SANDIP
UNIVERSITY
— UGC Recognised —

School of Computer Sciences and Engineering
Department of Computer Science and Application

A PROJECT REPORT
ON

Coder Forge

UNDER THE FACULTY OF COMPUTER SCIENCE & APPLICATION

Submitted by

Omkar Shejwal
Vinit Mahale

Guided by
Dr. Pankaj Patil
Assistant Professor

Department of Computer Science and Application
School of Computer Sciences and Engineering
Sandip University, Nashik

BCA Sem-VI

May - 2024

SANDIP
UNIVERSITY
— UGC Recognised —

Trimbak Road, A/p - Mahiravani, Tal. & Dist. – Nashik, Pin – 422 213

School of Computer Sciences and Engineering
Department of Computer Science and Application

website: http://www.sandipuniversity.edu.in

CERTIFICATE

This is to certify that **Mr. Omkar Shejwal** , PRN No. 210105011037 Student of BCA Semester-VI has successfully completed the Project work on Title **Coder Forge** under my guidance. This report was submitted to the Department of Computer Science and Application, School of Computer Sciences and Engineering, Sandip University, Nashik in the AY 2023-24 for partial fulfillment and requirement for the End Semester Examination.

This report has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Dr. Pankaj Pati
PROJECT GUIDE

Dr. Vaibhav Sonaje
HOD

SANDIP
UNIVERSITY
UGC Recognised

Trimbak Road, A/p - Mahiravani, Tal. & Dist. – Nashik, Pin – 422 213

School of Computer Sciences and Engineering

Department of Computer Science and Application

website: http://www.sandipuniversity.edu.in

CERTIFICATE

This is to certify that **Mr .Vinit Mahale**, PRN No. 210105011029 Student of BCA Semester-VI has successfully completed the Project work on Title **Coder Forge** under my guidance. This report was submitted to the Department of Computer Science and Application, School of Computer Sciences and Engineering, Sandip University, Nashik in the AY 2023-24 for partial fulfillment and requirement for the End Semester Examination.

This report has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Dr. Pankaj Pati
PROJECT GUIDE

Dr. Vaibhav Sonaje
HOD

SANDIP
UNIVERSITY
— UGC Recognised —

Trimbak Road, A/p - Mahiravani, Tal. & Dist. – Nashik, Pin – 422 213

School of Computer Sciences and Engineering
Department of Computer Science and Application

website: http://www.sandipuniversity.edu.in

DECLARATION

We hereby declare that the Project work titled **Coder Forge** submitted to Sandip University, Nashik is a record of an original work done by me/us under the guidance of

Dr. Pankaj Pati Department of Computer Science and Application, School of Computer Sciences and Engineering, Sandip University in the AY 2023-24 for a partial fulfillment of requirement for the Completion of End Semester Examination.

This report has not been submitted to any other University or Institute for the award of any degree or diploma.

Student Name:
Mr. Omkar Shejwal
Mr. Vinit Mahale

Date: 06/05/2024
Place: Trimbak Road, A/p - Mahiravani, Tal. & Dist. – Nashik, Pin – 422 213

SANDIP
UNIVERSITY
— UGC Recognised —

Trimbak Road, A/p - Mahiravani, Tal. & Dist. – Nashik, Pin – 422 213

School of Computer Sciences and Engineering

Department of Computer Science and Application

website: http://www.sandipuniversity.edu.in

ACKNOWLEDGEMENT

We would like to express my/our special thanks to our Project guide **Dr.Pankaj Pati ,** Department of Computer Science and Application, School of Computer Sciences and Engineering, Sandip University, Nashik for his able guidance and support in completing this report.

We would like to extend my gratitude to **Dr Amol Potgantwar**, Dean, SOCSE, **Dr. Vaibhav Sonaje**, HOD, CSA for providing me/us with all the facilities that were required to complete this report successfully.

We also thank the management of Sandip University, Nashik for providing me/us infrastructure and lab facility for completion of this Project.

At last but not the least we are thankful to all faculty members, staff, and friends who have been always helping and encouraging me/us throughout the period of this internship/project.

Student Name:
Mr. Omkar Shejwal
Mr .Vinit Mahale

Abstract

The **Coder Forge** is a web-based application and Programming Language Learning Platform designed to facilitate the learning process for individuals interested in mastering various programming languages. The platform offers a comprehensive set of resources including programming language tasks, tests, and video lectures tailored to each user's learning needs and preferences.

The main features of the platform include:

1. **Diverse Programming Language Tasks:** Users can access a wide range of programming language tasks covering fundamental concepts and advanced topics in languages such as JavaScript, Python, Java, C++, and HTML/CSS. These tasks are designed to challenge users at different skill levels and provide hands-on practice opportunities.

2. **Interactive Testing Environment:** The platform provides an interactive testing environment where users can assess their proficiency in different programming languages through quizzes and assessments. These tests are dynamically generated based on the user's selected language and skill level.

3. **Personalized Learning Paths:** Users can create personalized learning paths based on their learning goals and preferences. The platform offers recommendations for relevant tasks and resources, allowing users to focus on areas where they need improvement and track their progress over time.

4. **Comprehensive Video Lectures:** In addition to tasks and tests, the platform offers a library of video lectures delivered by experienced instructors. These lectures cover a wide range of topics and provide in- depth explanations and demonstrations to help users understand complex concepts more effectively.

5. **User Profiles and Progress Tracking:** Each user has a dedicated profile where they can track their learning progress, view completed tasks, and access personalized recommendations. The platform also provides insights into users' strengths and weaknesses, helping them identify areas for further improvement.

Overall, the Programming Language Learning Platform (**Coder Forge**) aims to provide a dynamic and interactive learning experience that empowers users to master programming languages and advance their careers in the field of technology

TABLE OF CONTENT

List of Figure

Chapter 1: Introduction

1.1 Introduction:

In the current era, where technology is advancing at an unprecedented pace, proficiency in programming languages has become a crucial skill. This skill is not only essential for individuals aspiring to pursue careers in fields such as software development, data science, and artificial intelligence, but it is also increasingly becoming a valuable asset in a wide range of other disciplines. However, the journey to acquiring this skill can be quite challenging, especially for beginners. The complexity of coding concepts, coupled with the vast array of languages available, can make learning programming languages a daunting task.

Coder Forge is designed to address this challenge by providing a comprehensive, user-friendly, and interactive environment for individuals to learn, practice, and master different programming languages. This platform serves as a centralized hub where users can access a wide range of resources tailored to their learning needs. These resources include programming tasks that cover a broad spectrum of concepts, interactive tests that provide immediate feedback, video lectures that offer in-depth explanations, and personalized learning paths that adapt to their skill levels and learning objectives.

The primary goal of **Coder Forge** is to democratize access to programming education. It aims to break down barriers to learning and empower learners from all backgrounds to acquire valuable coding skills. By offering a diverse array of programming language tasks and assessments, the platform caters to users at various proficiency levels. Whether it's beginners seeking to grasp foundational concepts or advanced practitioners looking to refine their skills, **Coder Forge** has something for everyone.

One of the key features of **Coder Forge** is its interactive testing environment. This environment allows users to apply what they've learned in a practical setting, reinforcing their understanding of the concepts. Additionally, the platform offers personalized learning paths. These paths are designed to provide users with a structured and engaging learning experience. They adapt to the pace of the learner, ensuring that the learning process is not overwhelming. By tracking users' progress and offering targeted recommendations, the platform ensures that learners can effectively identify areas for improvement and track their advancement over time.

