier tokens through all APIs.

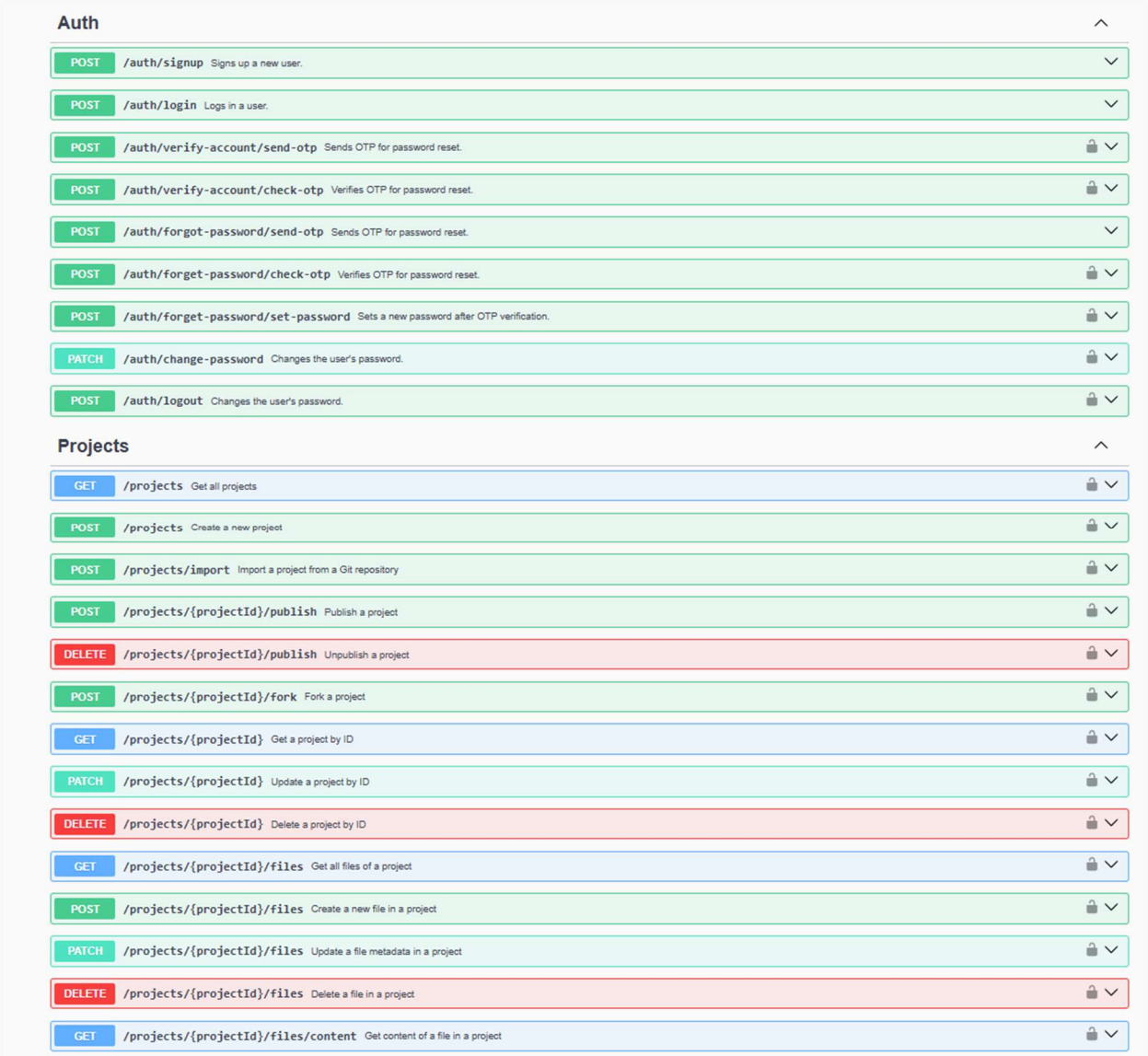

Figure 7 Swagger Interface A

Figure 8 Swagger Interface B

4.1.9 Version Control:

When it comes to version control and cooperation, GitHub is a vital tool that greatly improves collaborative software development processes. As a distributed version control system, GitHub helps teams easily monitor and manage changes to their codebase. The incorporation of GitHub into our development process has been crucial in promoting effective collaboration and maintaining the integrity of version control.

Serving as a single repository for our codebase is one of GitHub's main contributions to our collaborative efforts. With the ability to push code to GitHub, any team member may create a centralized, well-organized place for the project. This protects against potential data loss and facilitates cooperation in addition to acting as a safe backup. The version control features of GitHub are essential for supporting an organized and methodical approach to group coding. Developers may work on features or fixes in isolation (branching) and smoothly incorporate their changes back into the main codebase (merging) with the help of branching and merging techniques. By doing this, a clean and functioning codebase is maintained and it is ensured that several team members may contribute concurrently without interfering with each other's work.

One essential part of our collaboration approach is the pushing and pulling of code from GitHub. The revised code is available to the whole team when a team member pushes their modifications to GitHub after finishing a job or adding a new feature. On the other hand, team members pull the updates from GitHub when they require the most recent version of the code, guaranteeing that everyone is working with the most recent and well-organized codebase.

With capabilities like pull requests and code reviews, GitHub offers strong project management, problem tracking, and collaboration tools as well. Before integrating code into the main branch, pull requests provide team members the opportunity to suggest improvements, have a discussion about revisions, and check the quality of the work. This cooperative review procedure reduces possible problems and improves the quality of the code.

To sum up, GitHub is essential to our methods for collaboration and version control. Our development processes are now far more organized and efficient because to its centralized code storage, version control, and collaboration features. GitHub has become an essential component of our workflow because to its smooth pushing and pulling of code procedures, which enable our team to collaborate on projects, monitor changes methodically, and produce code that meets high quality standards.

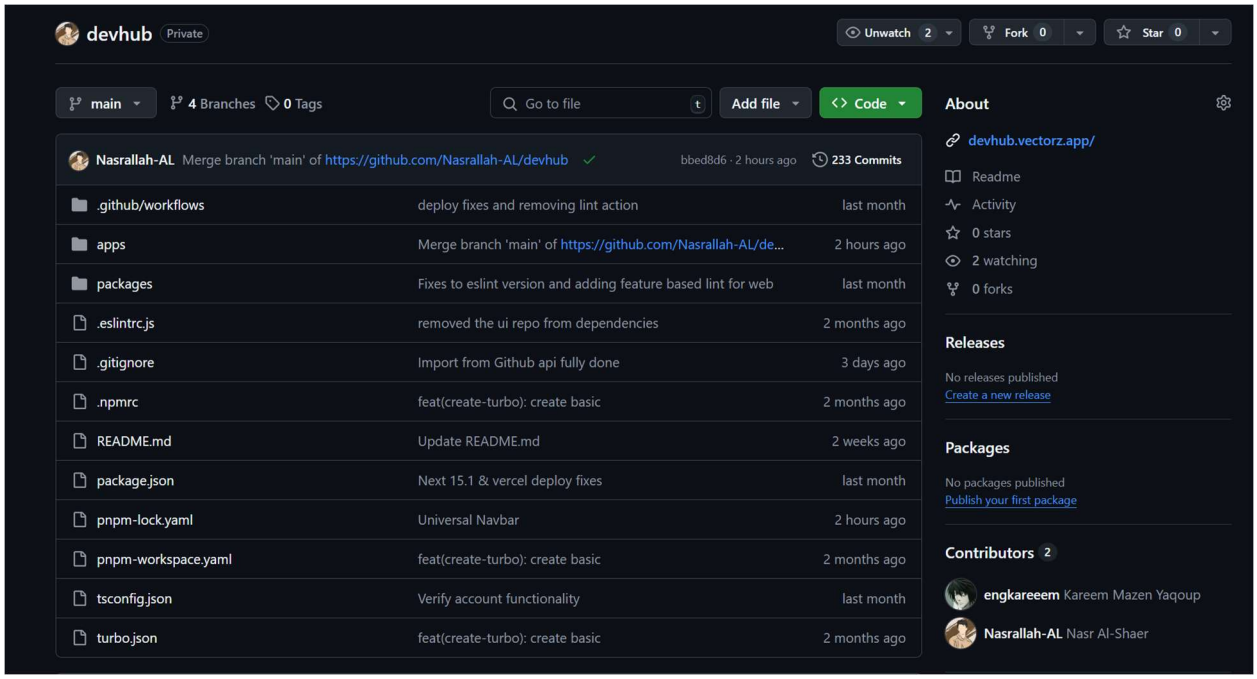

Figure 9 DevHub Github Repo

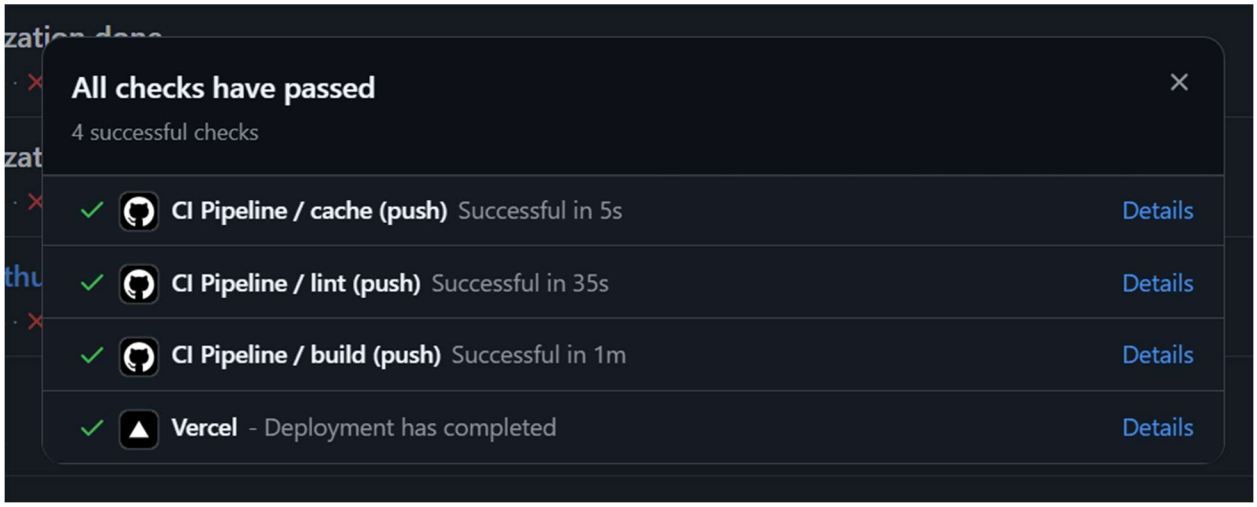

Figure 10 Github Actions in use

Figure 11 Github Production Deployment

4.2 Technologies and Tools

4.2.1 NestJs

NestJS is an advanced Node.js framework for leveraging TypeScript to create server-side apps that are scalable and maintainable. It is extremely organized and effective since it is based on Express (or alternatively Fastify) and uses an Angular-inspired modular design. Because NestJS comes with built-in support for WebSockets, GraphQL, dependency injection, and middleware, developers can create reliable microservices and APIs. NestJS is extensively utilized in business applications, real-time systems, and backend services that need high speed and maintainability due to its robust typing, scalability, and interaction with several databases and authentication techniques.

4.2.2 NextJs

Building high-performance online apps with capabilities like server-side rendering (SSR) and static site generation (SSG) to improve speed and SEO is possible with Next.js, a React-based framework. Development is streamlined by its support for TypeScript, GraphQL, and CSS frameworks, automated code splitting, and API routes. Next.js is widely used in content-driven, SaaS, and e-commerce websites and offers modern web applications flexibility, scalability, and optimization.

4.2.3 MongoDB

MongoDB is a NoSQL database that is perfect for modern online applications because of its excellent performance, scalability, and flexibility. It saves data in a format called BSON, which is similar to JSON and enables dynamic, schema-less structures that may change to meet evolving needs. MongoDB offers a robust query language for intricate data operations, replication for high availability, and horizontal scaling via sharding. It is a well-liked option for real-time analytics, Internet of Things applications, and massive distributed systems due to its compatibility with several programming languages and cloud services.

4.2.4 Mongoose

Mongoose is used to make database interactions in Node.js applications easier, Mongoose is an Object Data Modeling (ODM) module for MongoDB. It offers a flexible, schema-based method for generating data structures that permits middleware, validation, and query construction. Complex data models are easier to handle with Mongoose's built-in type casting, pre- and post-processing hooks, and support for connections between data. It is a well-liked option for creating scalable, data-driven apps with organized and maintainable code because of its smooth interaction with MongoDB and Node.js.

4.2.5 Atlas

The deployment, scalability, and administration of MongoDB databases are made easier with MongoDB Atlas, a fully managed cloud database service. It supports multi-cloud deployment across AWS, Azure, and Google Cloud and offers security features, performance optimization tools, and scheduled backups. Atlas is perfect for contemporary applications needing high availability and scalability since it has integrated monitoring, real-time analytics, and worldwide data dissemination. Atlas is popular for cloud-native apps, massive data processing, and corporate solutions because of its smooth interface with MongoDB and developer-friendly tools.

4.2.6 Turborepo

Designed to streamline development processes, Turborepo is a high-performance build system for handling TypeScript and JavaScript monorepos. It greatly accelerates builds and eliminates pointless calculations by enabling effective caching, parallel execution, and task scheduling. Code sharing and dependency management across several projects are made easier with Turborepo's integrated support for well-known package managers like npm, Yarn, and pnpm. It is an effective solution for teams creating large-scale, modular applications with enhanced productivity and maintainability because of its smooth interface with cloud caching and CI/CD

4.2.7 Vercel

Vercel is a cloud platform that prioritizes developer experience, scalability, and performance when developing and hosting frontend apps. It offers serverless features, automated deployments, and worldwide content distribution via its edge network, all while integrating seamlessly with frameworks like Next.js, React, and Vue.js. Vercel is a great option for contemporary web apps, static websites, and serverless solutions needing high availability and low latency since it allows for quick previews, continuous deployment from Git repositories, and integrated performance

4.2.8 Render

Render is a cloud platform that facilitates the automatic scalability, cost-effectiveness, and ease of use of databases, web apps, and APIs. Along with capabilities like serverless functions, managed PostgreSQL, and integrated HTTPS, it supports a number of runtimes, including Node.js, Python, and Go. Render offers worldwide CDN for optimal speed, intelligent scalability based on traffic, and smooth Git integration for continuous deployment. Render's pay-as-you-go pricing and developer-friendly methodology make it perfect for personal projects, startups, and scaled production applications.

4.2.9 React-native (expo)

The React Native (Expo) framework allows developers to construct native apps for iOS and Android from a single codebase by combining JavaScript with React to create cross-platform mobile applications. With built-in libraries, over-the-air upgrades, and simple deployment that doesn't need knowledge of native code, Expo streamlines development. It has capabilities like device API access, rapid reloading, and smooth cloud service integration. For creating scalable and effective mobile apps, React Native (Expo) is popular because to its quick development cycle and low complexity.

4.2.10 Tailwind

Tailwind CSS is a utility-first CSS framework that offers a collection of pre-defined classes for styling components straight in HTML, hence facilitating quick user interface creation. Tailwind promotes consistency and maintainability across apps by encouraging developers to create bespoke designs by mixing utility classes, in contrast to standard CSS frameworks that provide pre-designed components. Tailwind is a well-liked solution for creating visually appealing and responsive user interfaces because of its responsive design features, customization choices, and interaction with contemporary JavaScript frameworks. Tailwind CSS, which prioritizes flexibility and speed, allows developers to keep control over the appearance of their apps while streamlining the design process.

We have used Tailwind and Nativwind to style our web and mobile application.

4.2.11 Docker

With the help of the Docker platform, developers can automate the deployment and administration of apps inside of small, portable containers. From development to production, these containers guarantee consistency across various contexts by encapsulating all the dependencies and parameters needed to operate an application. Docker makes version control, scalability, and application separation simple, which streamlines the software development lifecycle. When paired with technologies like Docker Compose and Kubernetes, its orchestration features make it perfect for continuous integration/continuous deployment (CI/CD) processes and microservices architectures. Teams may optimize resource use, expedite deployment procedures, and increase communication using Docker.

We have used docker container to support our sandbox functionality by creating customized image for each supported sandbox, so that each user can run his project using isolated ready-to-use containers on the AWS ECS.

4.2.12 AWS S3

AWS's scalable, high-performance object storage solution, Amazon S3 (Simple Storage solution), is made to store and retrieve any volume of data from any location on the internet. With capabilities like data encryption, versioning, and lifecycle management, it offers a robust and safe option for data storage. Backup and recovery, big data analytics, content distribution, and storing static webpages are just a few of the many applications that S3 is perfect for. Developers and companies can effectively manage and access their data in the cloud using S3's user-friendly administration dashboard and interaction with other AWS services, guaranteeing high availability and dependability.

We have used AWS S3 to store user projects, profile pictures and templates of the sandboxes/projects.

4.2.13 AWS ECR

Developers can safely store, manage, and deploy container images with AWS's fully managed Docker container registry service, Amazon Elastic Container Registry (ECR). Building and deploying containerized apps is made easier by ECR's smooth integration with other AWS services like Amazon ECS and EKS. ECR improves the security and effectiveness of container operations with features including fine-grained access control using AWS Identity and Access maintenance (IAM), lifecycle rules for automatic image maintenance, and image scanning for vulnerabilities. Because of its scalable design, which guarantees high availability and performance, ECR is a crucial part of any organization's development and deployment processes that use

We have used ECR to store our sandbox images on the cloud so that it can be used by the ECS easily and updated on the fly.

4.2.14 AWS ECS

The deployment, administration, and scalability of containerized applications on AWS are made easier using Amazon Elastic Container Service (ECS), a fully managed container orchestration service. Users may execute apps on serverless infrastructure or Amazon EC2 instances thanks to ECS's support for Docker containers and smooth integration with other AWS services. ECS gives developers the ability to effectively manage microservices architectures with capabilities like load balancing, job scheduling, and service discovery. It is perfect for developing and executing scalable, high-performance cloud applications because it offers strong security, monitoring, and logging features. ECS is especially helpful for businesses trying to improve application delivery and optimize their DevOps procedures.

AWS ECS was the main service used to run our container, so that we enable the developers to run their code/projects inside an isolated containers hosted and managed on the cloud.

4.2.15 Github Actions

With the help of GitHub Actions, an effective automation tool built into the platform, developers can design continuous integration and continuous deployment (CI/CD) processes from inside their repositories. It enables users to automate processes like code deployment, testing, and building in response to pull requests and code pushes, among other events. The GitHub Marketplace offers a vast array of pre-built activities that developers can readily modify to meet their demands, guaranteeing consistency and effectiveness in their development procedures. With its support for a wide range of programming languages and frameworks, GitHub Actions is a vital tool for current software engineering methods that aim to improve software quality, accelerate development cycles, and foster collaboration.

We have utilized github actions for code checking, linting, caching and deploying. With github actions we have managed to control the flow of continuous deployment based on the checks of the commits/merges on the main branch.

4.3 Features:

4.3.1 Web Authentication System:

Authentication was done using Access and refresh token. These tokens are stored in the user browser cookies once entered valid credentials to the system.