To further enhance the learning experience, **Coder Forge** includes comprehensive video lectures. These lectures, delivered by experienced instructors, provide in-depth explanations and demonstrations of key programming concepts. They serve as valuable supplements to the hands-on tasks and assessments offered on the platform, allowing learners to gain a deeper understanding of the material.

In conclusion, **Coder Forge** strives to foster a supportive and collaborative learning community where individuals can thrive and achieve their programming goals. It leverages the power of technology and education to create a platform that not only teaches programming but also equips users with the knowledge and skills they need to succeed in the ever-evolving field of programming and technology. Through its innovative approach to learning, **Coder Forge** endeavors to shape the future of programming education.

1.2 Existing System

In the current landscape of online learning platforms, there is a noticeable lack of universally accessible platforms that effectively assist individuals in understanding and rectifying their mistakes in programming. The existing systems, while numerous, often fall short in providing user-friendly interfaces and comprehensive feedback mechanisms. This deficiency makes the learning process not only challenging but also inefficient. Learners may find themselves struggling to navigate through complex interfaces or decipher vague feedback, which can hinder their progress and dampen their enthusiasm for learning.

1.3 Objective

The primary objective of this project is to bridge this gap by developing a platform that is universally accessible and works seamlessly on any system. This platform, tentatively named **Coder Forge**, aims to provide users with a user-friendly environment where they can learn programming, understand their mistakes, and correct them effectively. The platform seeks to enhance the learning experience by offering personalised feedback and recommendations based on users' performance, thereby fostering a more engaging and productive learning environment.

1.4 Scope of Work

The scope of work for this project encompasses the development of a web-based platform that is accessible from any device with internet connectivity. The platform will feature a variety of resources designed to facilitate learning, including interactive programming tasks, automated code evaluation mechanisms, personalized learning paths, and a community forum. The community forum will serve as a space for users to interact, share knowledge, and seek help, thereby fostering a collaborative learning environment.

1.5 Operating Environment

The platform will operate in a web-based environment and will be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge. This ensures that users can access the platform from any device, including desktops, laptops, tablets, and smartphones, thereby enhancing accessibility and convenience.

1.5.1 Software Specification

The software specification for the platform includes a combination of frontend and backend technologies. The frontend will be developed using HTML, CSS, JavaScript, and React.js, which will enable the creation of dynamic and interactive user interfaces. The backend will be powered by Node.js and Express.js, which will handle user requests, process data, and interact with the database. Other tools such as Git for version control and Docker for containerization will also be used to ensure efficient development and deployment processes.

1.5.2 Hardware Specification

Since the platform operates in a web-based environment, it does not have specific hardware requirements. However, for optimal performance, users should have devices capable of running modern web browsers and a stable internet connection.

1.6 Description Of Technology Used

The platform will leverage a combination of frontend and backend technologies to ensure a seamless user experience. React.js, a popular JavaScript library, will be used for the frontend to create dynamic and interactive user interfaces. On the backend, Node.js with Express.js will serve as the server to handle user requests, process data, and interact with the database. MongoDB, a NoSQL database, will be used to store user data and programming tasks, ensuring efficient data management and retrieval.

Chapter 2: Literature review

The literature review for this project delves into the exploration of existing educational platforms and tools that are specifically designed to teach programming languages. This exploration is not just a cursory glance, but a deep dive into the structure, functionality, and effectiveness of these platforms.

The review begins by examining the effectiveness of various learning methods employed by these platforms. These methods include but are not limited to interactive coding exercises, automated feedback systems, and personalized learning paths. Each of these methods is dissected to understand how they contribute to the learning process and what makes them effective.

Interactive coding exercises, for instance, are studied in terms of their design, the programming concepts they cover, and the level of engagement they offer to the learners. Automated feedback systems are analyzed based on their timeliness, accuracy, and the quality of feedback provided. Personalized learning paths are evaluated on their ability to adapt to the learner's pace, their flexibility in accommodating different skill levels, and their effectiveness in meeting individual learning goals.

Several studies that highlight the importance of providing immediate and constructive feedback to learners are also reviewed. These studies provide empirical evidence on how such feedback enhances learners' understanding and retention of programming concepts. They shed light on the psychological aspects of learning, such as the role of reinforcement, the importance of correcting misconceptions at an early stage, and the impact of feedback on learners' motivation.

In addition to studying the teaching methods, the review identifies the need for platforms that cater to diverse learning styles and skill levels. It discusses the principles of inclusive education and how they apply to online learning environments. It also highlights the challenges in implementing these principles, such as the difficulty in designing exercises that are universally accessible and creating an online community that is welcoming and supportive.

However, the review also uncovers a significant gap in the current landscape of educational platforms. Many existing platforms lack comprehensive feedback mechanisms and struggle to provide personalized learning experiences. Some platforms provide generic feedback that is not tailored to the learner's specific mistakes, while others use a one-size-fits-all approach to learning, ignoring the unique needs and preferences of individual learners.

This literature review underscores the importance of addressing these limitations in the development of Coder Forge. It sets the stage for the subsequent chapters, which will discuss the design and implementation of a platform that not only teaches programming but also provides a rich, personalized, and inclusive learning experience.

Chapter 3: Proposed System/ Methodology

3.1 User Requirement Specification

The user requirements for the system were identified through a meticulous process involving a series of surveys and interviews with potential users. These users ranged from beginners to advanced learners, ensuring a comprehensive understanding of the needs of different user groups. The key requirements that emerged from this process include:

- A user-friendly interface that is intuitive and easy to navigate, reducing the learning curve for new users and enhancing the overall user experience.
- Access to a variety of programming language tasks and assessments that cover a wide range of topics, allowing users to practice and reinforce their understanding of different programming concepts.
- Immediate and constructive feedback on programming tasks, enabling users to understand their mistakes, learn from them, and improve their coding skills.
- Personalized learning paths based on users’ skill levels and learning objectives, ensuring that the learning experience is tailored to meet the unique needs of each user.
- Comprehensive video lectures covering a wide range of programming topics, providing users with in-depth explanations and demonstrations of key concepts.
- User profiles with progress tracking and personalized recommendations, allowing users to monitor their learning progress and receive suggestions for further improvement.

3.2 Creation of a Dataset

A dataset was created by compiling a variety of programming tasks and assessments in different programming languages. This dataset was meticulously designed to cover fundamental concepts as well as advanced topics, catering to users at various proficiency levels. The tasks and assessments were sourced from reputable resources and carefully selected to ensure a comprehensive coverage of key programming concepts.

3.3 Pre-processing

The dataset underwent rigorous pre-processing to ensure consistency and quality. This involved removing duplicates, standardizing formats, and verifying the correctness of the tasks and assessments. The pre-processing stage was crucial in ensuring that the dataset was clean, accurate, and ready for further analysis.

3.4 Feature Extraction

Features were extracted from the pre-processed dataset to enable personalized learning paths and recommendations. These features included the difficulty level of tasks, the programming language used, and the user's performance history. The feature extraction process was instrumental in understanding the characteristics of the tasks and assessments, and how they relate to the users' learning progress.

3.5 Sequence Diagram

A sequence diagram was created to illustrate the flow of interactions in the proposed system. This diagram shows how users interact with the platform to access tasks, receive feedback, and track their progress. It provides a visual representation of the user journey, highlighting the key touchpoints and interactions within the system.

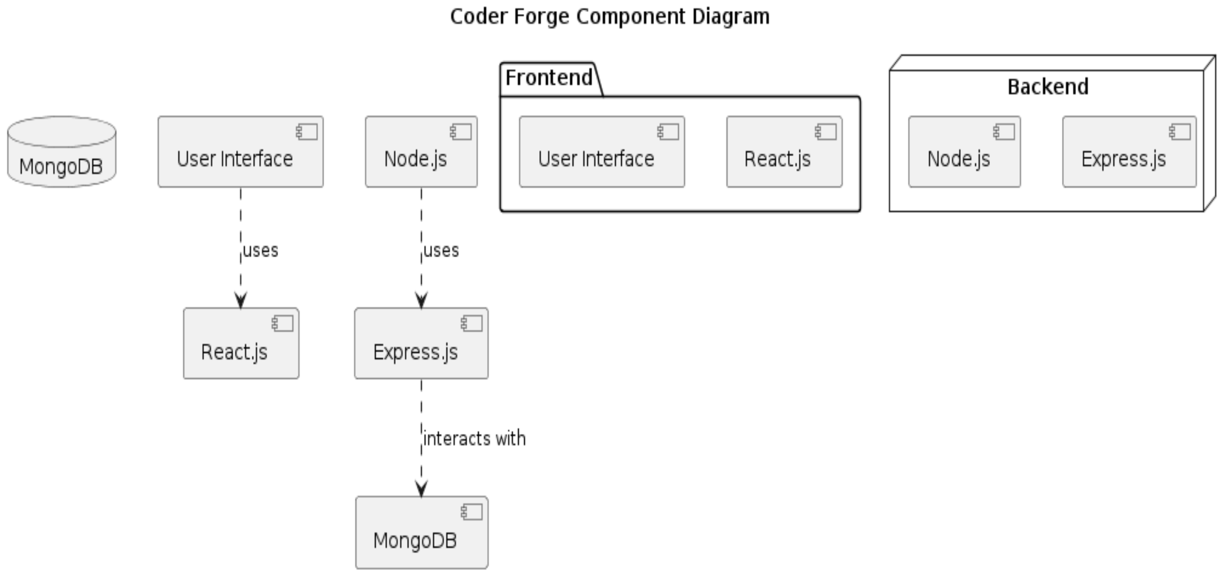


Fig No: 3.5

3.6 Deployment Diagram

This diagram represents the deployment of the Coder Forge system. The frontend, developed with React.js, provides the user interface which interacts with the web browser. The backend, developed with Node.js and Express.js, handles user requests and interacts with the MongoDB database. The entire system is accessible via the internet, allowing users to access the platform from any device with a modern web browser. Please note that this is a simplified representation and the actual deployment may involve more components and complexities.

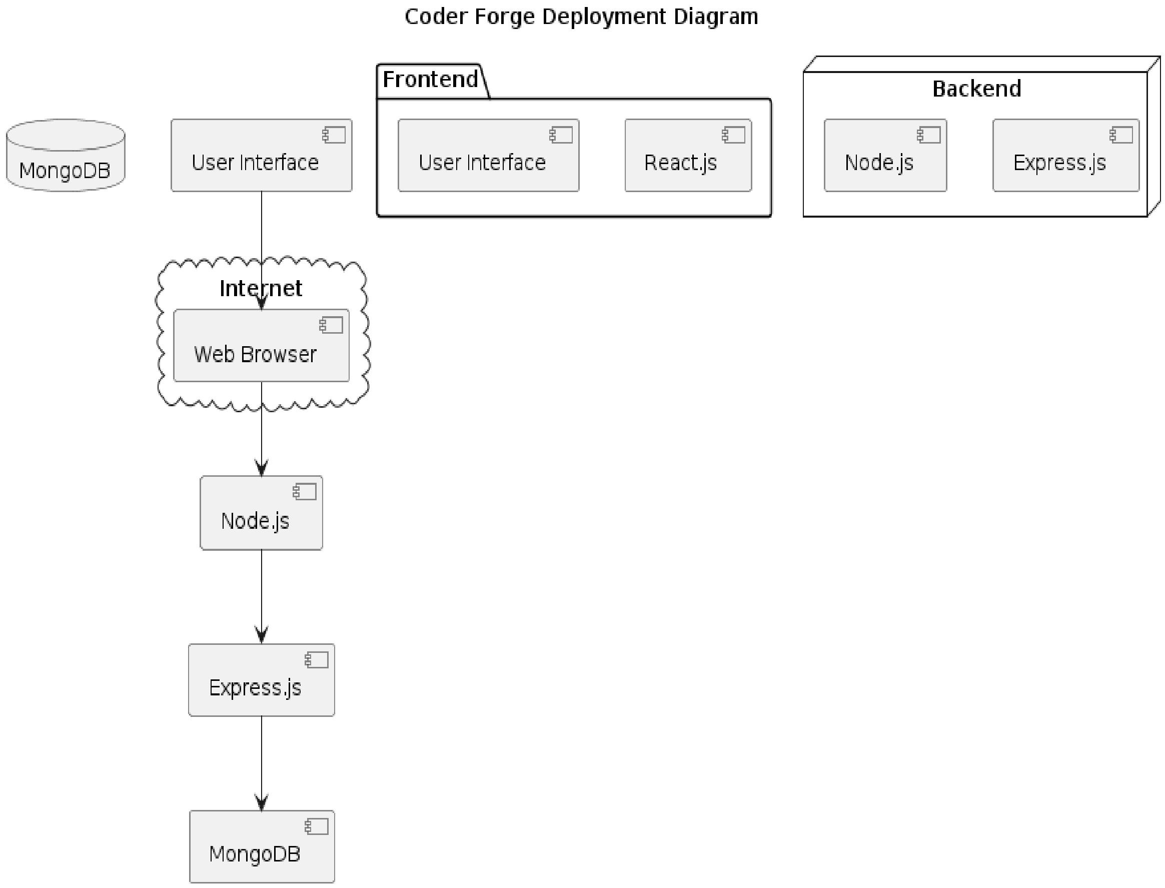


Fig No: 3.6

3.7 Component Diagram

This diagram represents the components of the Coder Forge system. The frontend, developed with React.js, provides the user interface. The backend, developed with Node.js and Express.js, handles user requests and interacts with the MongoDB database. Please note that this is a simplified representation and the actual system may involve more components and complexities.