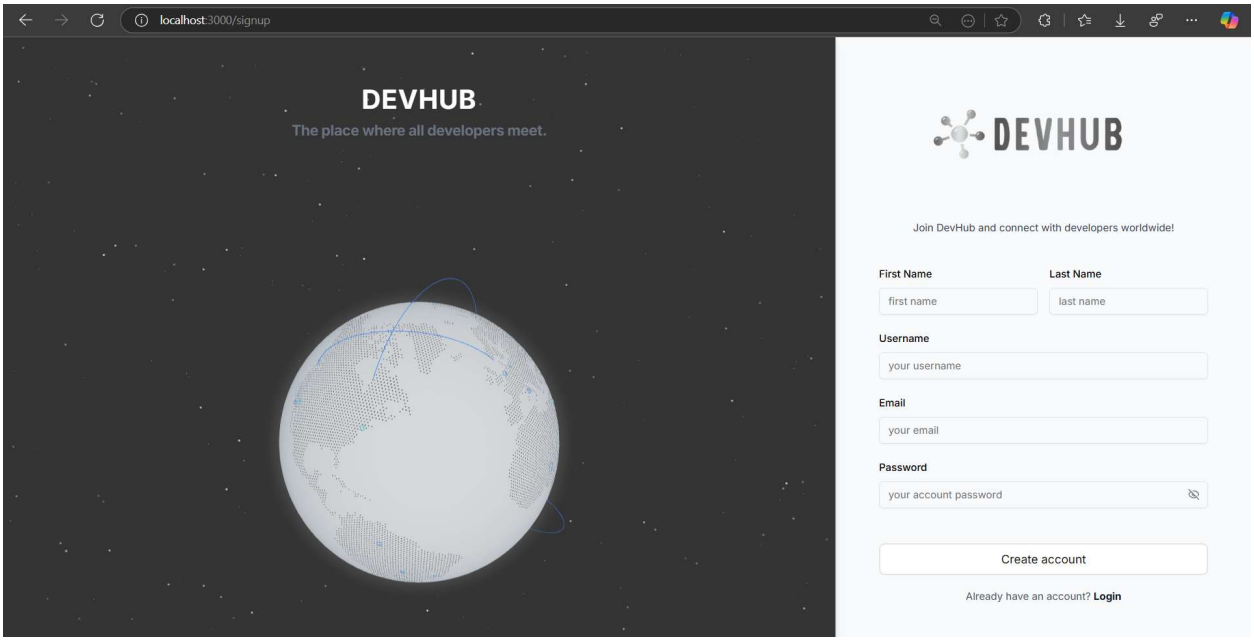

Figure 12: Signup Page

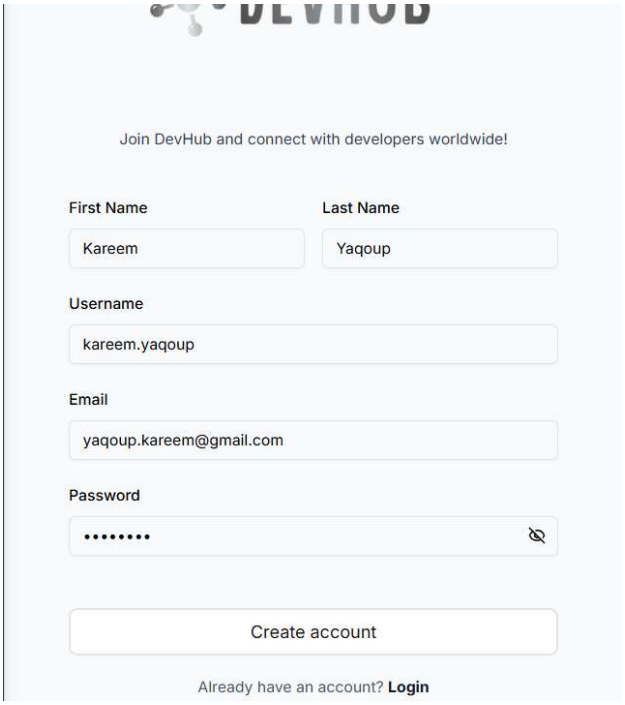

Figure 13: Signup Process

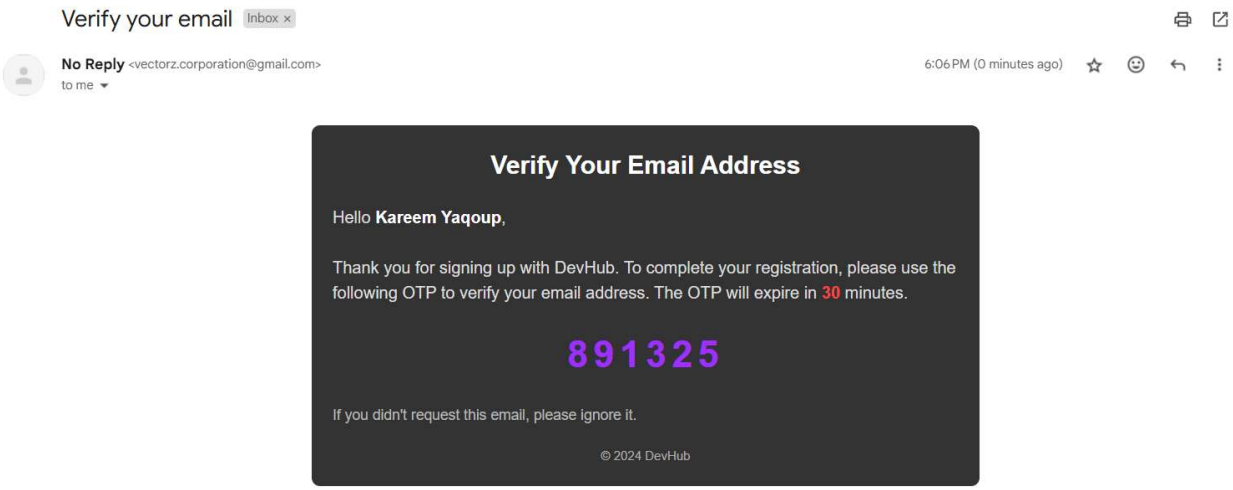

Figure 14: OTP Email

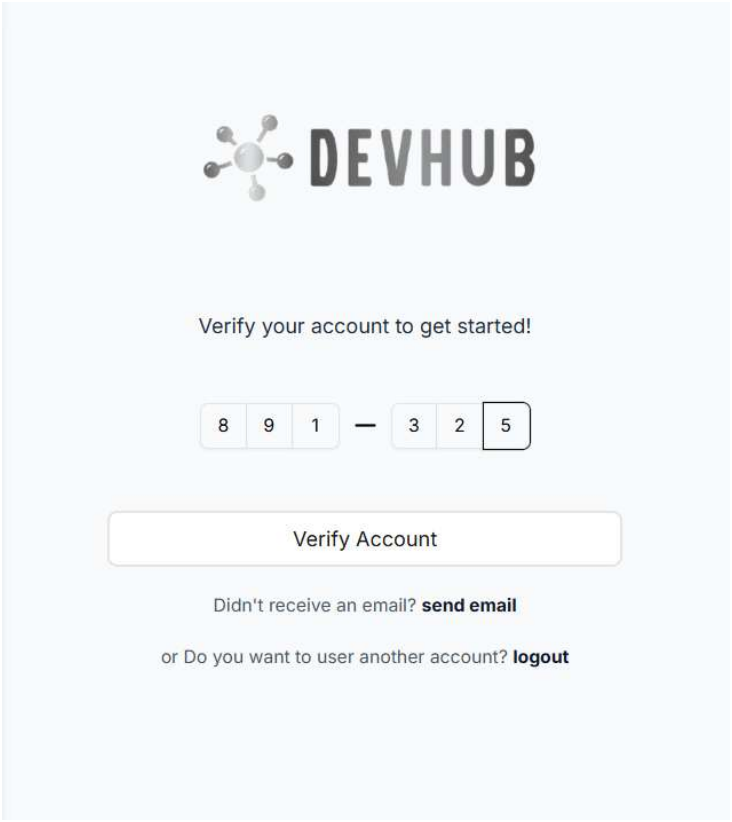

Figure 15: Verifying OTP

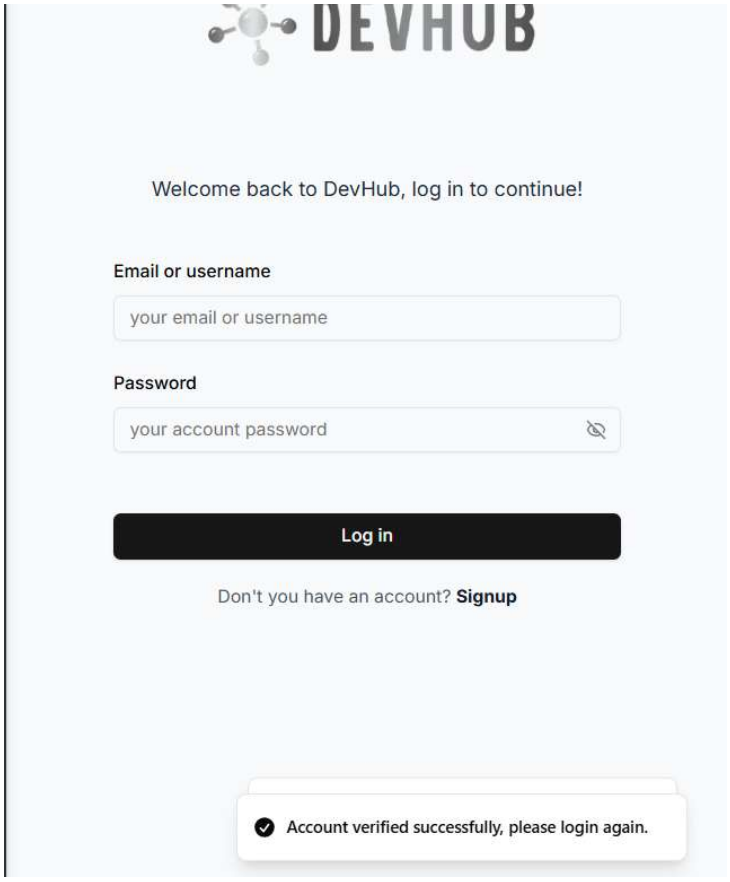

Figure 16: Account Creation completed

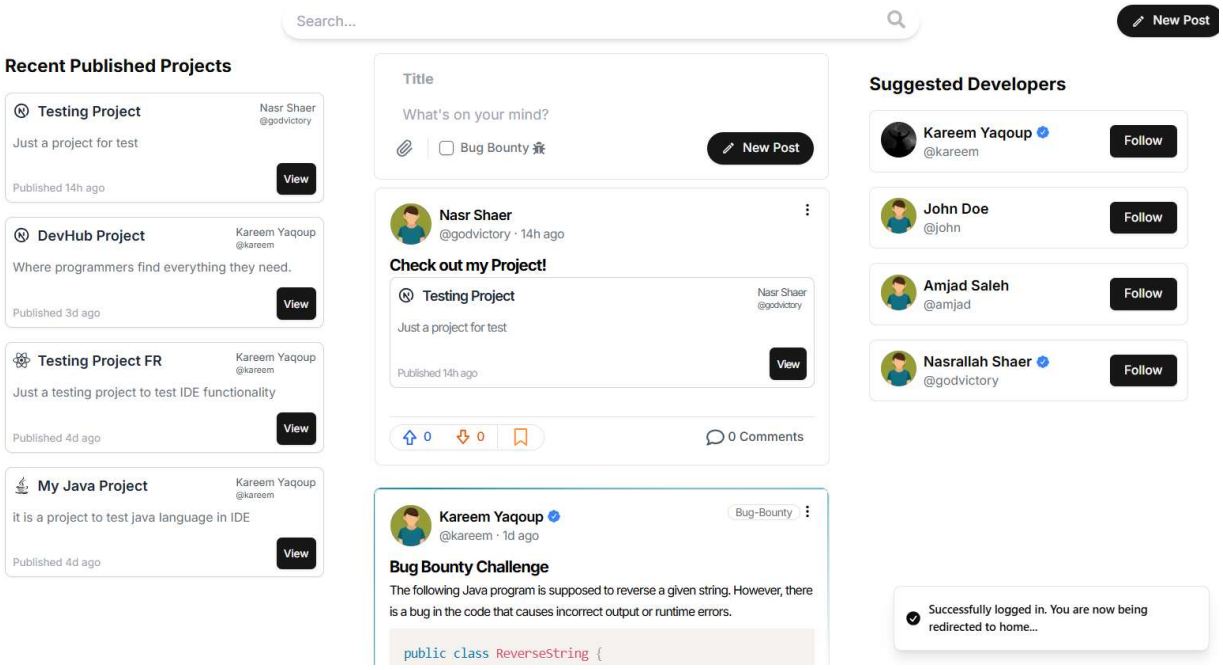

Figure 17: Authentication Process Complete

4.3.2 Mobile Authentication

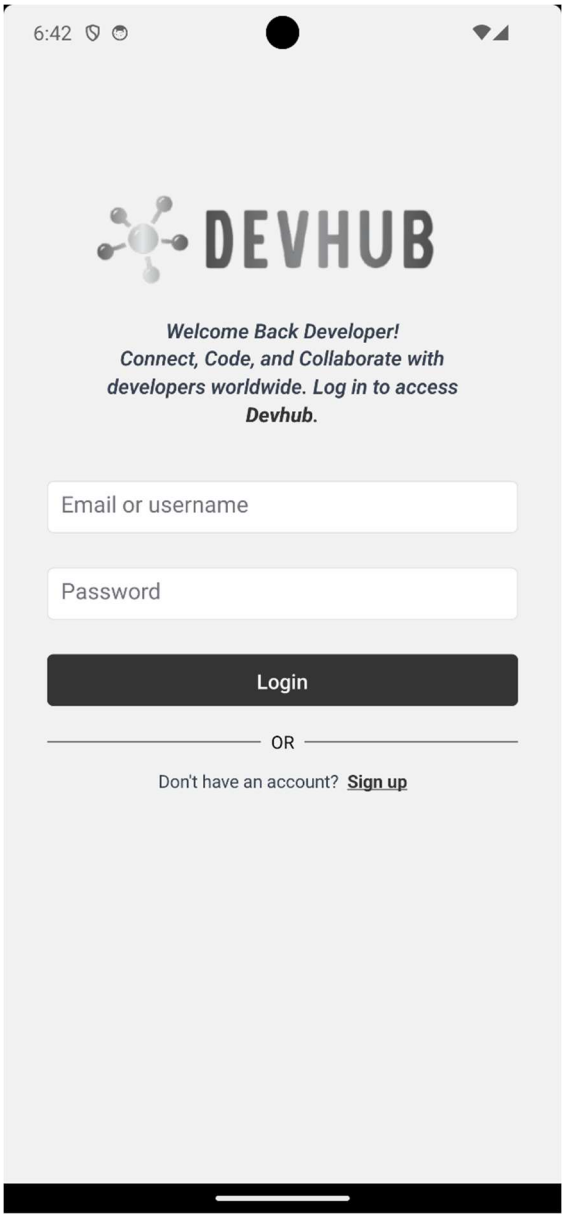

Figure 18 Mobile Signup View

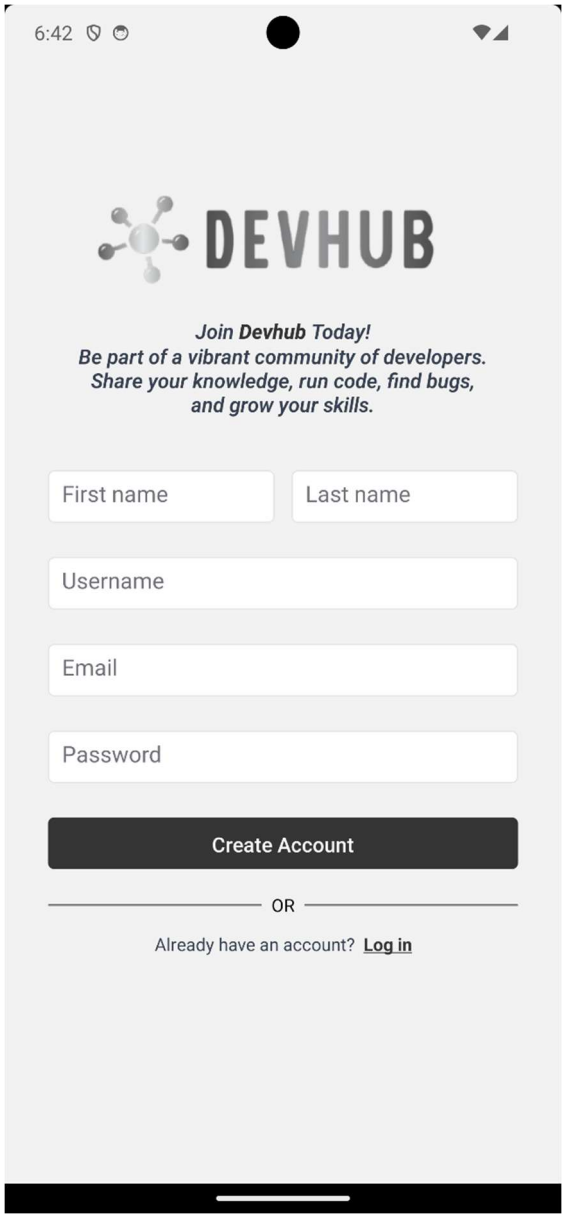

Figure 19 Mobile Login View

4.3.3 Web Community

4.3.3.1 Home Page

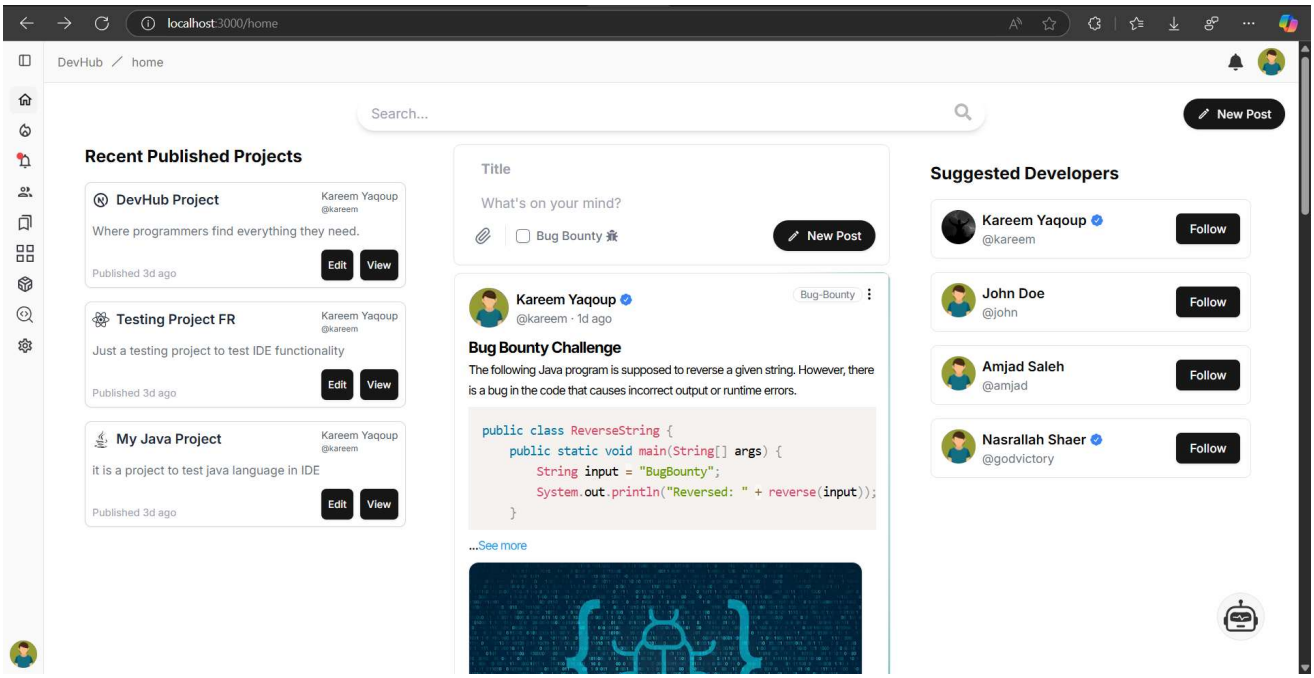

Figure 20: Home Page

Here is the home page of the website, where the developer can share his ideas and help from others.