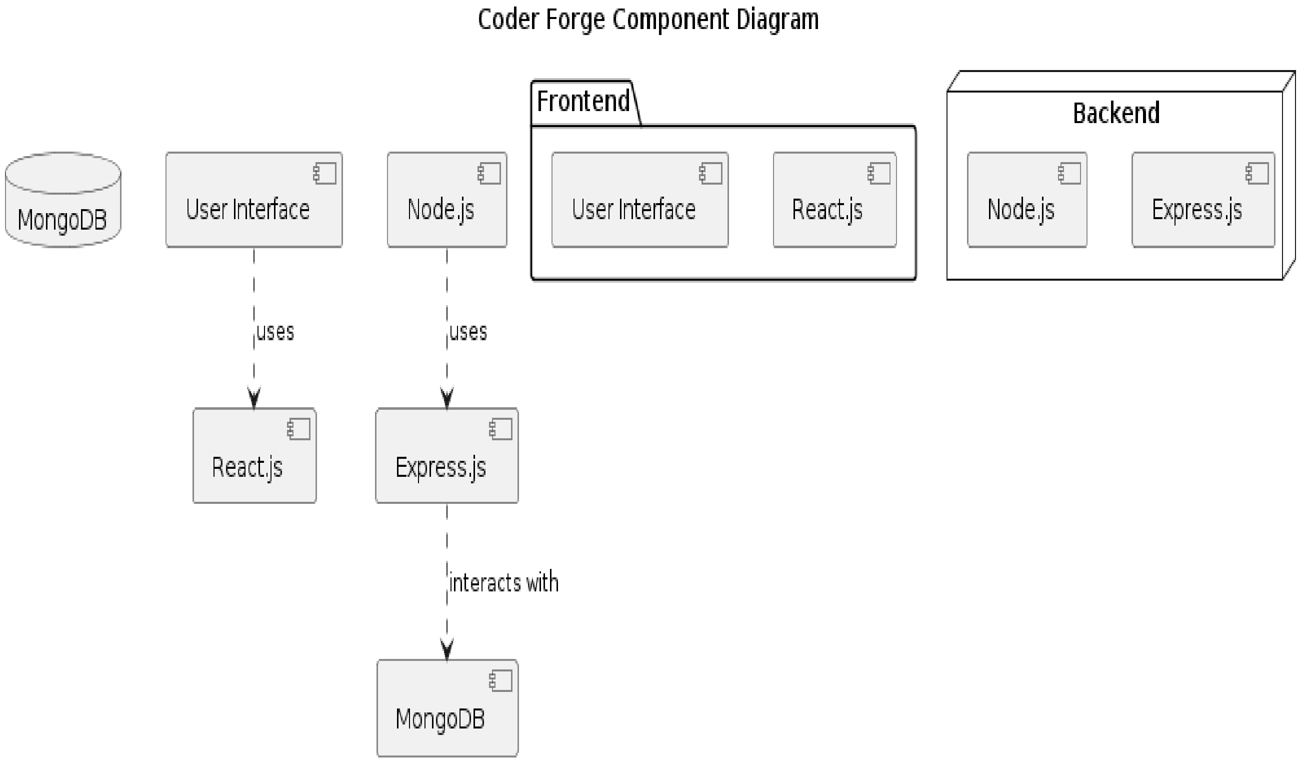


Fig No: 3.7

Chapter 4: Result and Implementation

4.1 Methods/Techniques Used

Several methods and techniques were employed during the implementation phase of the project, each contributing to the efficient and effective development of the **Coder Forge** platform:

Agile Development: The project was carried out using an agile development approach. This approach is characterized by iterative development, where the project is broken down into small parts called ‘sprints’. Each sprint results in a usable part of the software, which is then reviewed and improved upon in subsequent sprints. This method allows for regular feedback and flexibility to changes, ensuring that the final product is well-aligned with user needs and expectations.

Test-Driven Development (TDD): TDD is a software development approach where test cases are written before the actual code. In TDD, a failing test case is first written that defines a desired improvement or new function. Then, the minimum amount of code is written to pass that test, and finally the code is refactored to acceptable standards. TDD was used in this project to ensure the reliability and correctness of the code. It helped in maintaining a high standard of code quality and also made the debugging process easier.

Continuous Integration/Continuous Deployment (CI/CD): CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment. CI/CD practices were followed in this project to automate the integration and deployment of code, ensuring that the code in the repository is always in a deployable state.

Version Control (Git): Git is a distributed version control system that allows multiple developers to work on a project simultaneously without conflicts. It keeps track of changes made to the project and allows developers to revert back to any previous version of the project. Git was used for version control in this project, allowing for efficient collaboration among developers.

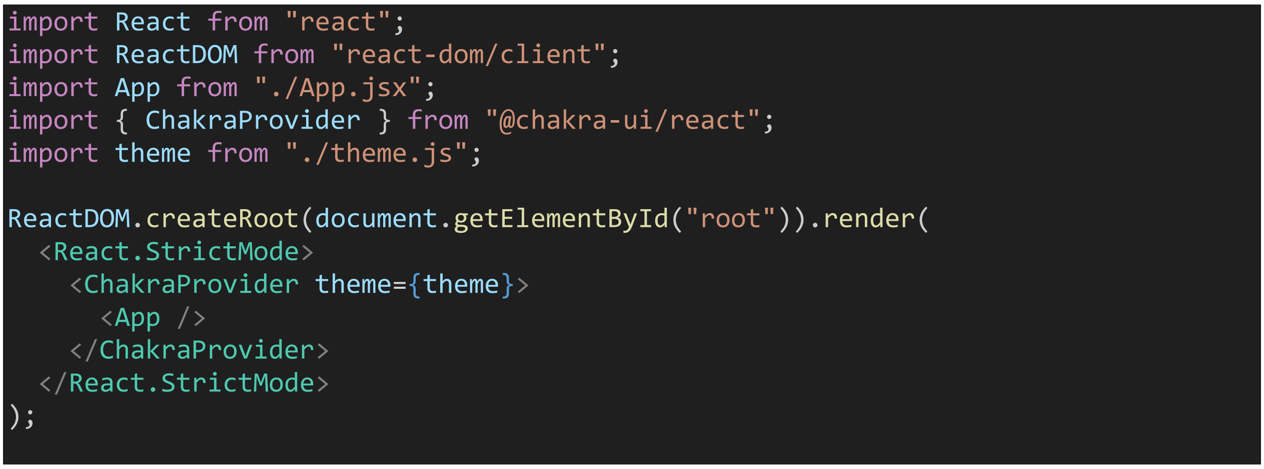
The implementation phase was crucial in turning the concepts and designs from the previous phases into a working platform. It involved a lot of coding, testing, debugging, and refining to ensure that the platform functions as intended and provides a seamless user experience. The result is a comprehensive and user-friendly platform that effectively helps individuals learn programming languages. This phase was marked by rigorous testing and quality assurance to ensure the reliability and robustness of the platform. The use of these methods and techniques ensured a systematic and efficient approach to the development of the platform, resulting in a high-quality end product.

4.2 Implementation

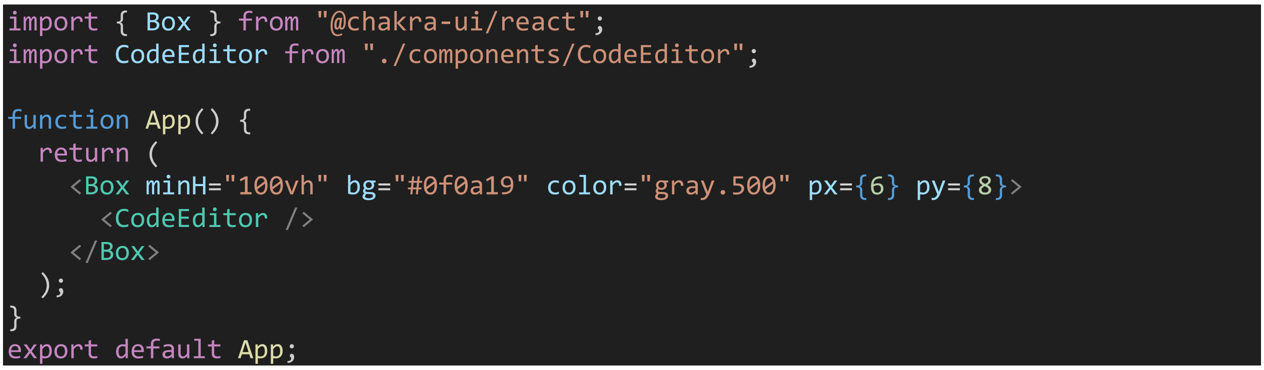
The implementation phase of the project involved the actual coding and development of the **Coder Forge** platform. This phase was carried out using a combination of frontend and backend technologies, as well as database management systems. Here’s a more detailed breakdown:

Frontend Implementation (React.js)

The frontend of the platform was implemented using React.js, a popular JavaScript library for building user interfaces. The user interface was designed to be user-friendly and intuitive, with clear navigation and interactive elements. It includes components for displaying programming tasks, providing feedback, tracking progress, and managing user profiles. Each component was carefully designed and tested to ensure a seamless user experience. The use of React.js allowed for efficient development of reusable components, improving the consistency and maintainability of the code.



```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import { ChakraProvider } from "@chakra-ui/react";
import theme from "./theme.js";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <ChakraProvider theme={theme}>
      <App />
    </ChakraProvider>
  </React.StrictMode>
);
```



```
import { Box } from "@chakra-ui/react";
import CodeEditor from "./components/CodeEditor";

function App() {
  return (
    <Box minH="100vh" bg="#0f0a19" color="gray.500" px={6} py={8}>
      <CodeEditor />
    </Box>
  );
}
export default App;
```

Backend Implementation (Node.js, Express.js)

The backend of the platform was implemented using Node.js, a JavaScript runtime built on Chrome’s V8 JavaScript engine, and Express.js, a minimal and flexible Node.js web application framework. The backend handles user requests, processes data, and interacts with the database. It includes modules for managing tasks, handling submissions, generating personalized learning paths, and providing recommendations. Each module was developed with careful consideration of performance and scalability, ensuring that the platform can handle a large number of users and tasks without compromising on speed or reliability.

Database Implementation (MongoDB)

MongoDB, a source-available cross-platform document-oriented database program, was used for data storage. It stores user data, programming tasks, and other relevant information. The database was designed to ensure efficient data management and retrieval, with appropriate indexing and data validation rules. The use of MongoDB allowed for flexible data modeling and efficient querying, making it an ideal choice for storing the diverse and complex data associated with the platform.

In addition to these technologies, the implementation phase also involved setting up a robust development environment, establishing coding standards and guidelines, and setting up version control and CI/CD pipelines. Regular code reviews and testing were carried out to ensure the quality of the code and the reliability of the platform. The result is a comprehensive and user-friendly platform that effectively helps individuals learn programming languages.

```
const MongoClient = require('mongodb').MongoClient;
const url = 'mongodb://localhost:27017';

// Database Name
const dbName = 'Coder Forge';

// Create a new MongoClient
const client = new MongoClient(url, { useUnifiedTopology: true });

// Use connect method to connect to the server
client.connect(function(err) {
  if (err) {
    console.error('An error occurred connecting to MongoDB: ', err);
  } else {
    console.log("Connected successfully to server");

    const db = client.db(dbName);
    client.close();
  }
})
```

4.3 Entity Relationship Diagram

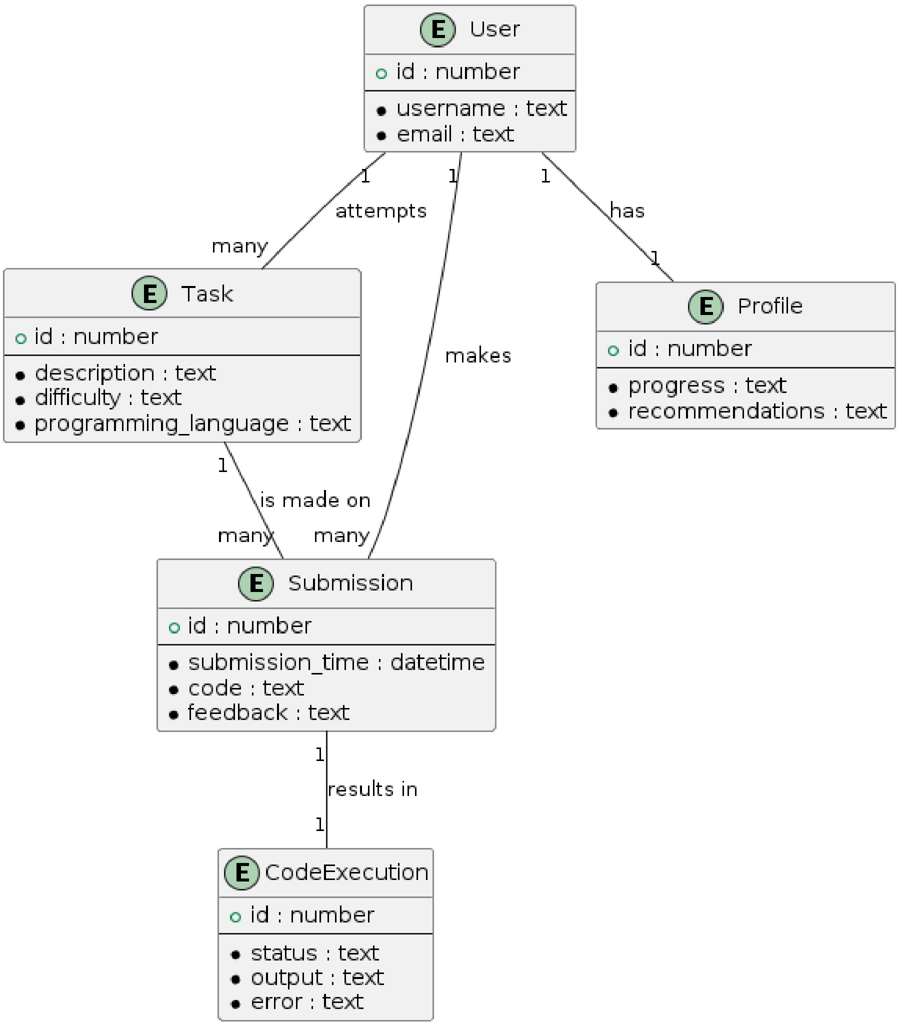


Fig No: 4.3

In this diagram:

- User, Task, Submission, Profile, and CodeExecution are entities.
- Each entity has attributes. For example, User has id, username, and email.
- The relationships between entities are represented by lines. For example, a User attempts Task, makes Submission, and has Profile.
- A new entity CodeExecution is added to represent the result of running the code. It has a status attribute which can be ‘success’ or ‘error’, an output attribute for the output of the code when it runs successfully, and an error attribute for the error message when the code fails to run.
- A new relationship results in is added between Submission and CodeExecution to represent that a submission results in a code execution