The home page contains **Recent Published Projects, Developers Posts** and **Suggested Developers** to connect**.**

4.3.3.2 Posts

There is Three types of Posts:

1- Normal Post
2- Project Post
3- Bug-Bounty Challenge Post

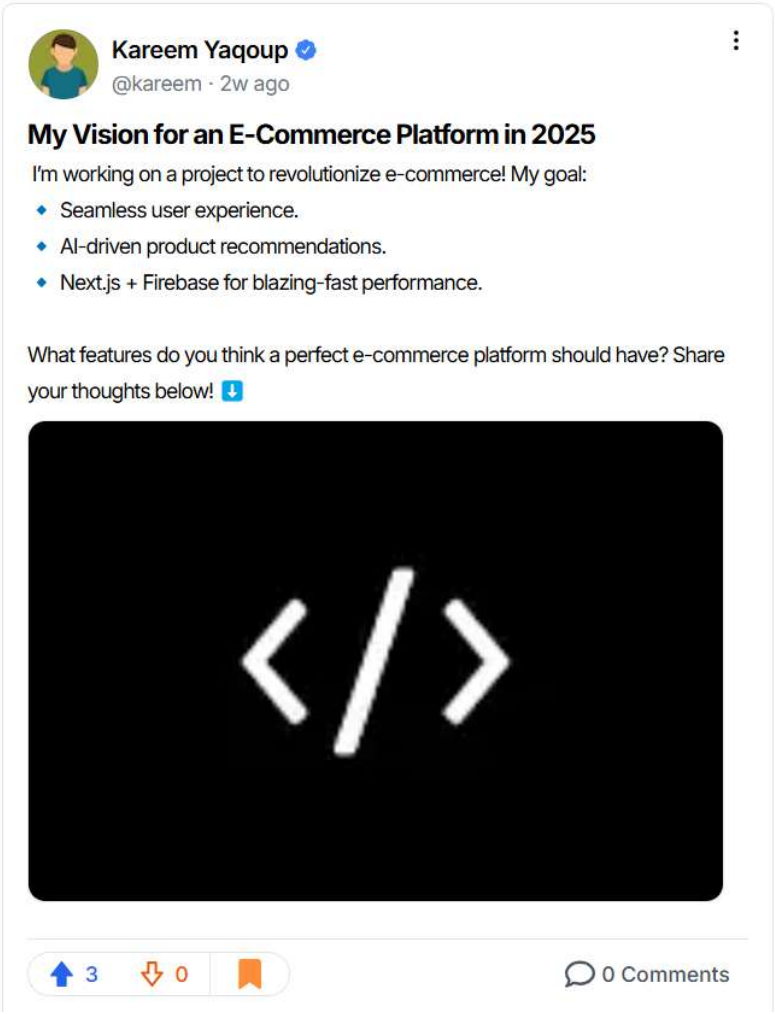

Figure 21: Normal Post

- Normal Post
  Developers can post a normal post to share an information or knowledge between each other, and the other developers can upvote the post or downvote it, and they can comment.

Figure 22: Project Post

- Project Post
  The developer also can post a project that he published before.

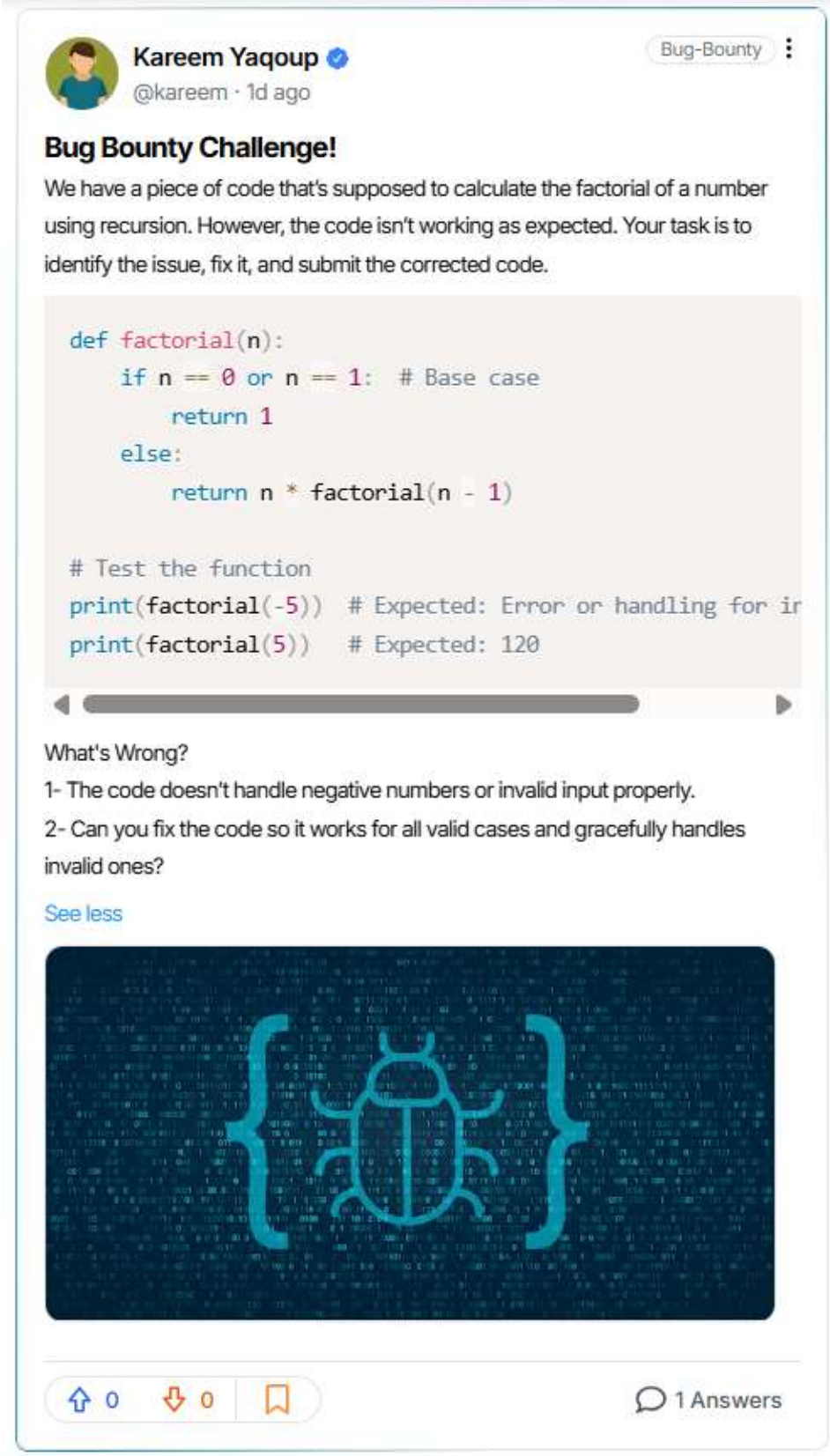

Figure 23: Bug-Bounty Challenge Post

- Bug-Bounty Challenges
  Verified Developers can make bug-bounty challenge and post it in the community, the developers should find the best solution and the challenge author can accept solutions that he deems correct.

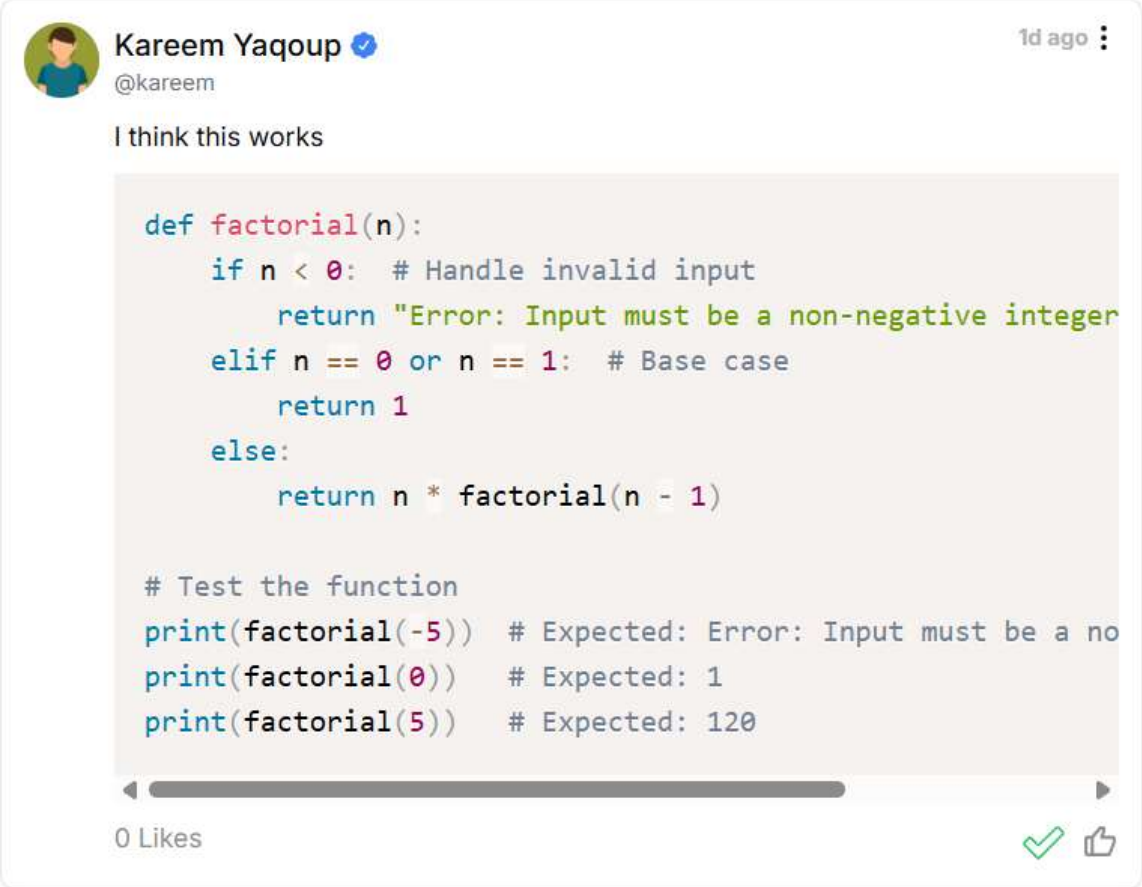

Figure 24: Bug-Bounty solution/answer

- Bug-Bounty Solution/Answer
  Users can put their answers in the comments section, and the author can accept the answer by clicking the green check mark.

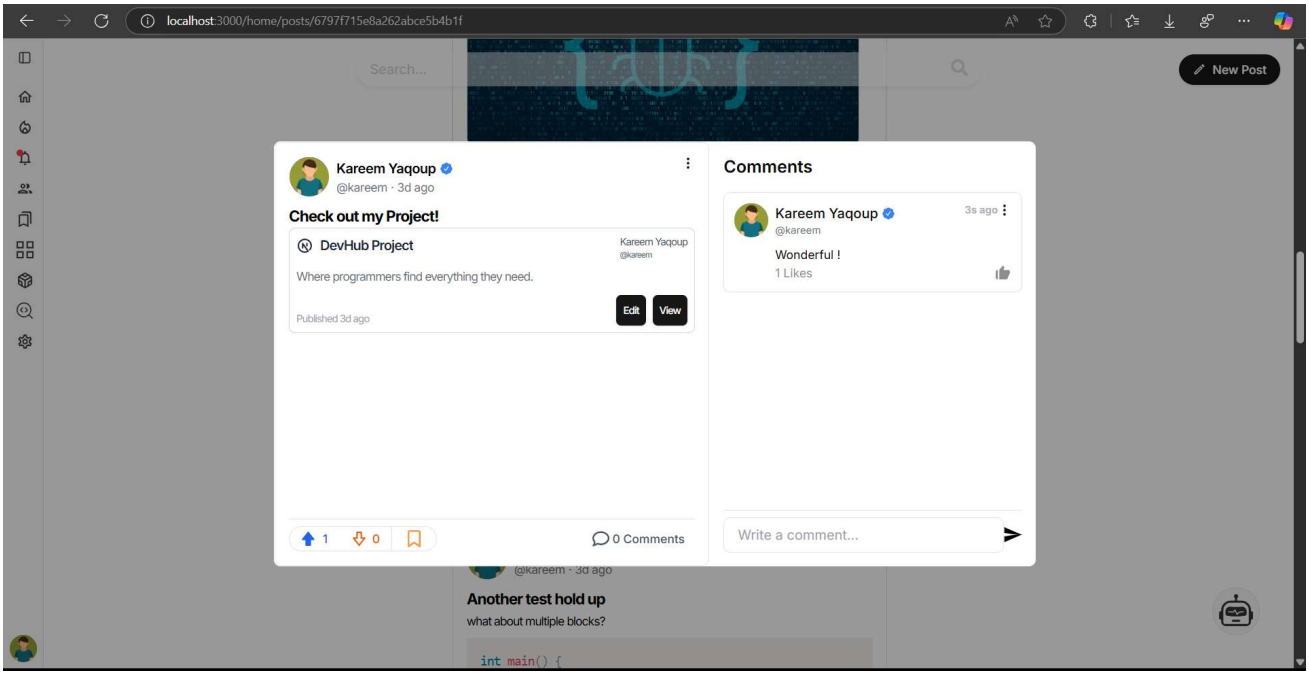

Figure 25: Post Comments

- Post Comments
  The developer can comment on posts and share his opinion about it, also he can like other comments.

Figure 26: Post content format

- Post content format
  The post can contain mentions, hashtags, URLs and code blocks.
  And the user who mentioned will be notified about the mention.

Figure 27: Comment format

- Comment content format

Comments also have the same format as posts.

Figure 28: Post Options

- Post Options
  The author can edit his post or delete it.

- Posts Pagination
  The home page has Auto-Pagination system, it fetches **5** on every page, and when the user scroll down, another **5** will be fetched.

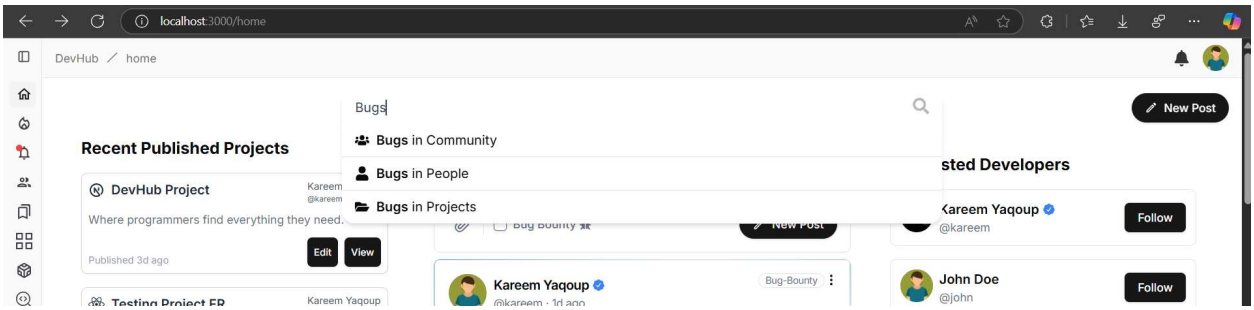

Figure 29: Home Page Search

- Search
  The developer can search in the community for posts, users and projects.

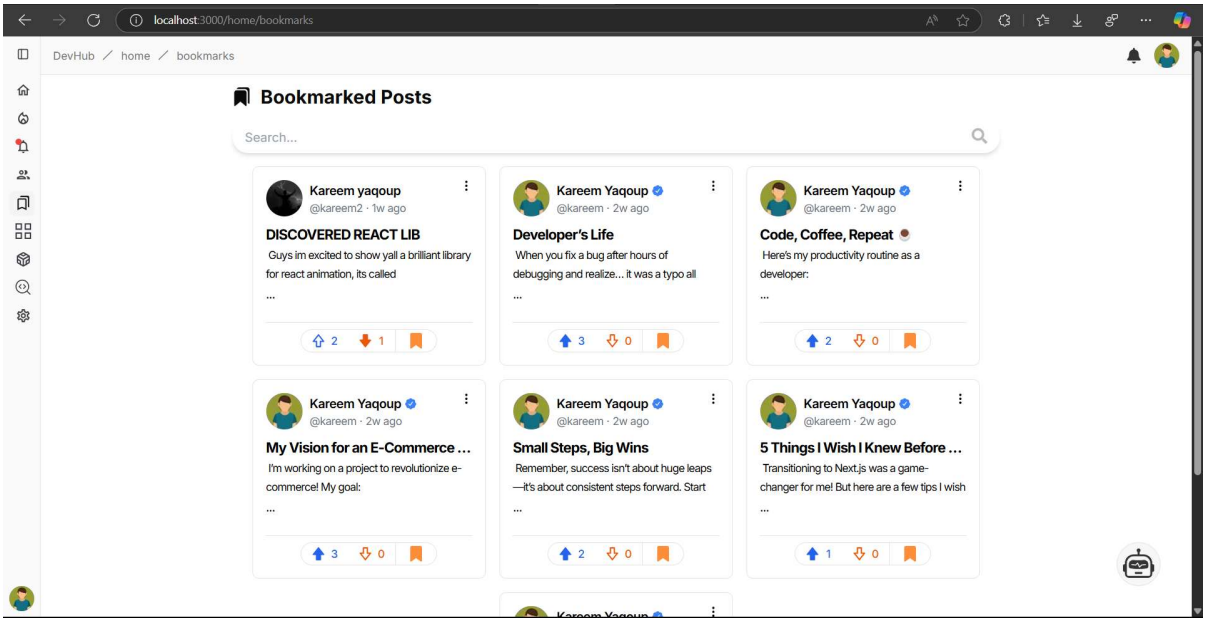

Figure 30: Bookmarked posts

- Bookmarked Posts
  The developer can bookmark a post to come back to later, and there is a page that shows the bookmarked posts in shortened way, and he can also search for a bookmarked post.