4.4 UML diagram

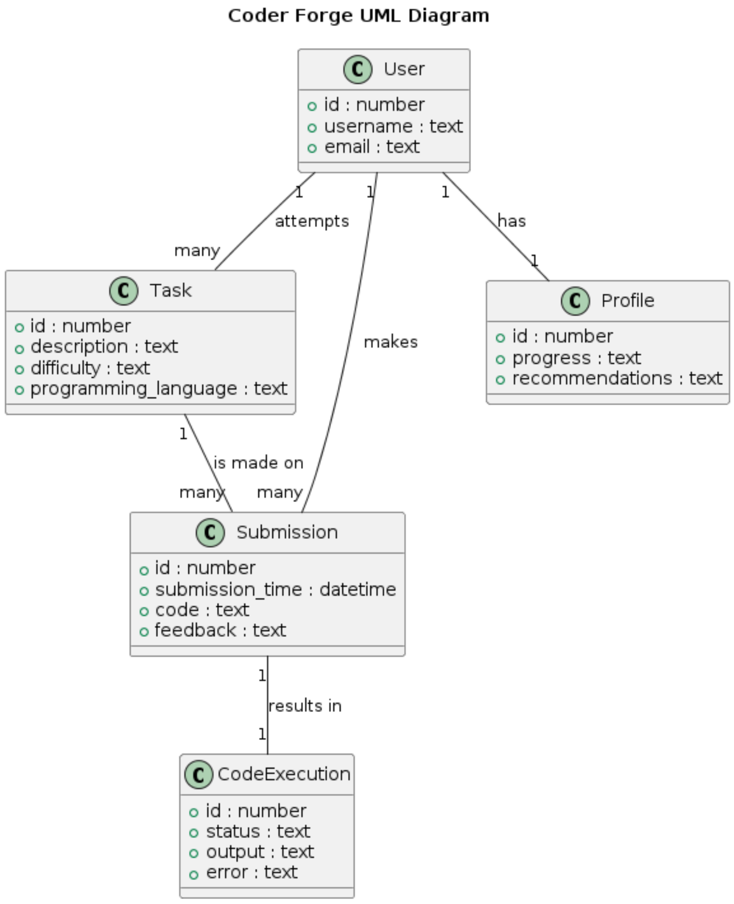


Fig No: 4.4

In this diagram:

- `User`, `Task`, `Submission`, `Profile`, and `CodeExecution` are classes.
- Each class has attributes. For example, `User` has `id`, `username`, and `email`.
- The relationships between classes are represented by lines. For example, a `User` attempts `Task`, makes `Submission`, and has `Profile`.
- A new class `CodeExecution` is added to represent the result of running the code. It has a `status` attribute which can be ‘success’ or ‘error’, an `output` attribute for the output of the code when it runs successfully, and an `error` attribute for the error message when the code fails to run.
- A new relationship `results in` is added between `Submission` and `CodeExecution` to represent that a submission results in a code execution.

4.5 Data flow diagram

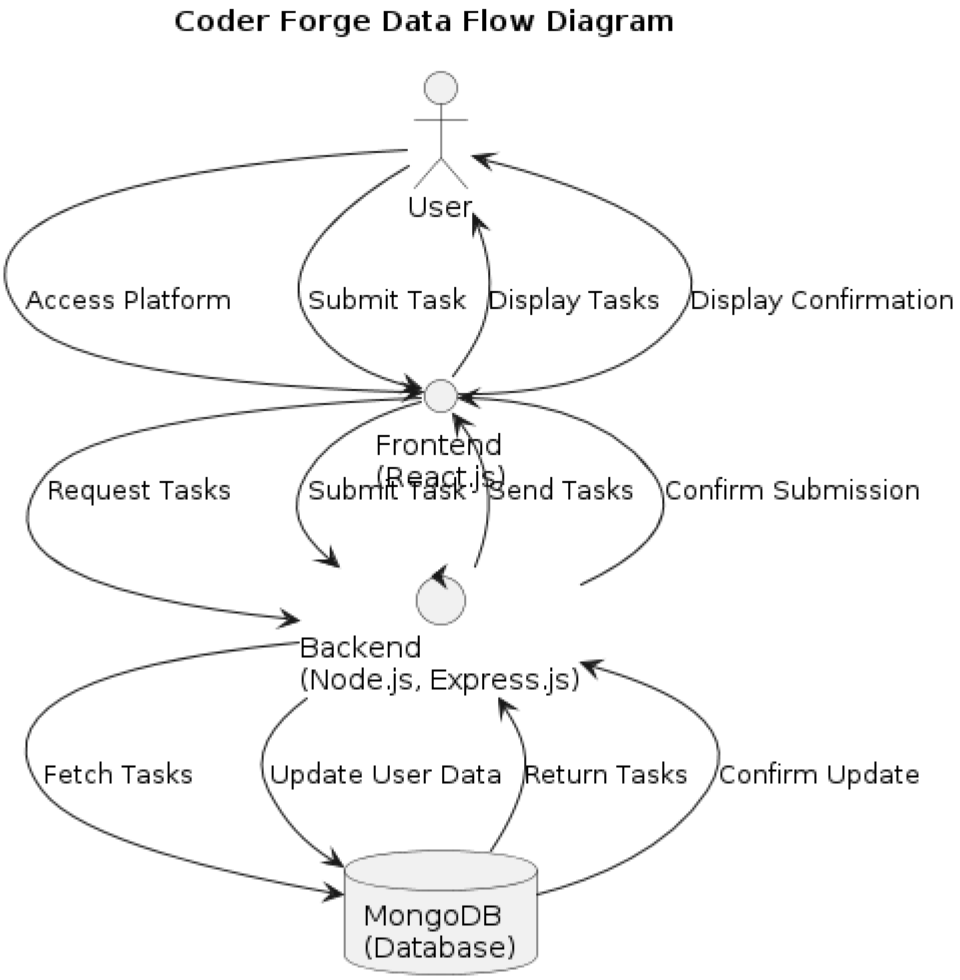


Fig No: 4.5

In this diagram:

- `User`, `Frontend`, `Backend`, and `Database` are the main components of the system.
- The arrows represent the flow of data between these components. For example, the user accesses the platform and requests tasks, which are fetched from the database and displayed to the user. When the user submits a task, the user data in the database is updated, and a confirmation is displayed to the user.