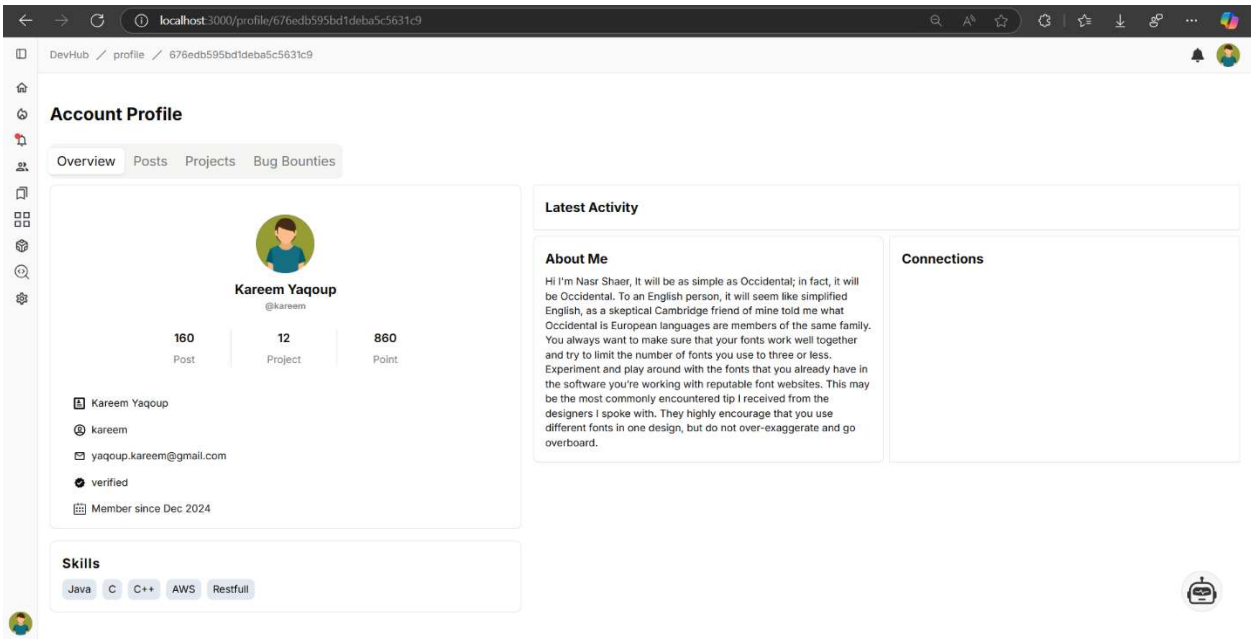

Figure 31: Account Profile

- Account Profile

4.3.3.3 Notifications

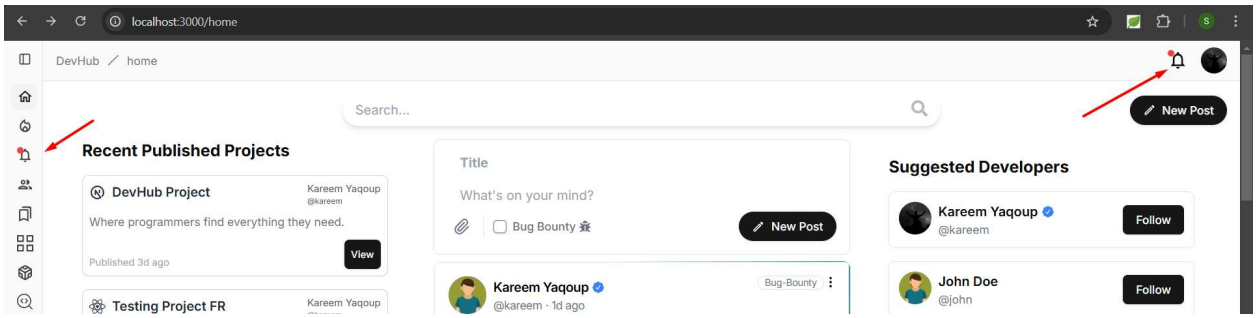

Figure 32: User Notifications

- User Notifications
  The user can see his notifications by clicking the bell icon on the nav-bar or on the side-bar

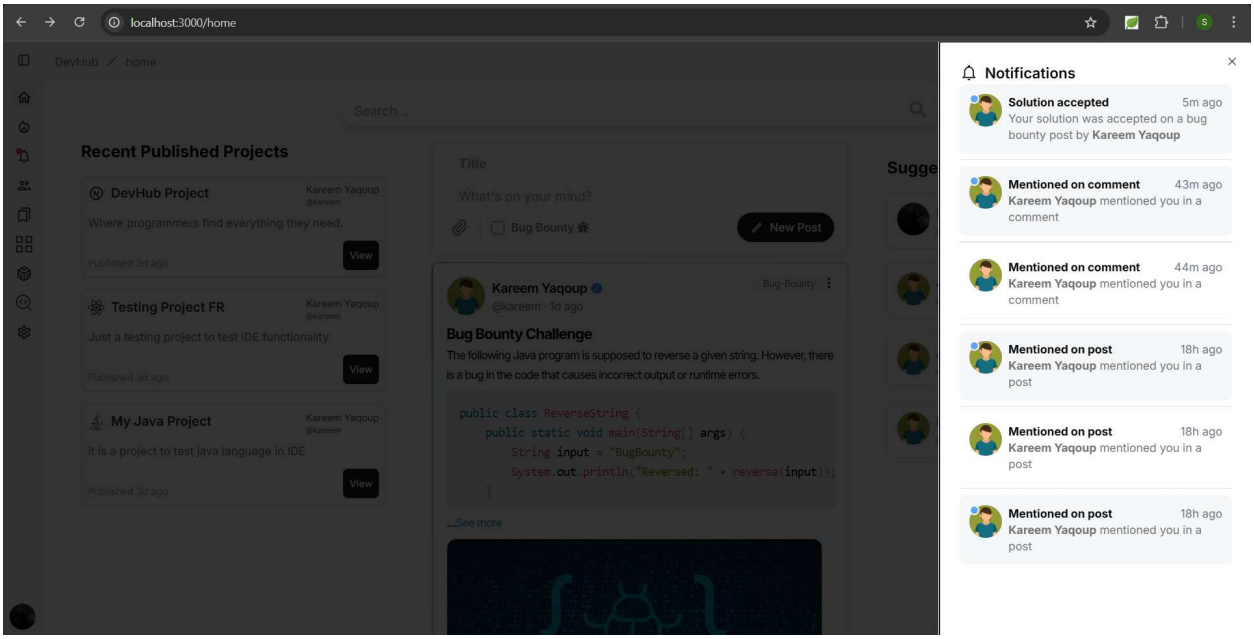

Figure 33: Notifications Display

- Notifications Display
  When clicking the bell icon, Side-sheet that contain user's notifications will show up

4.3.4 Mobile Community

Figure 34 Mobile Users View

Figure 35 Mobile Posts View

4.3.5 Code Lense

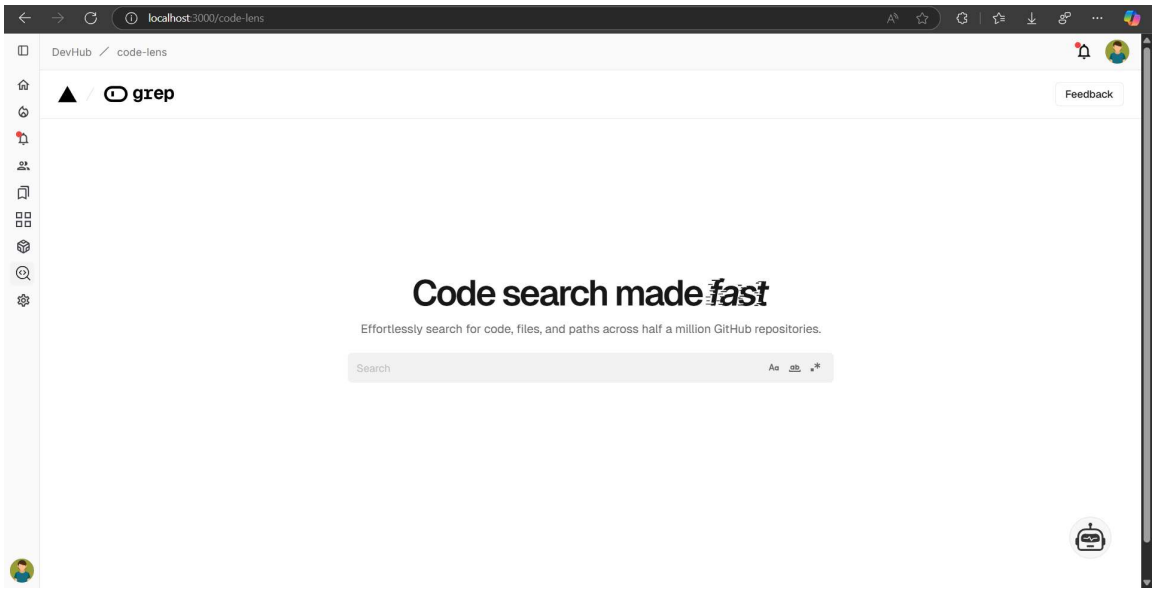

Figure 36: Code Lens

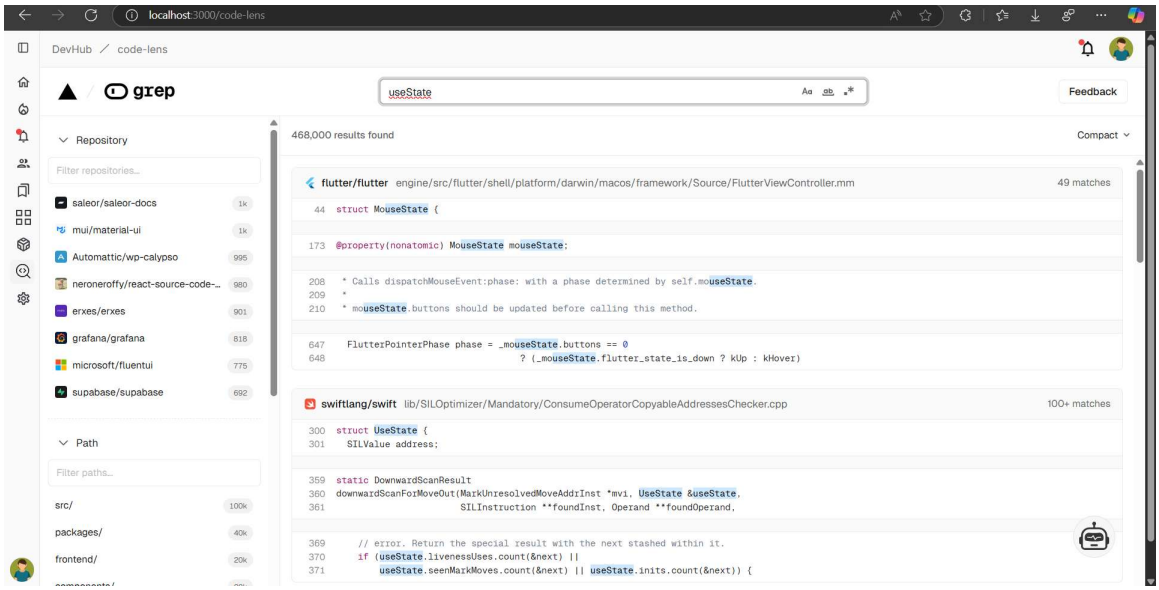

Figure 37: Code Lens Results

- Code Lens
  Most of studies agree on that best way to learn something is by doing it, so by letting developers of this platform search across whole GitHub open-source project for a specific segment of code to see it in action. Code lens make it easy by just provide the search query as a plain text or even a regex to enable the user to see all related pieces of codes to this code and by providing a lot of customization and options to let the user find what he really wants

4.3.6 Projects

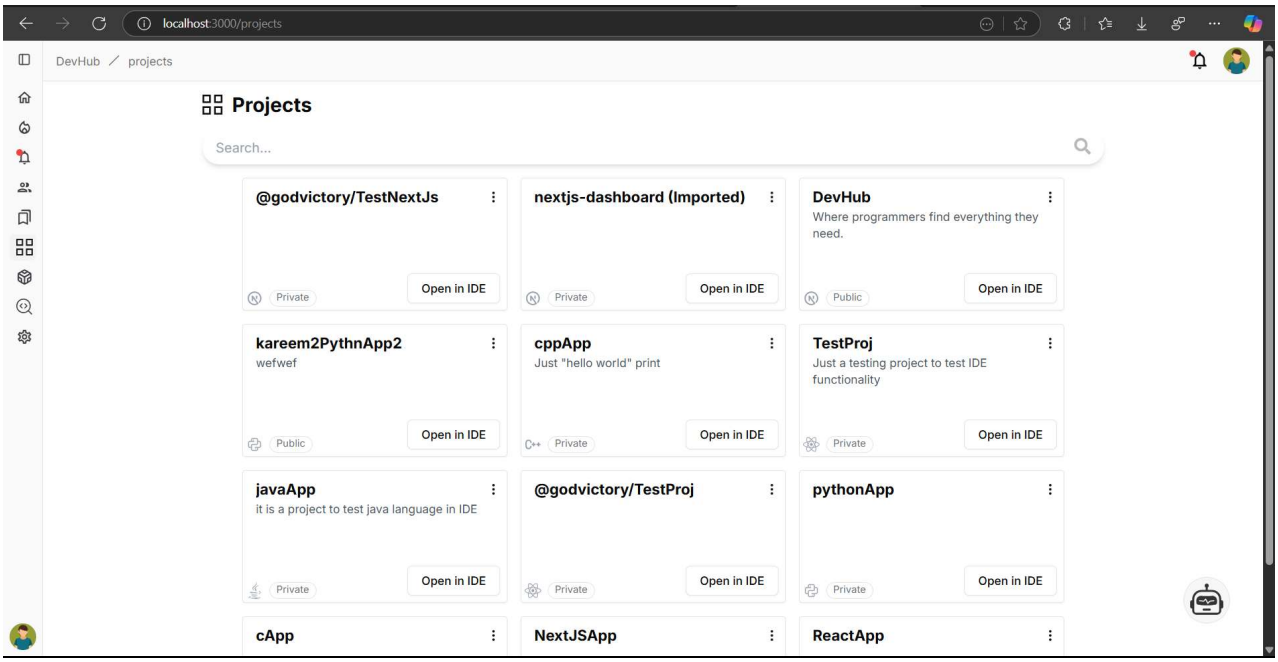

Figure 38: Projects Page

- Projects Page
  Here where the developer can find his projects and the projects he contributes in.

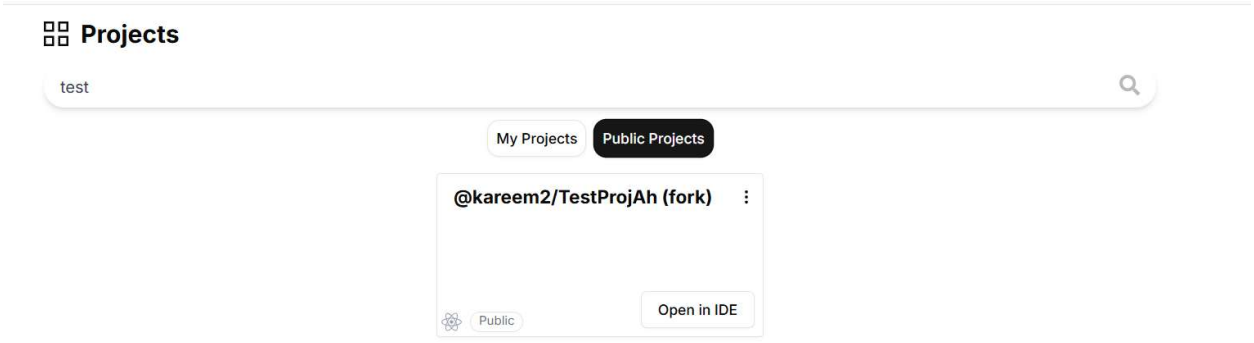

Figure 39: Search in public projects

- Projects Search
  The developer can also search for his projects and public projects from other users.

4.3.7 IDE (Integrated Development Environment)

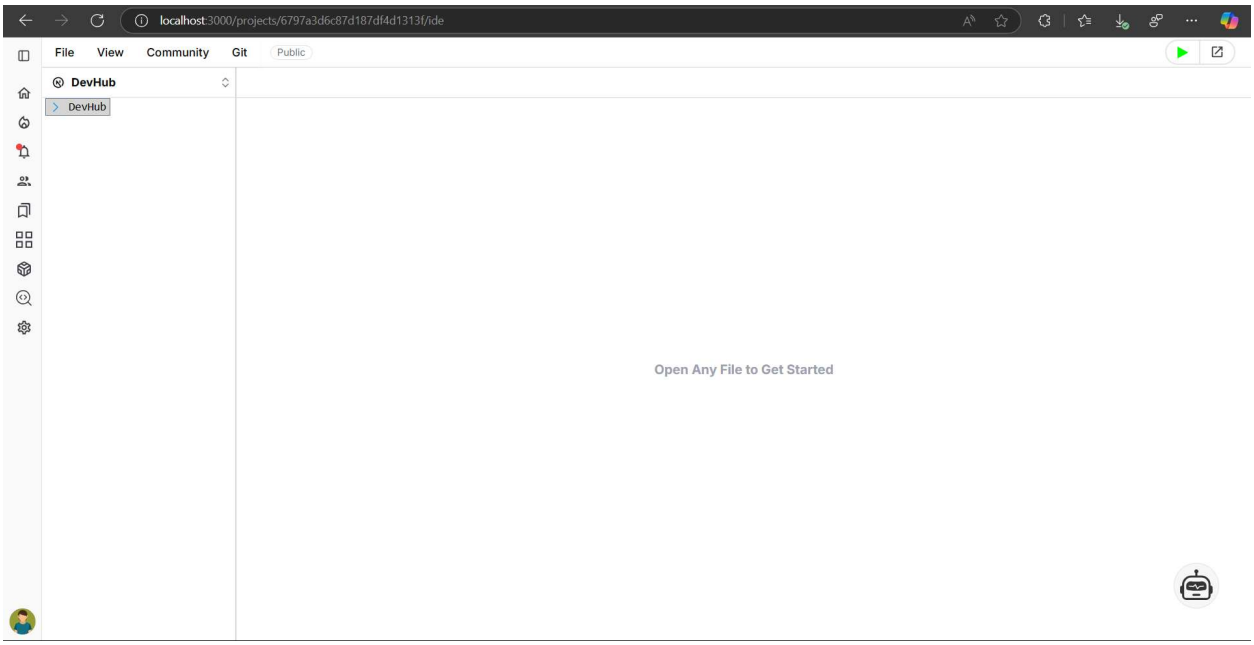

Figure 40: IDE

- IDE
  The developer can open any project he has access on it in IDE, and develop his project.

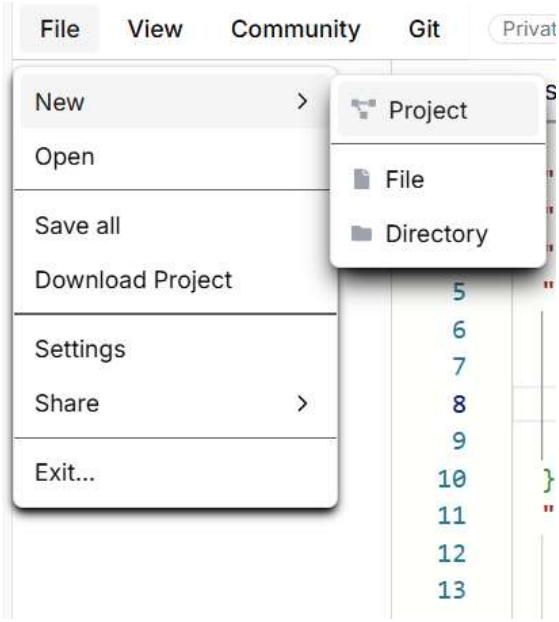

Figure 41: New project option

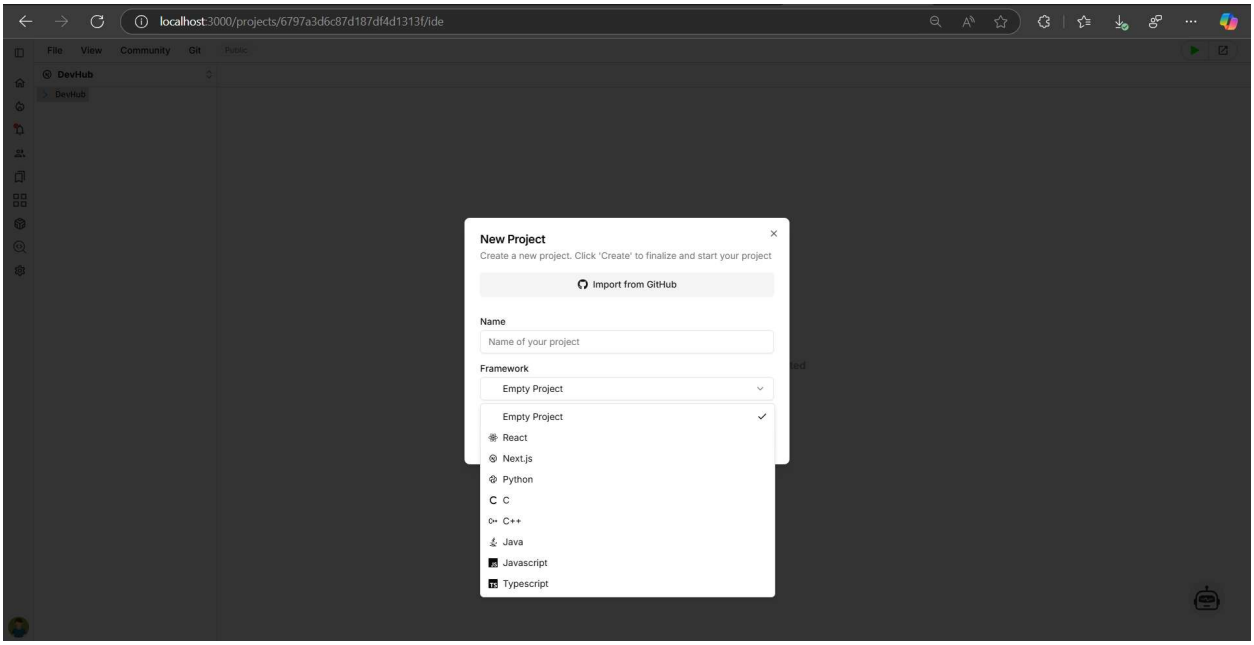

Figure 42: New Project

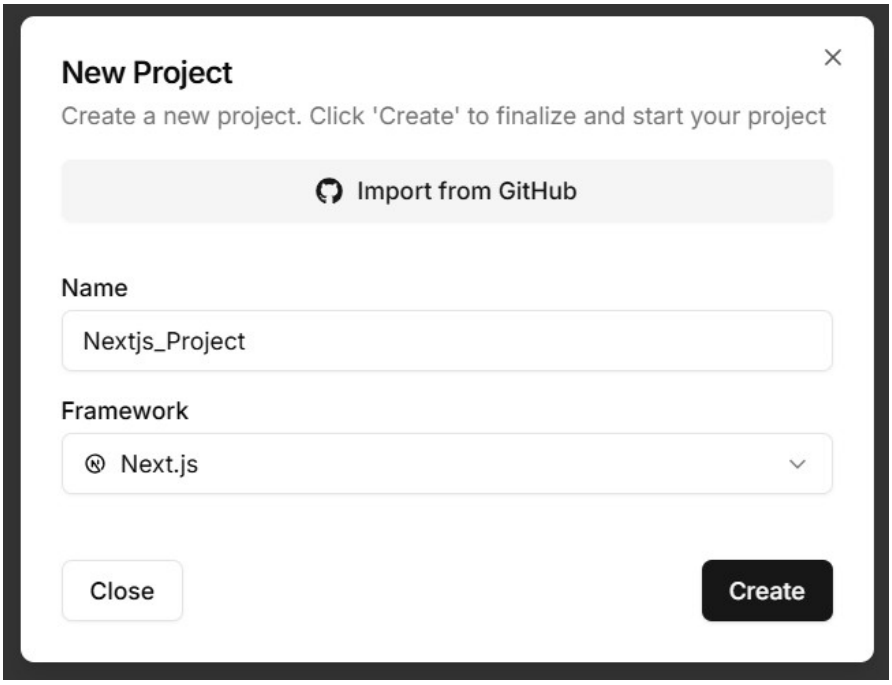

Figure 43: Creating new project

- New Project

The developer can create new project and select one of templates that IDE supports, and he can also import a project (Repository) from GitHub by clicking on “Import from GitHub”.

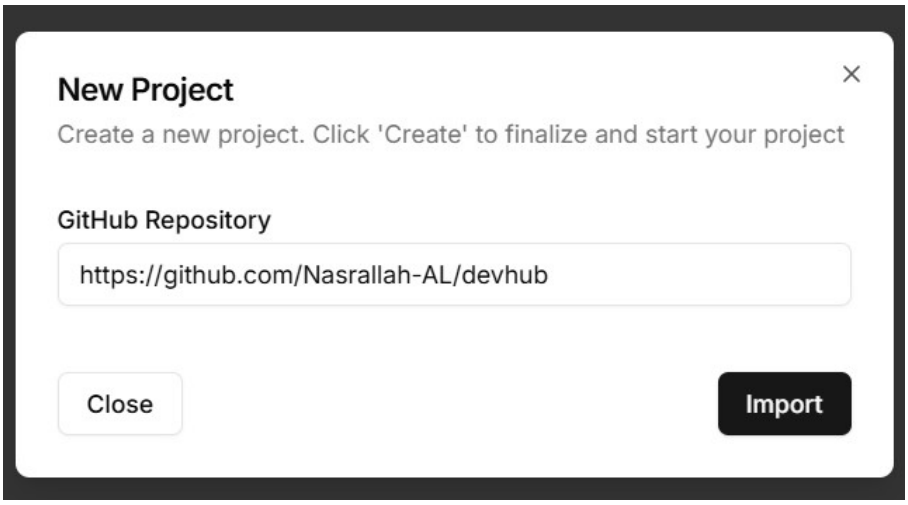

Figure 44: Import from GitHub

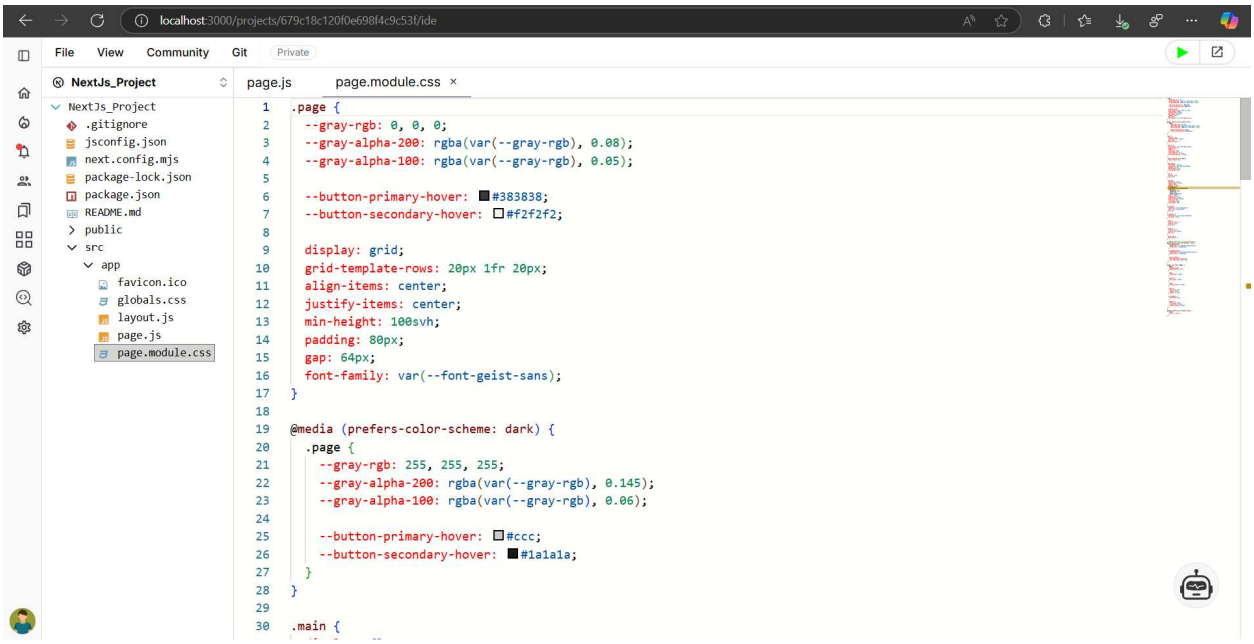

Figure 45: Developing Project

- Project in IDE
  Developer can develop his project in IDE and make something great.
  The file structure loads when the project opens, and the content fetched by demand (Lazy load).

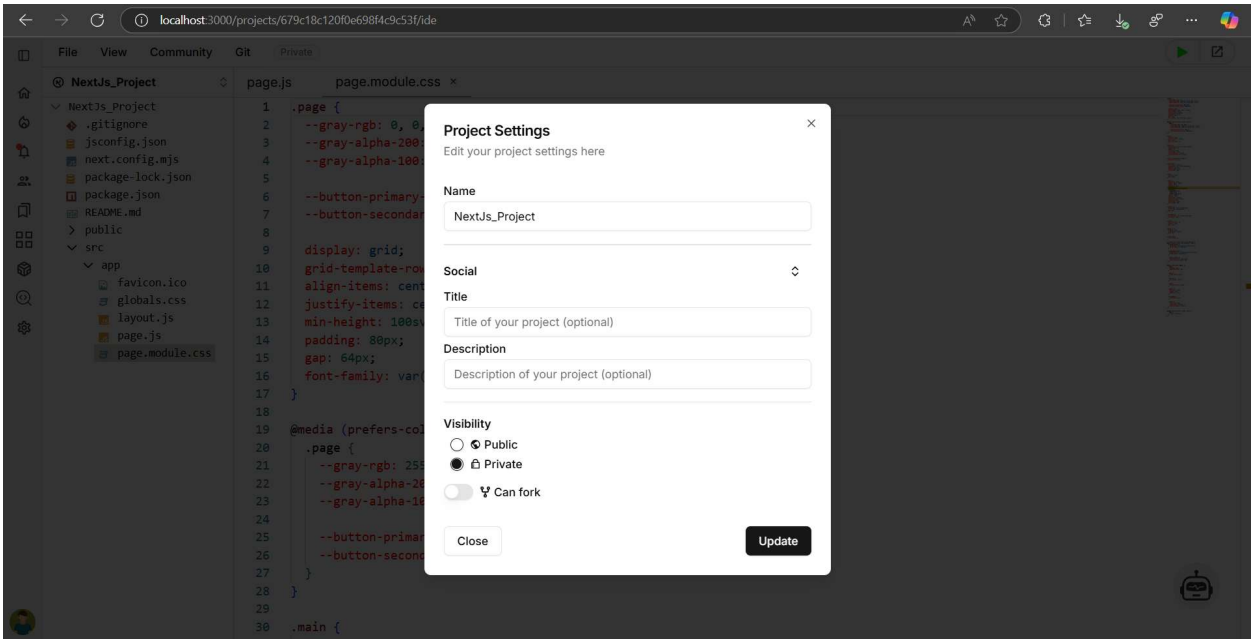

Figure 46: Edit Project

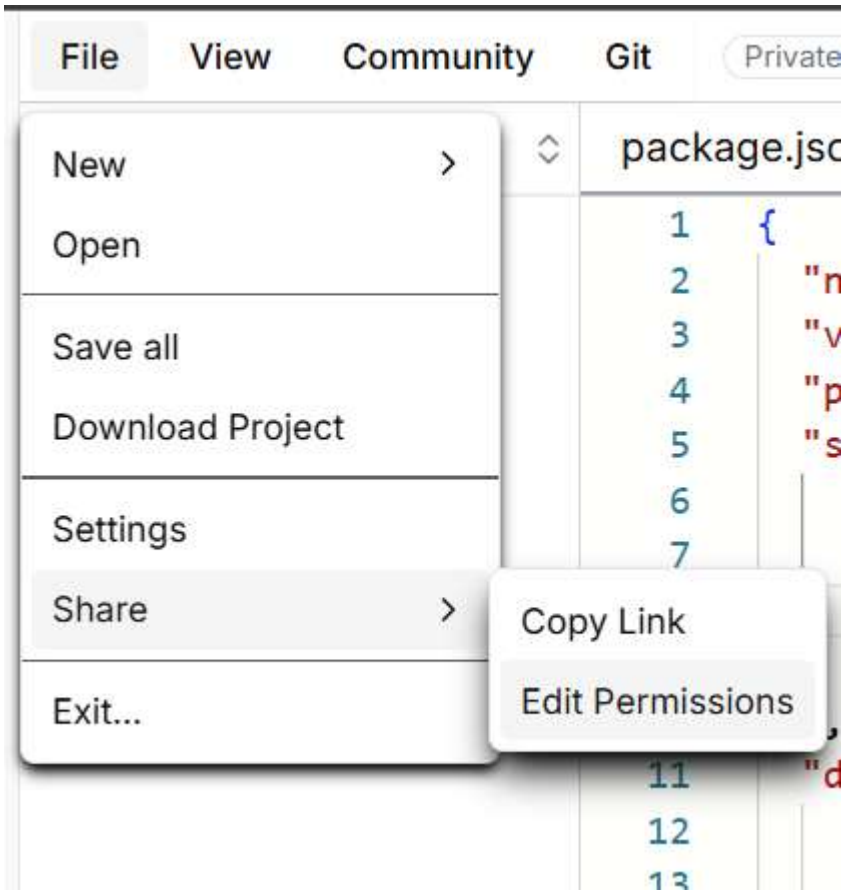

Figure 47: Permission edit option

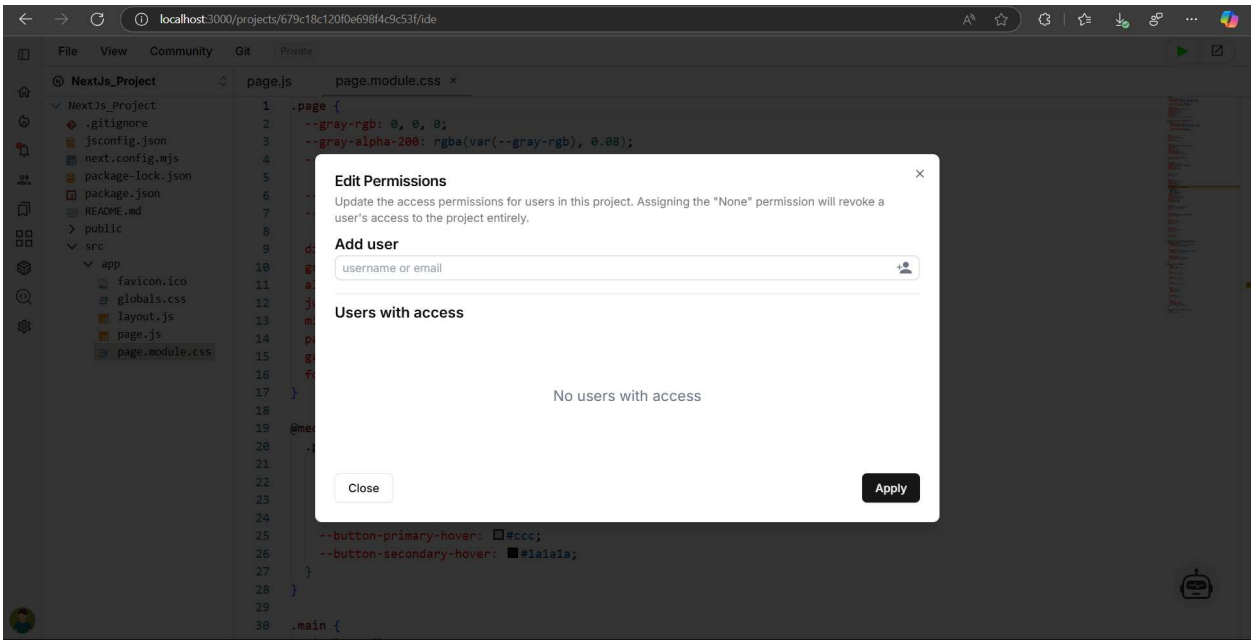

Figure 48: Edit Project Permissions

- Project Permissions
  Project owner can give access to other developers to participate the development process, he can give “read only”, “data edit” and “full access” permissions.

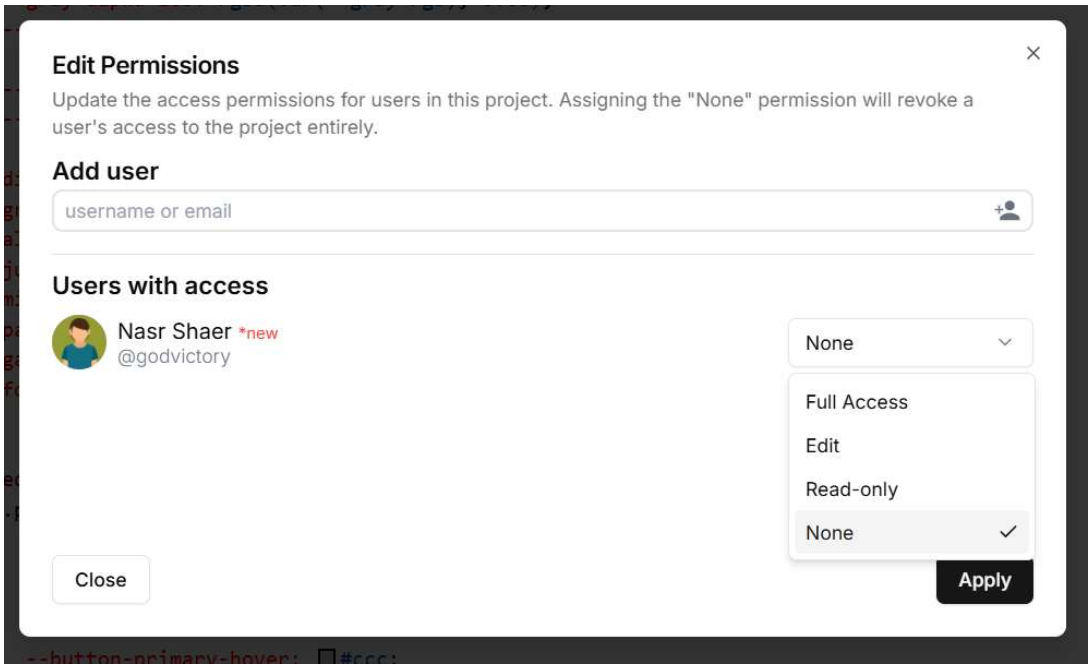

Figure 49: Give access to a developer

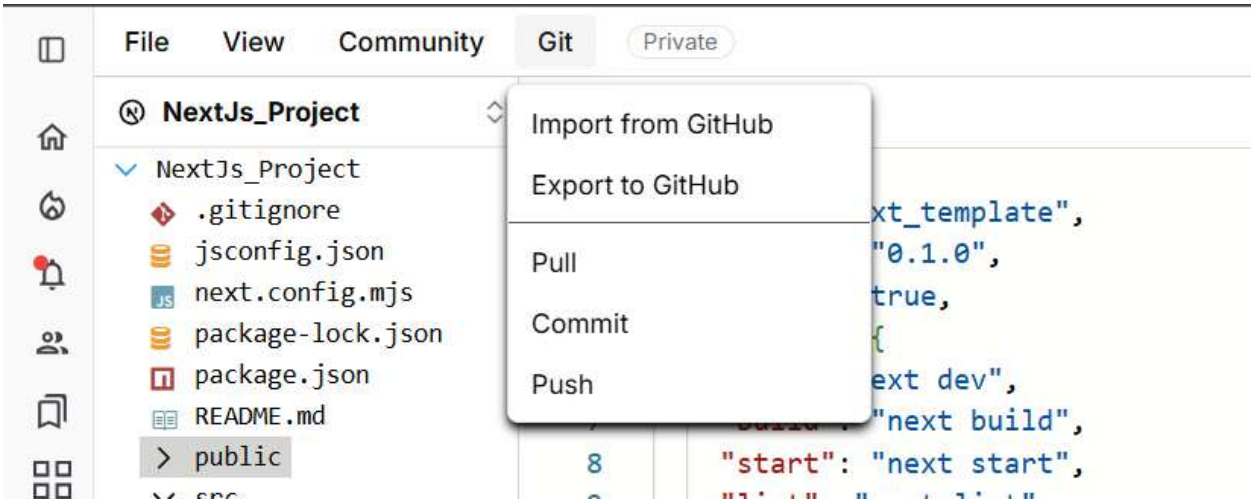

Figure 50: Git Options

- Git Options
  The developer can make Git operation in the IDE, like import or export a project