4.6 Use case diagram

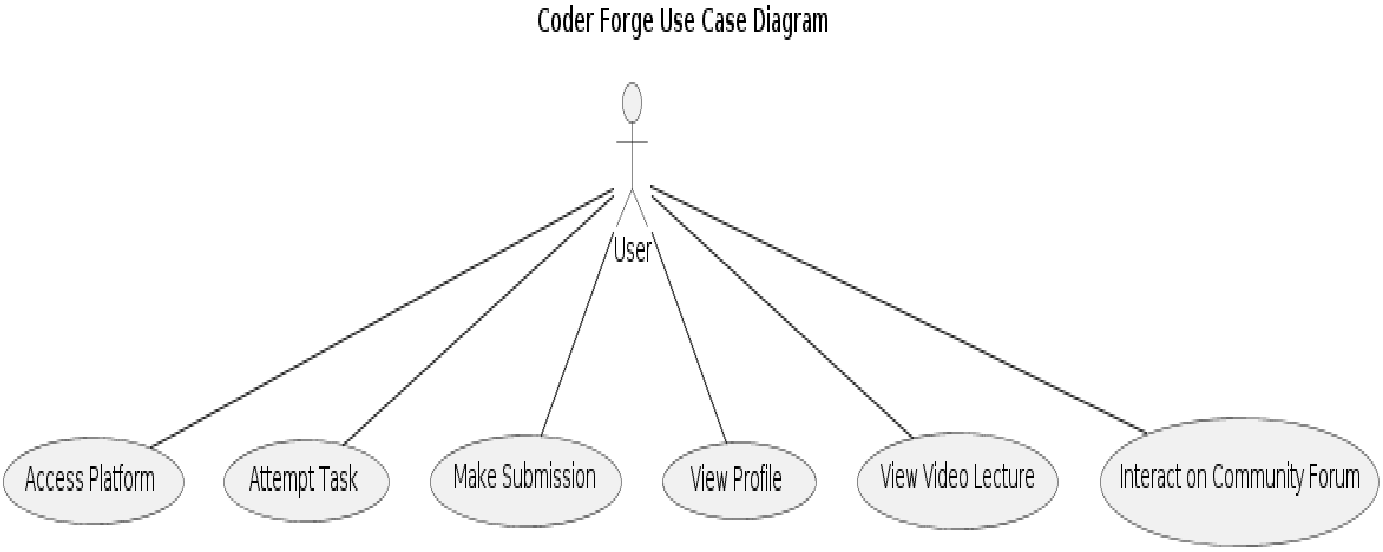


Fig No: 4.6

In this diagram:

- `User` is the actor who interacts with the system.
- The use cases are the actions that the user can perform on the system. These include `Access Platform`, `Attempt Task`, `Make Submission`, `View Profile`, `View Video Lecture`, and `Interact on Community Forum`.

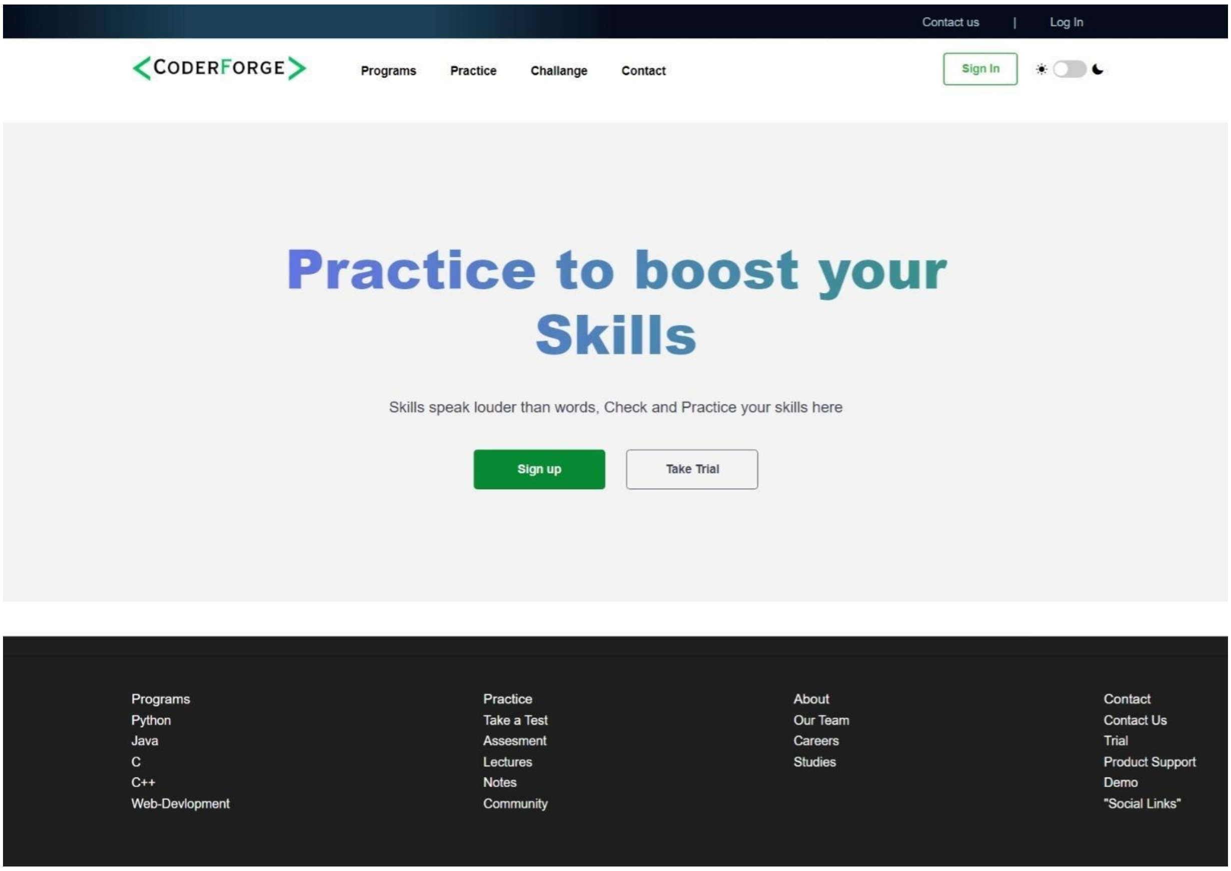
4.7 Module specification

1. **User Management Module**: This module handles all operations related to user management. It includes features for user registration, login, profile management, and progress tracking. It ensures that each user has a unique identifier and that their data is securely stored and easily retrievable.
2. **Task Management Module**: This module manages the programming tasks available on the platform. It includes features for task creation, categorization, and retrieval. It ensures that tasks are properly categorized by difficulty level and programming language, and that they cover a wide range of programming concepts.
3. **Submission Management Module**: This module handles the submissions made by users. It includes features for submission creation, validation, and feedback generation. It ensures that each submission is linked to a specific task and user, and that feedback is generated based on the correctness of the submission.
4. **Learning Path Management Module**: This module manages the personalized learning paths for users. It includes features for learning path generation, update, and retrieval. It ensures that each learning path is tailored to the user's skill level and learning objectives, and that it adapts over time based on the user's progress.
5. **Video Lecture Management Module**: This module manages the video lectures available on the platform. It includes features for video lecture creation, categorization, and retrieval. It ensures that the video lectures cover a wide range of programming topics and that they are easily accessible to users.
6. **Code Execution Module**: This module handles the execution of user-submitted code. It includes features for code execution, error handling, and output generation. It ensures that the code is executed in a secure environment and that the output or error message is returned to the user.
7. **Community Forum Module**: This module manages the community forum where users can interact and seek help. It includes features for post creation, comment creation, and post retrieval. It ensures that users can easily share knowledge and support each other in their learning journey.