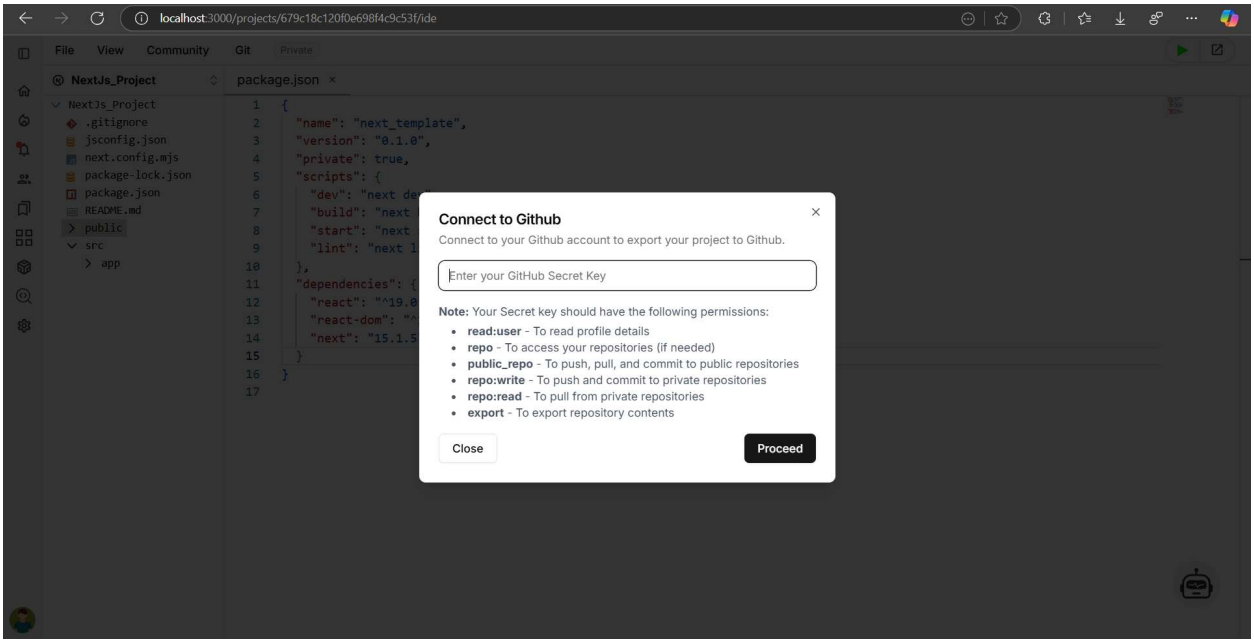

Figure 51: Connect to GitHub

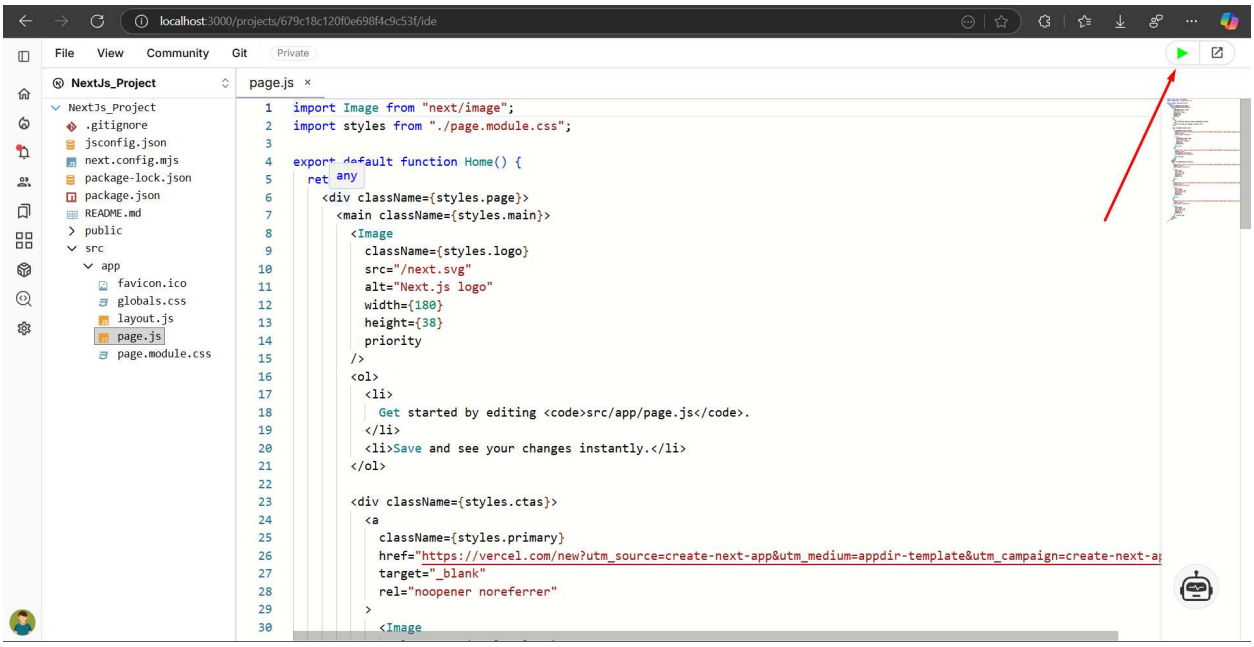

Figure 52: Run a framework project

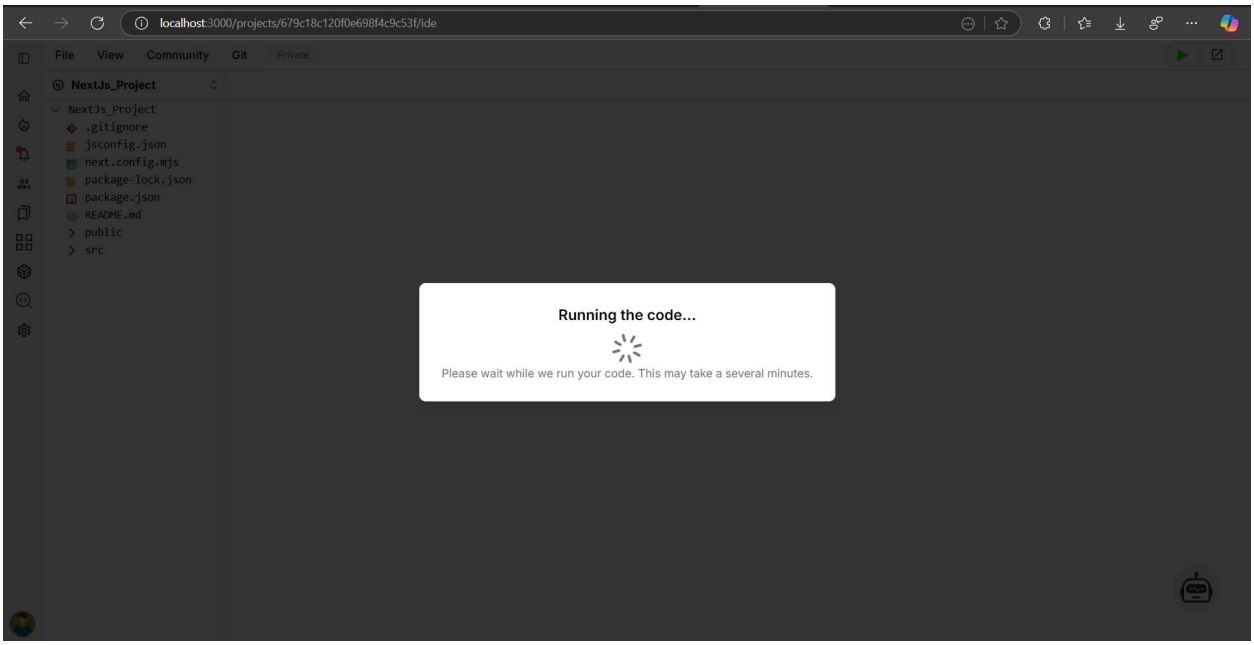

Figure 53: framework project is running

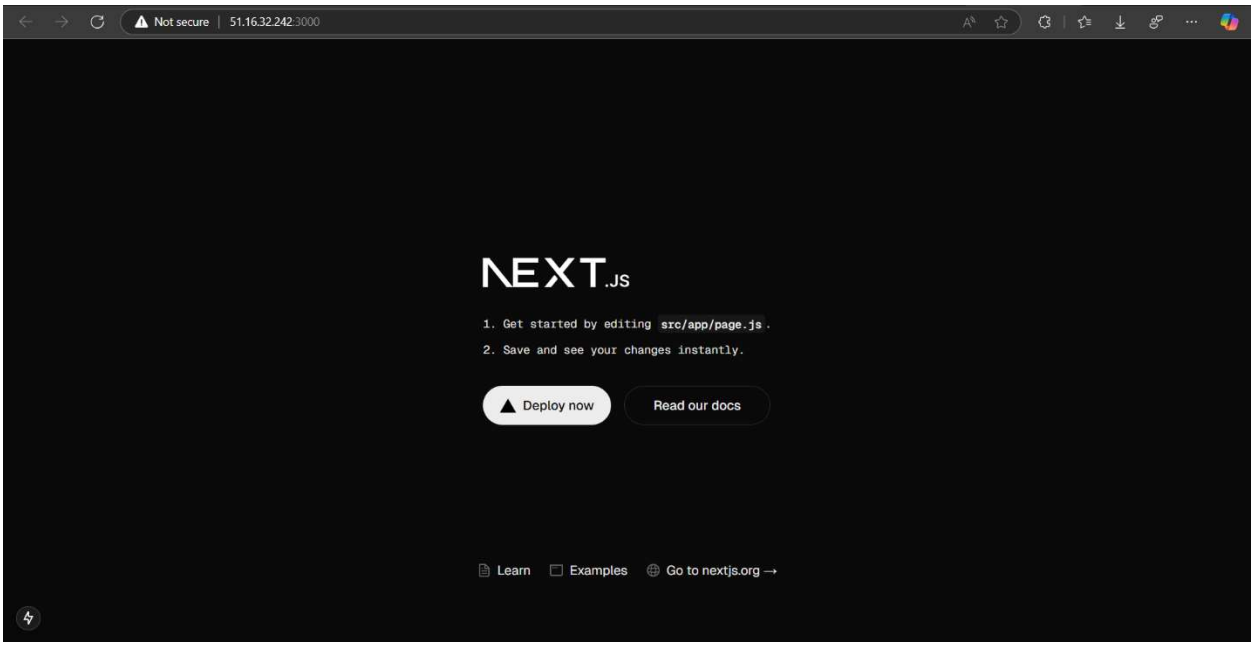

Figure 54: framework project result

- Project result
  After running the nextjs project, the website will redirect you to a new tab with output (Rendered NextJs Project)

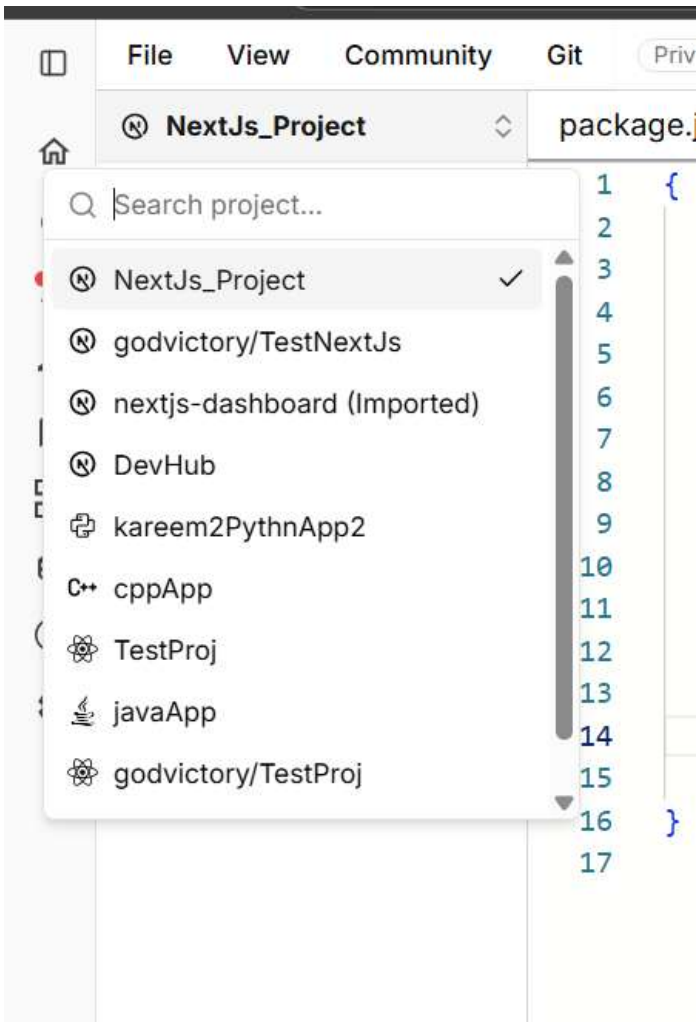

Figure 55: Projects Quick Access

- Developer can see his latest projects by clicking on the current project header and when clicking on any project it navigates him to the project directly

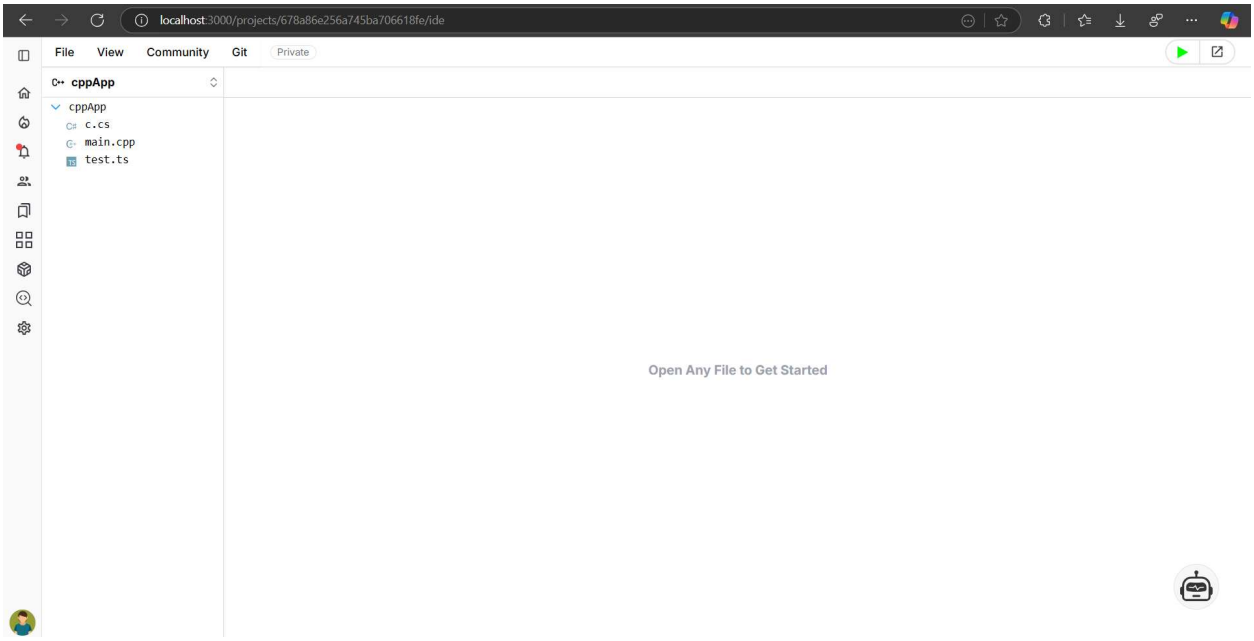

Figure 56: Navigating to another project

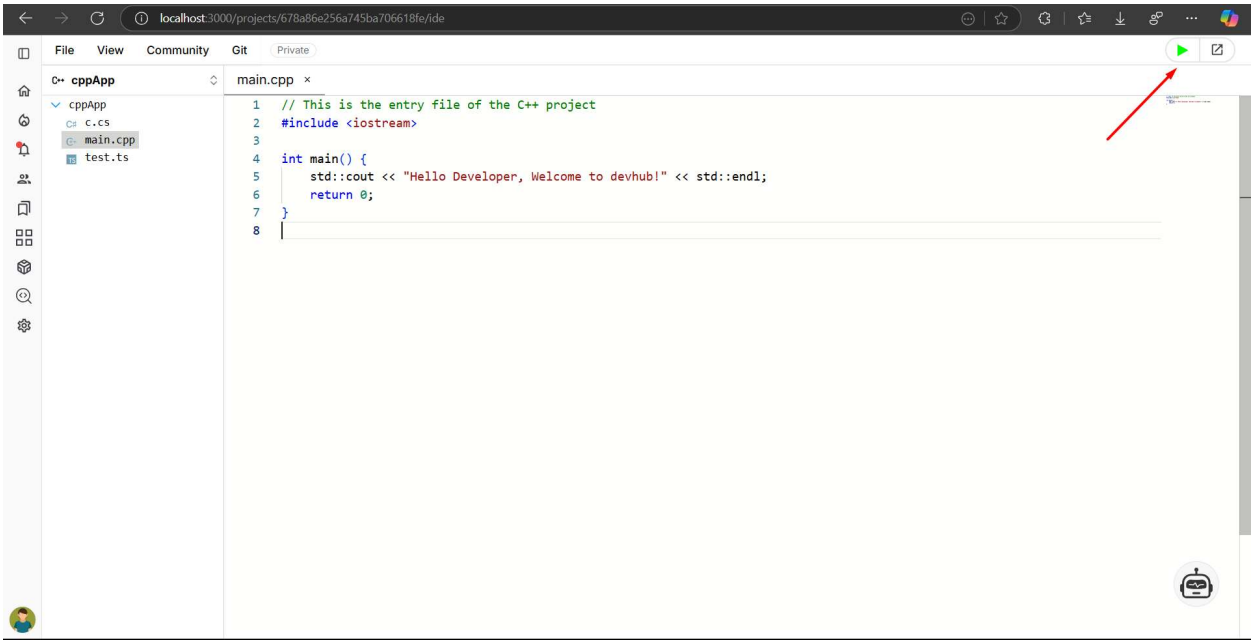

Figure 57: Run the project

- Run the project
  Also can run a raw code without render like c++ projects.

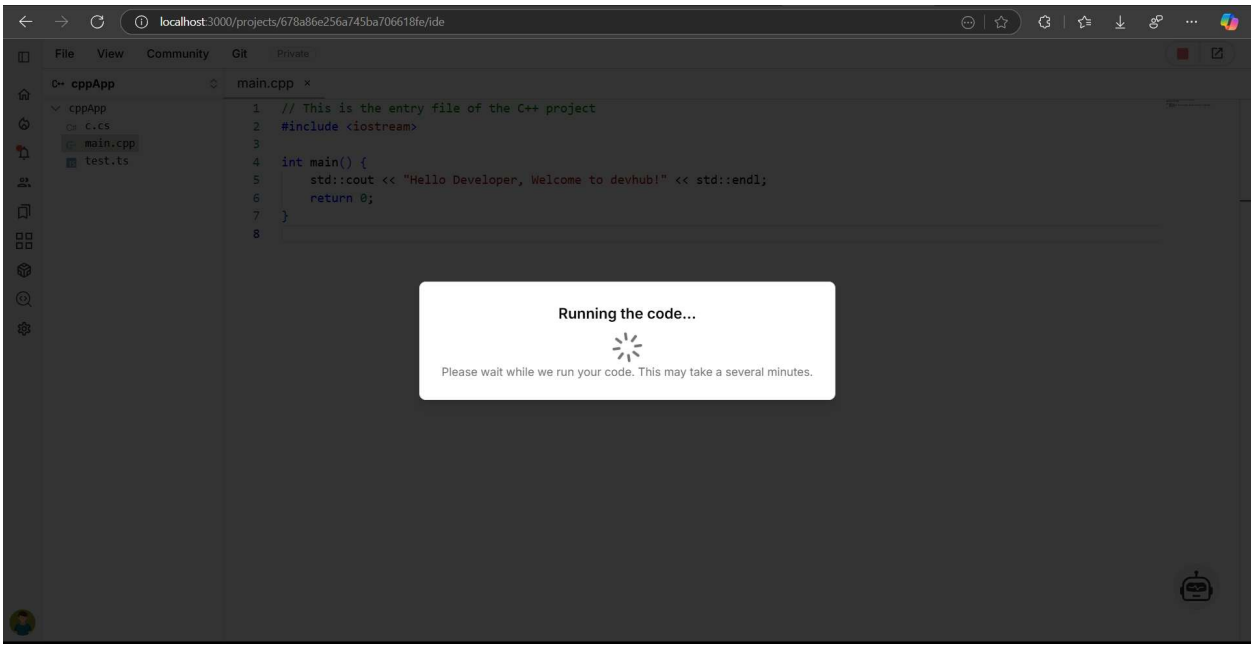

Figure 58: The project is running

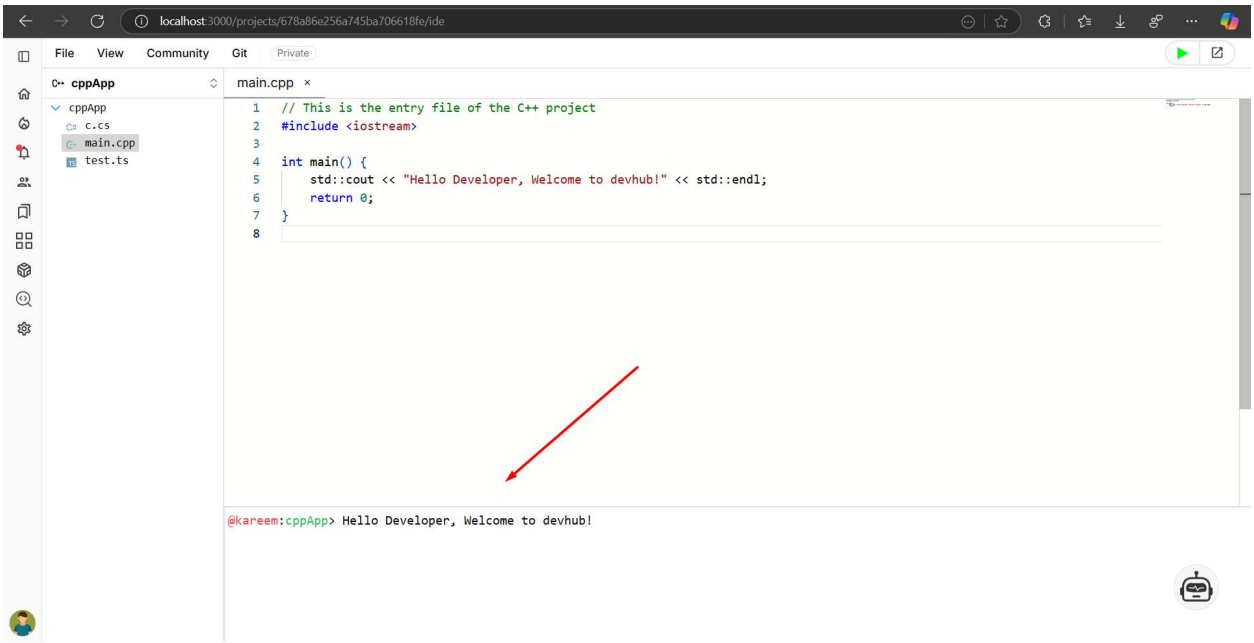

Figure 59: Project running completed

4.3.8 Project Publish

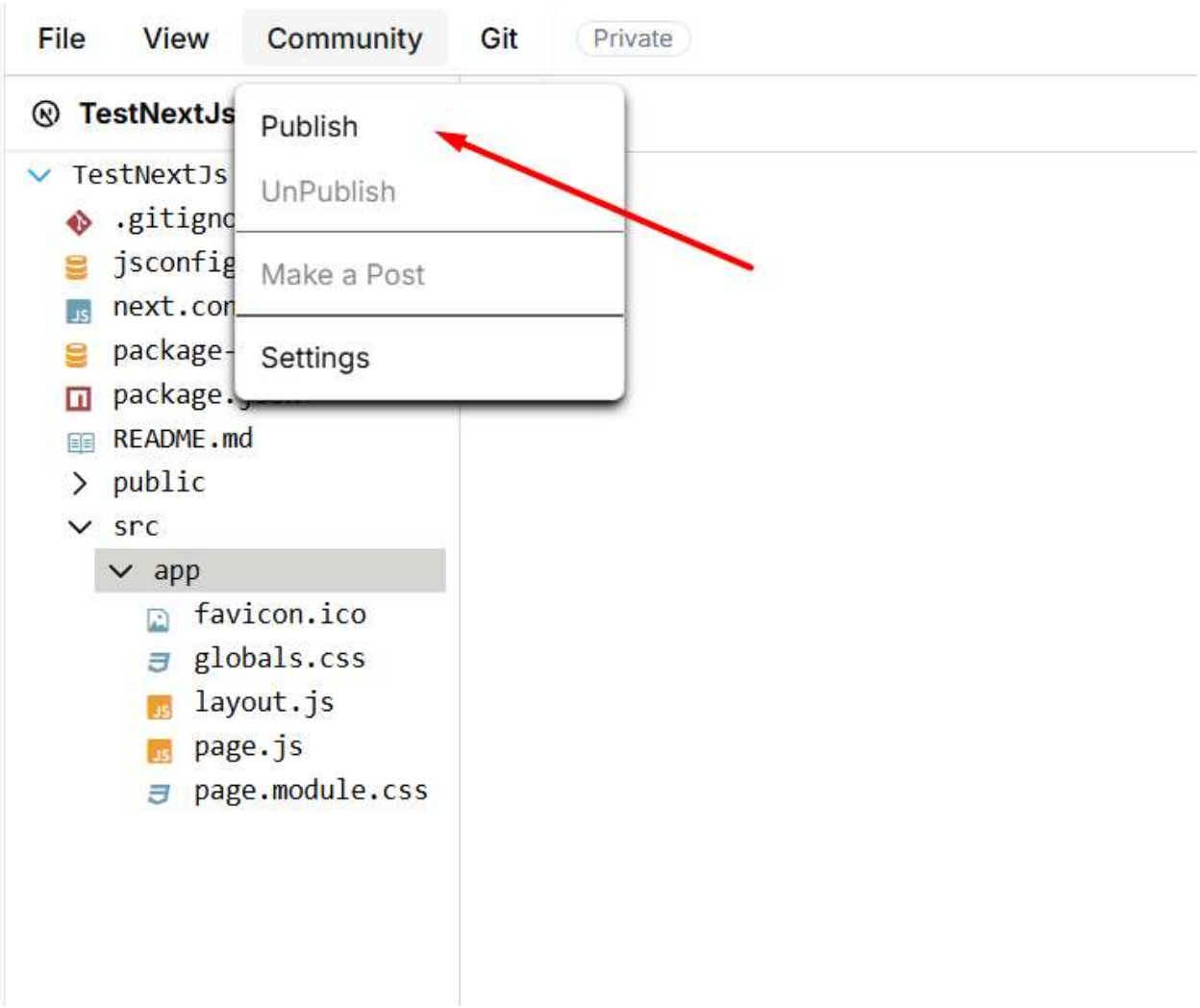

Figure 60: Publish a project

- Publish a project
  By clicking on Publish, the server will make snapshot of the project and make it public to everyone as Published Project. And everyone can access it and make a fork.

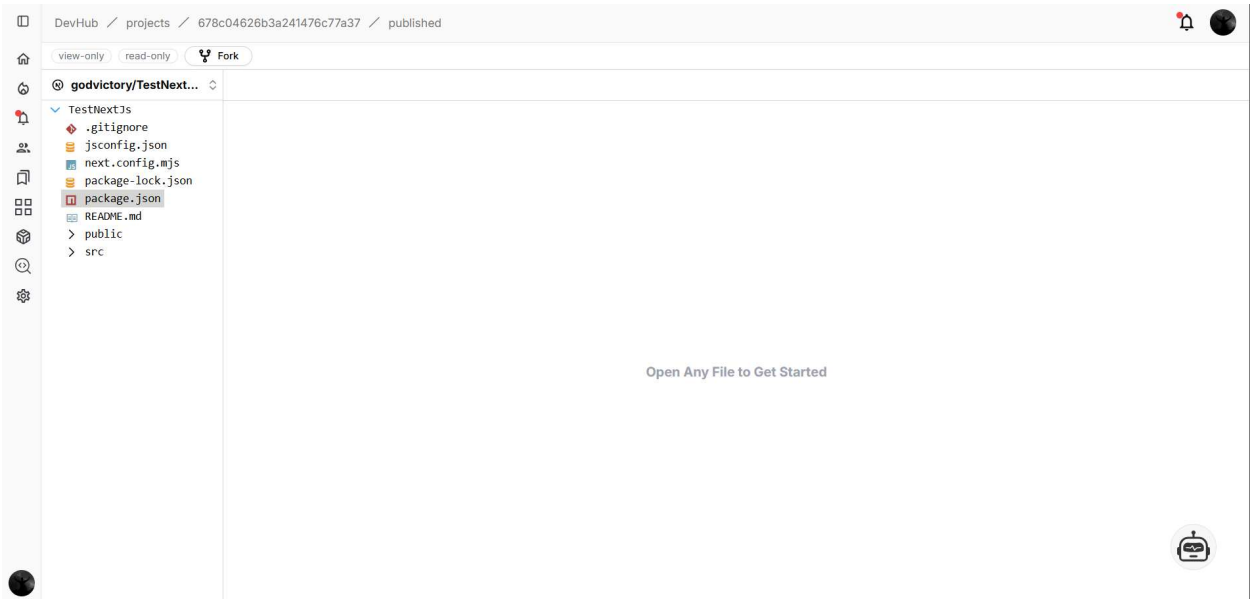

Figure 61: Published project

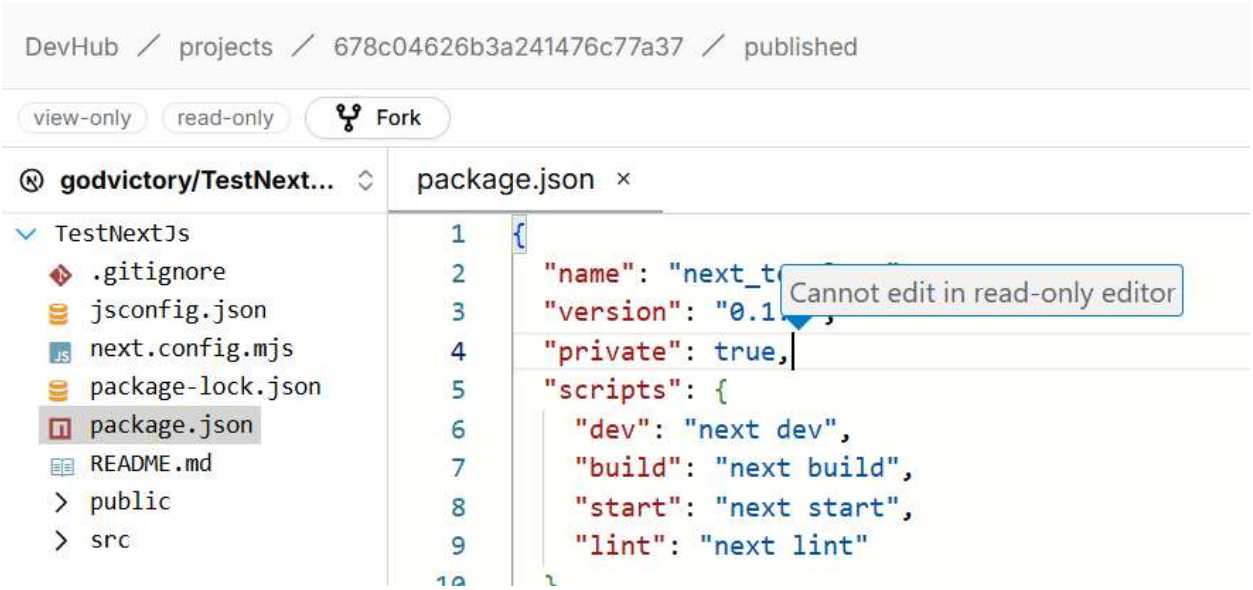

Figure 62: View-only project

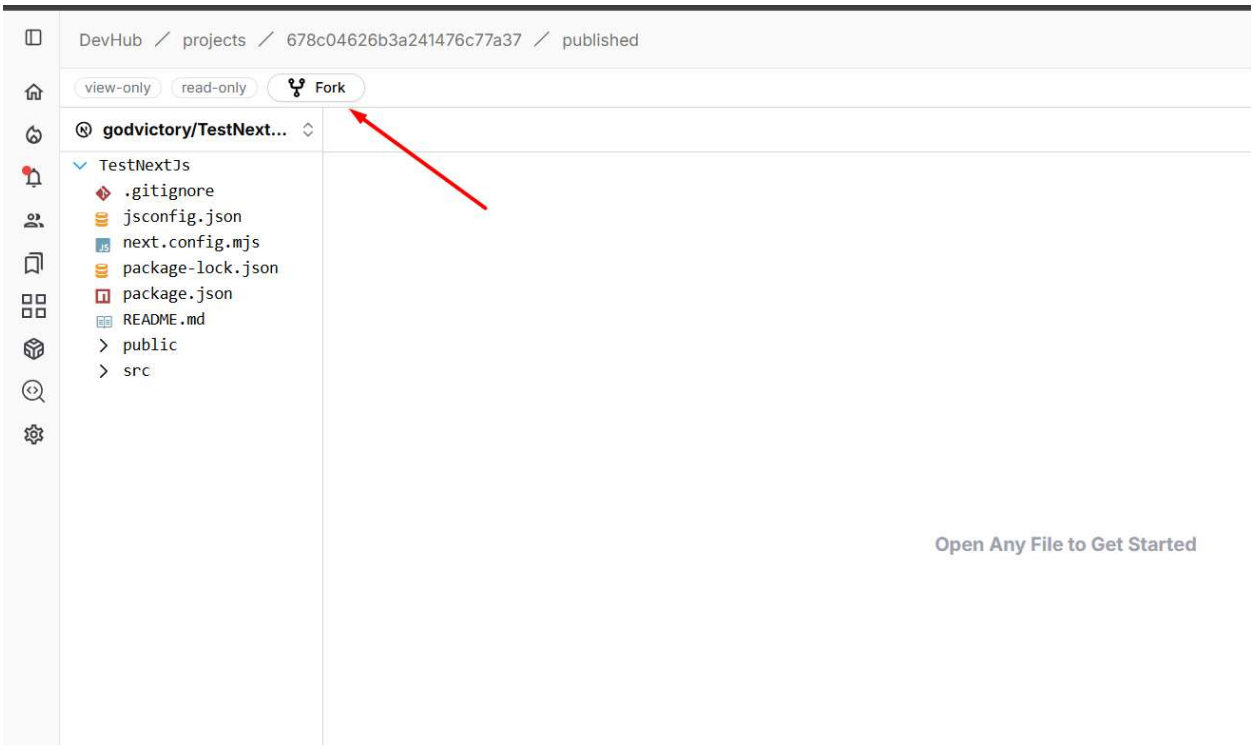

Figure 63: Fork a published project

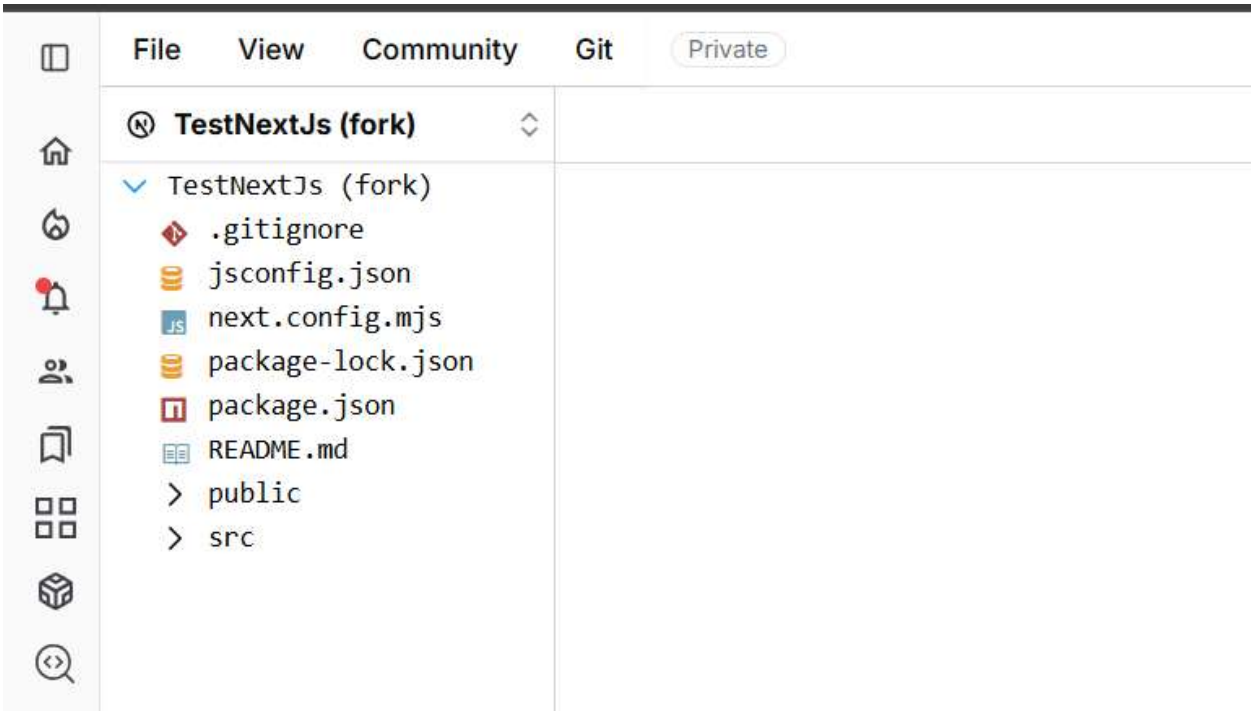

Figure 64: Forked project

- Project Fork
  The developer can fork any published project, and also public projects that allow the fork.

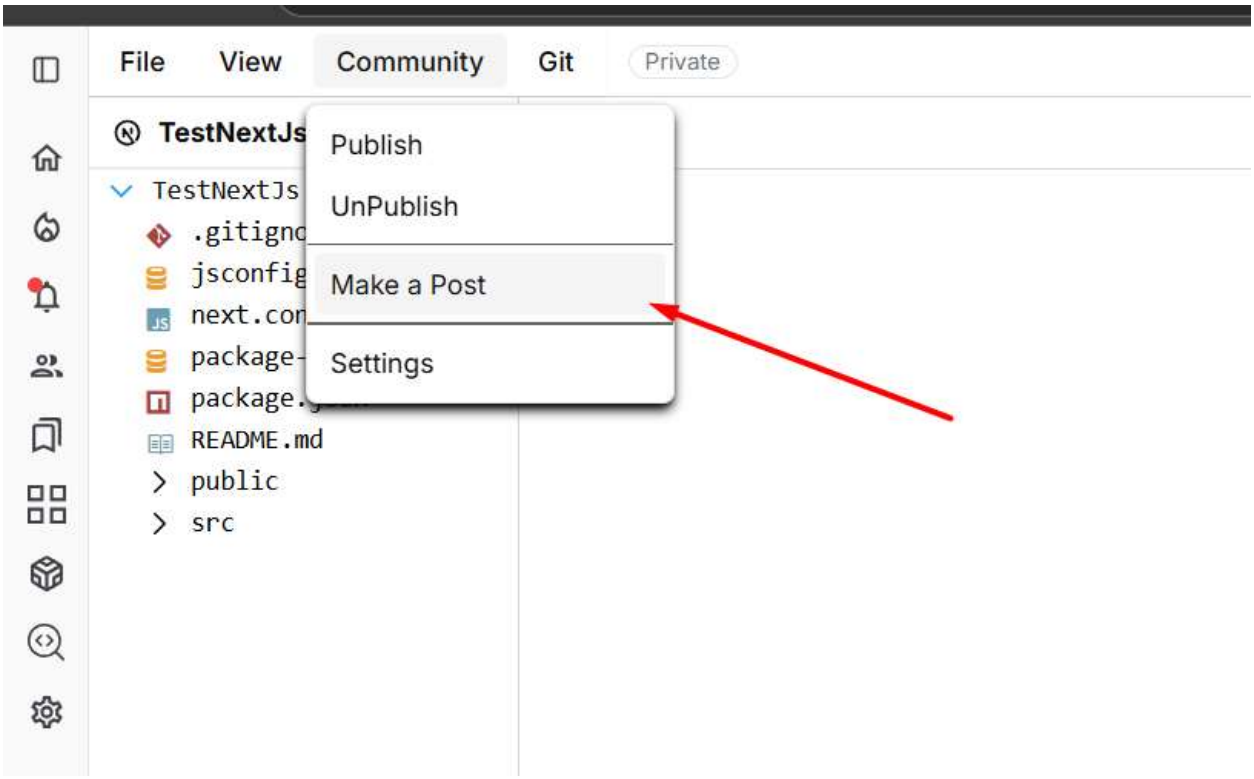

Figure 65: Post the published project

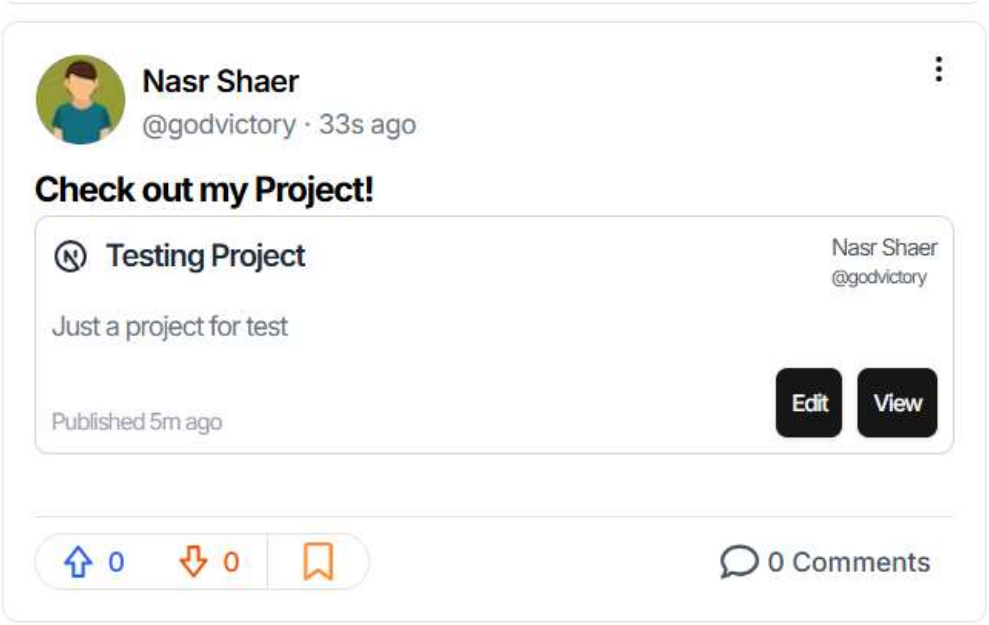

Figure 66: Published project's post

4.3.9 AI Assistant

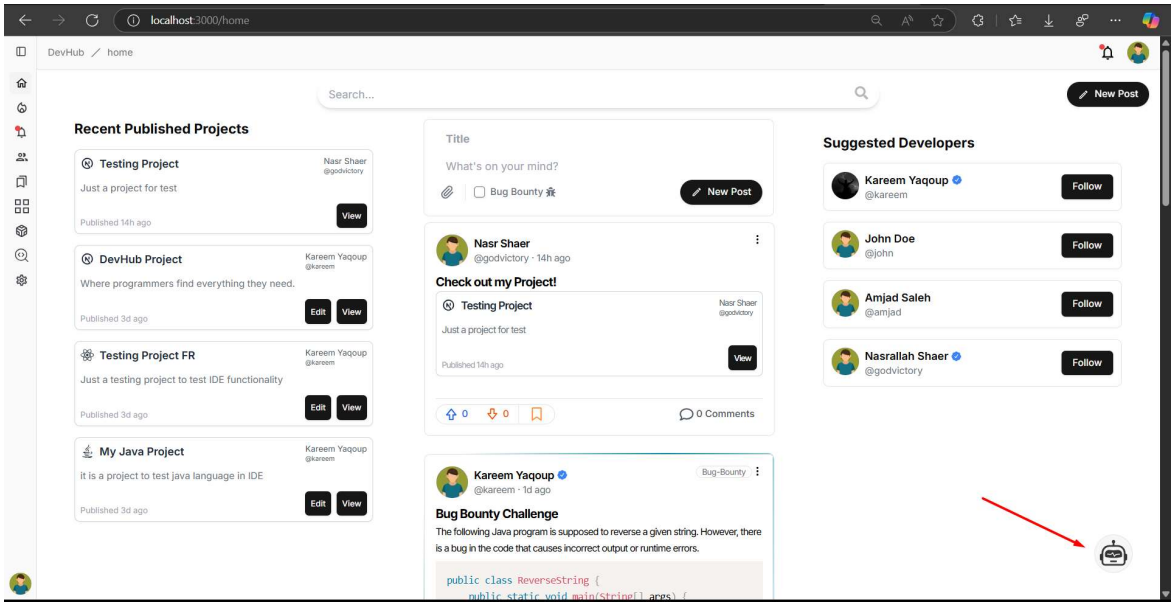

Figure 67: Ai Assistant

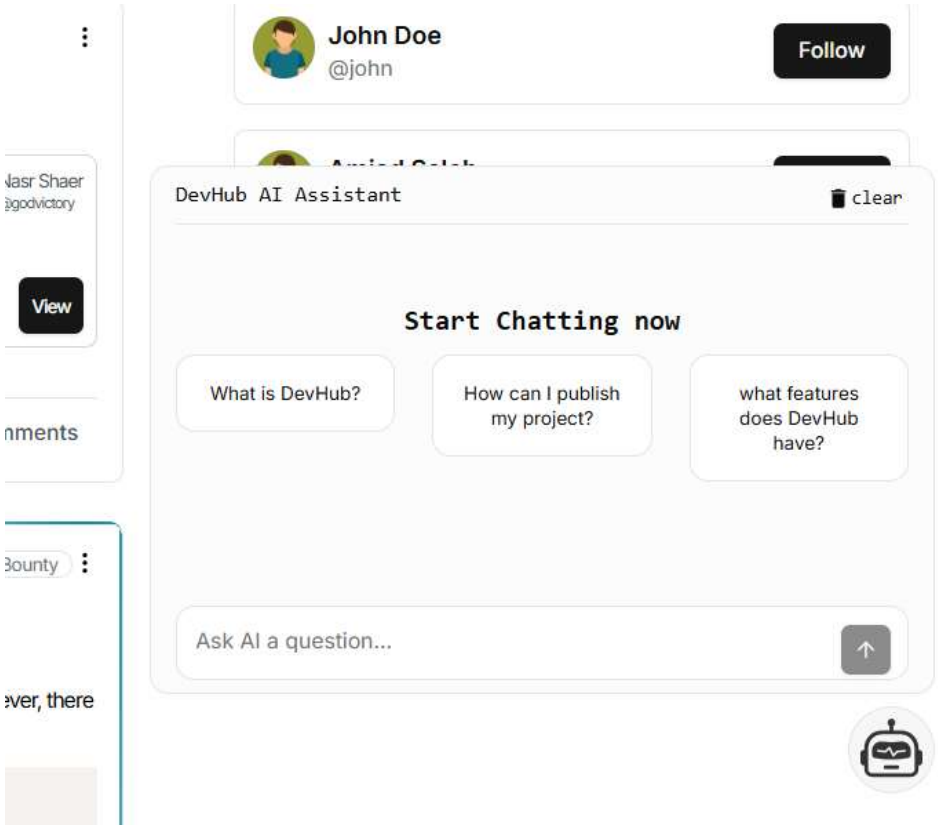

Figure 68: Ai Assistant Suggestions

- Ai Assistant
  Developer can chat with our Ai Assistant and you can ask him about anything, whether about the website in general or about programming.

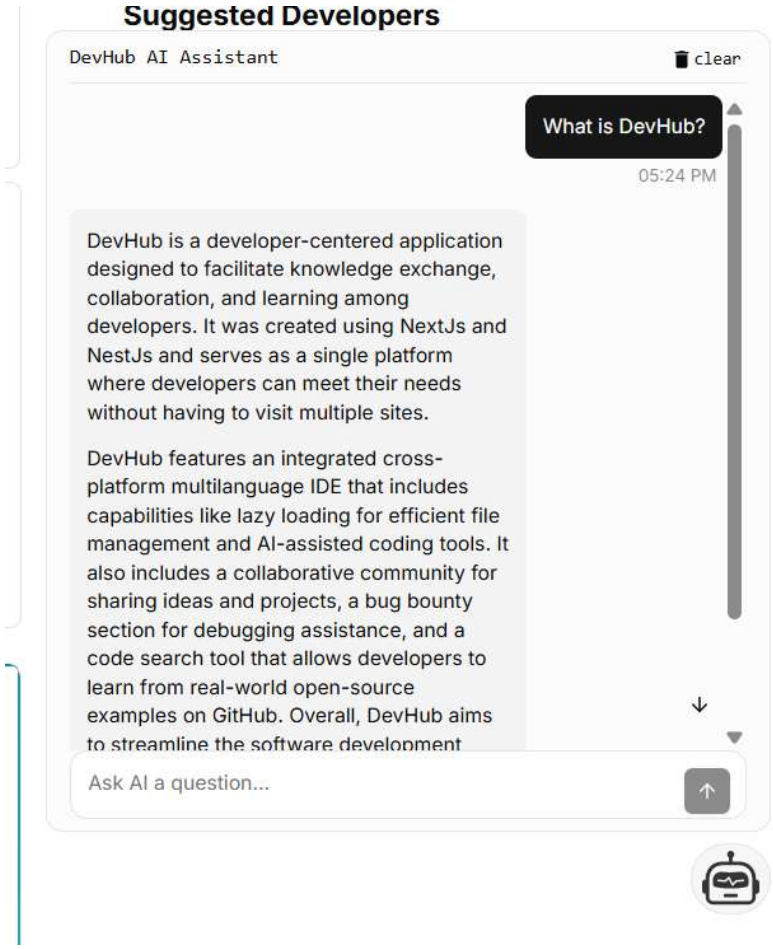

Figure 69: Ai Assistant Answer

- Ai Assistant in coding
  You can ask it about any code and you can even ask it about the code you program in our IDE feature.

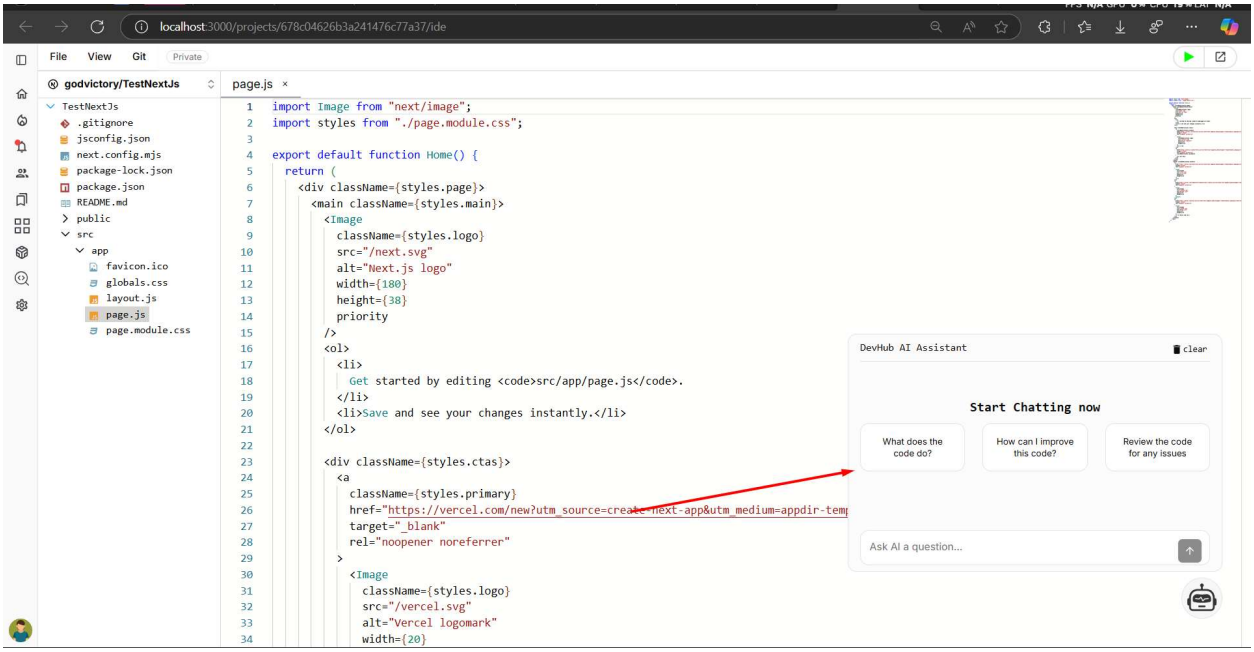

Figure 70: Ai Assistant Suggestion in IDE Page

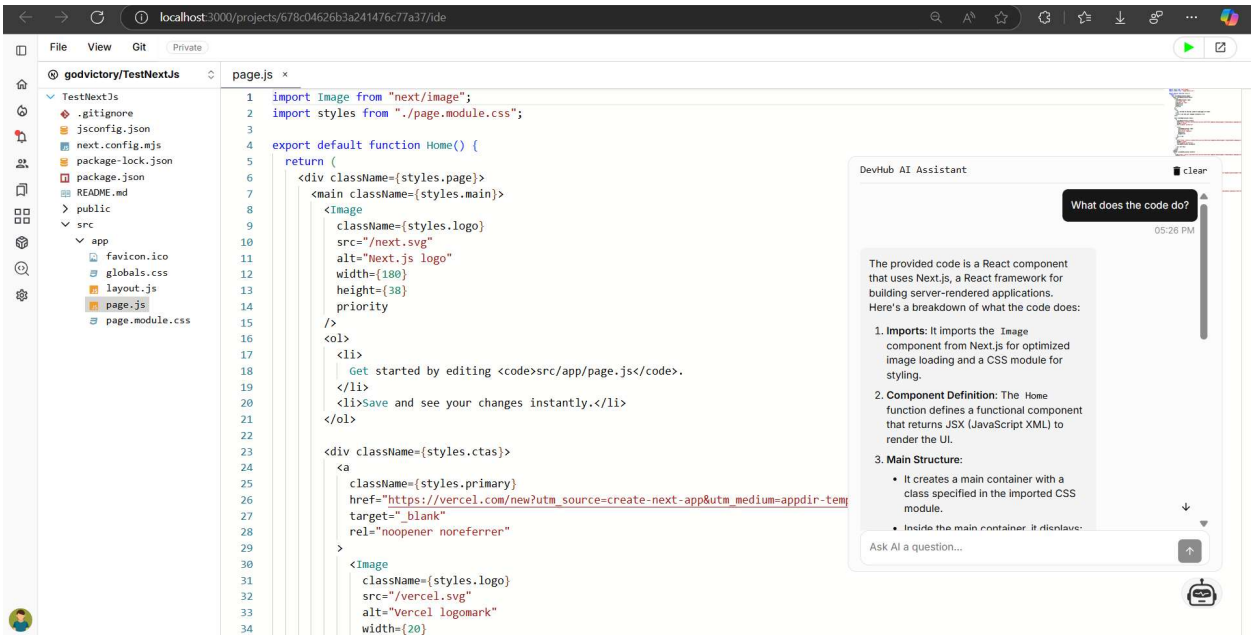

Figure 71: Ai Assistant Answer

4.3.10 Mobile User Profile

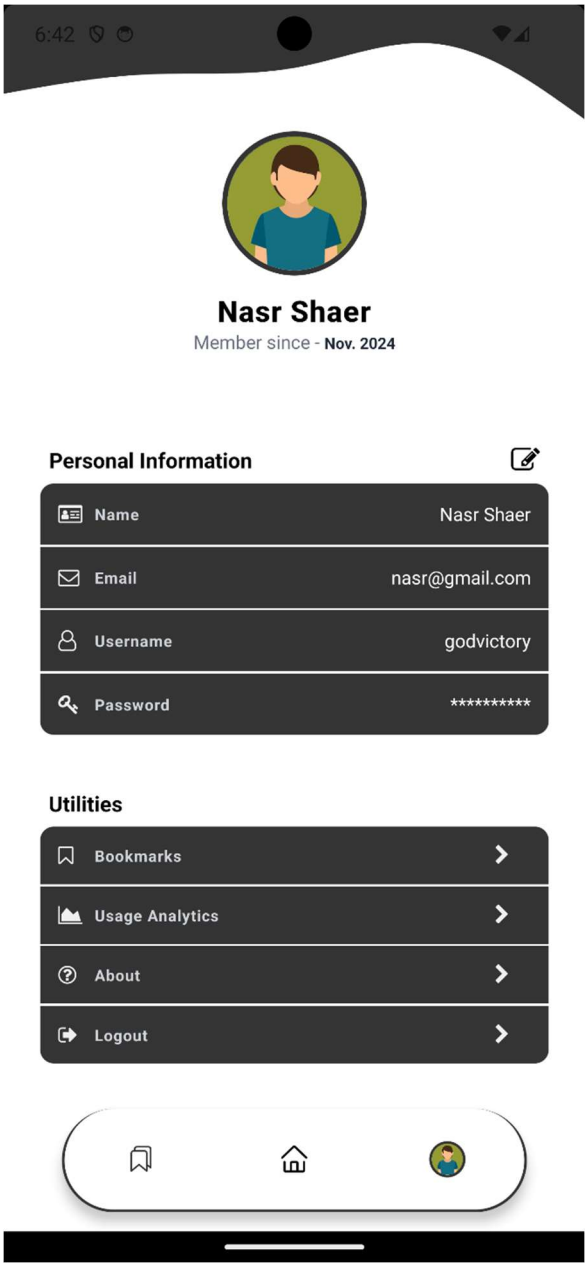

Figure 72 Mobile User Profile

5 Conclusion/ Recommendations and Future Works

In conclusion, by offering developers hub to enable developers to stay focused using our platform, offering the developer an integrated cloud IDE, Sandboxes to run different code/projects using various technologies and frameworks also enabling the integration with github for both importing/exporting, pull, commit and push into github respos.

DevHub community is more than just a place to share ideas, it also enable to share your projects or pieces of codes. Our bug hunter section allows the devs to get help from other developers which allow them to debug and find their bugs using DevHub IDE and earn points by solving others bugs.

For our future works, we aim to support more sandboxes, and terminal access to them. Also, integration with other software tools like notion and Jira. Moreover, having a desktop application for easier access to the platform.

Also, for the future additions, DevHub will include squads which is a place to share related things. For example, having a squad for frontend, or backend. Furthermore, Including AI roadmap generator, to generate personalized roadmap for each developer based on their skills and codes.

[5]

6 References

[1] R. A. a. L. T. a. B. G. Poldrack, "AI-assisted coding: Experiments with GPT-4," *arXiv preprint arXiv:2304.13187,* 2023.

[2] A. a. G. Y. a. B. T. a. A. I. Sergeyuk, "Using AI-based coding assistants in practice: State of affairs, perceptions, and ways forward," *Information and Software Technology,* vol. 178, p. 107610, 2025.

[3] A. a. B. S. Antonova, "An overview of the advantages of cloud computing and online IDE," *Automation of technological and business processes,* vol. 12, pp. 50--54, 2020.

[4] S. Bartkova, "Research of online IDE and analysis of directions of development," 2021.

[5] K. a. N. E. D. a. P. A. Kusumaningtyas, "Online integrated development environment (ide) in supporting computer programming learning process during covid-19 pandemic: A comparative analysis," *IJID (International Journal on Informatics for Development),* vol. 9, no. 2, pp. 66--71, 2020.