4.8 User Interface Design

- **Homepage**: Screenshots showcasing the layout, design, and content of the homepage:

Fig 4.8.1 Homepage:

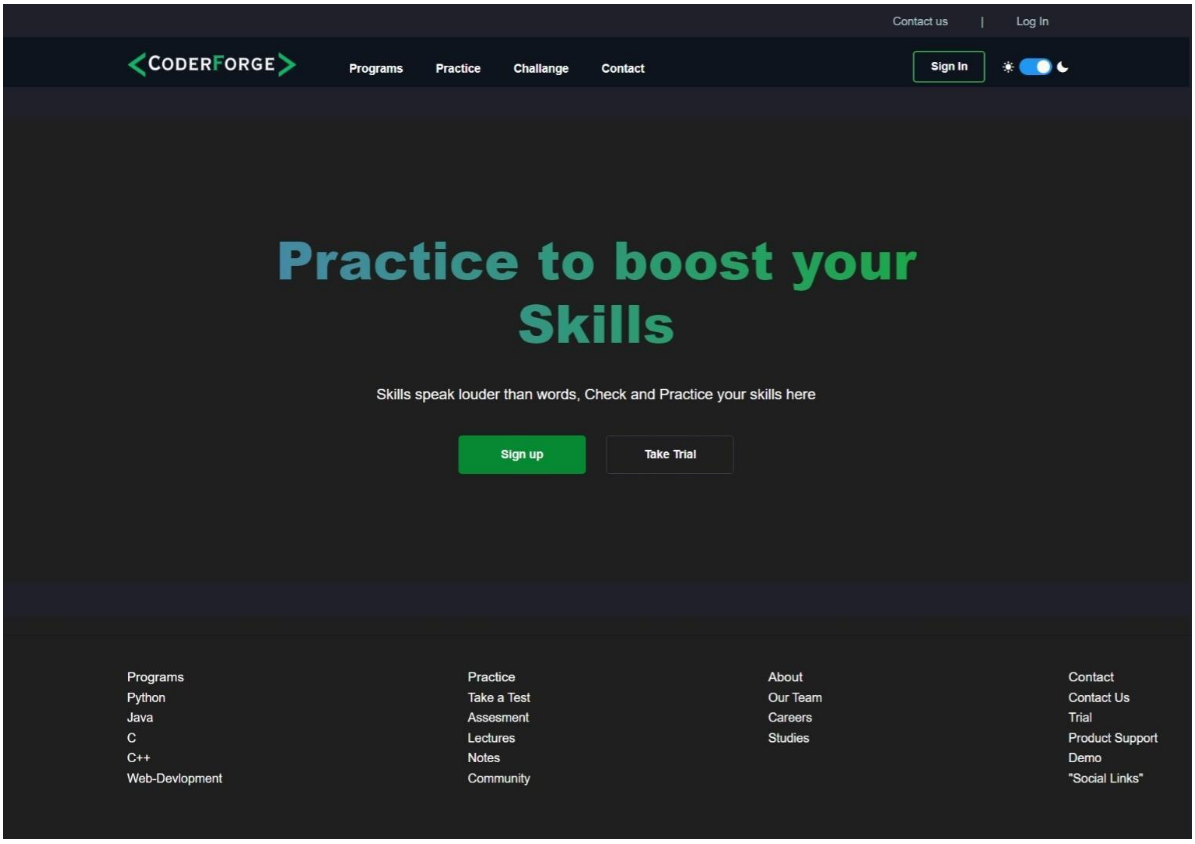


Practice to boost your Skills

Skills speak louder than words, Check and Practice your skills here

- **Homepage**: Screenshots showcasing the layout, design, and content of the homepage in dark mode:

Fig 4.8.2 Homepage:

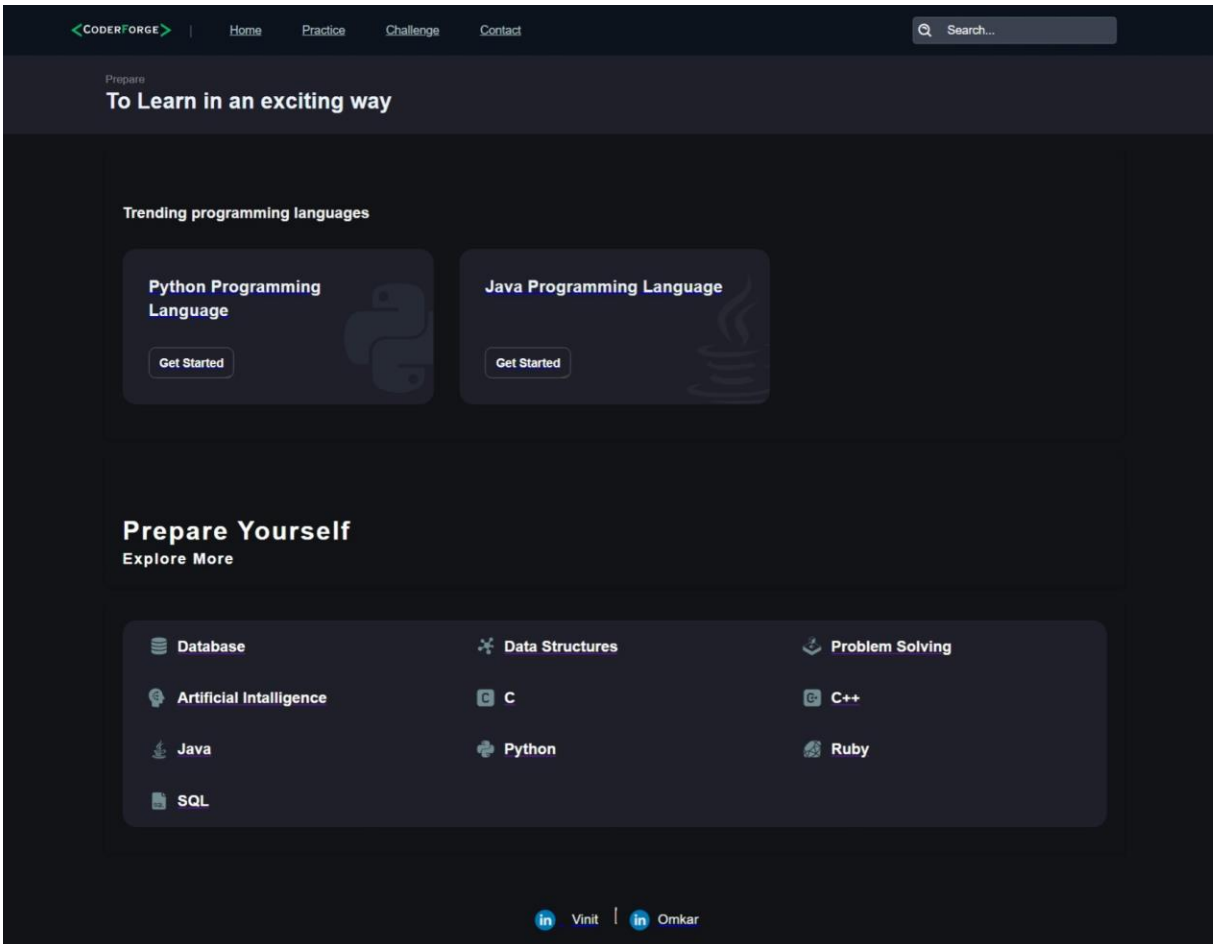


Practice to boost your Skills

Skills speak louder than words, Check and Practice your skills here

- **Challenge Section**: In this section you can choose any language to test your skills:

Fig 4.8.3 Challenge section



Prepare

To Learn in an exciting way

Trending programming languages

Python Programming Language

Java Programming Language

Prepare Yourself

Explore More

Database

Data Structures

Problem Solving

Artificial Intalligence

C

C++

Java

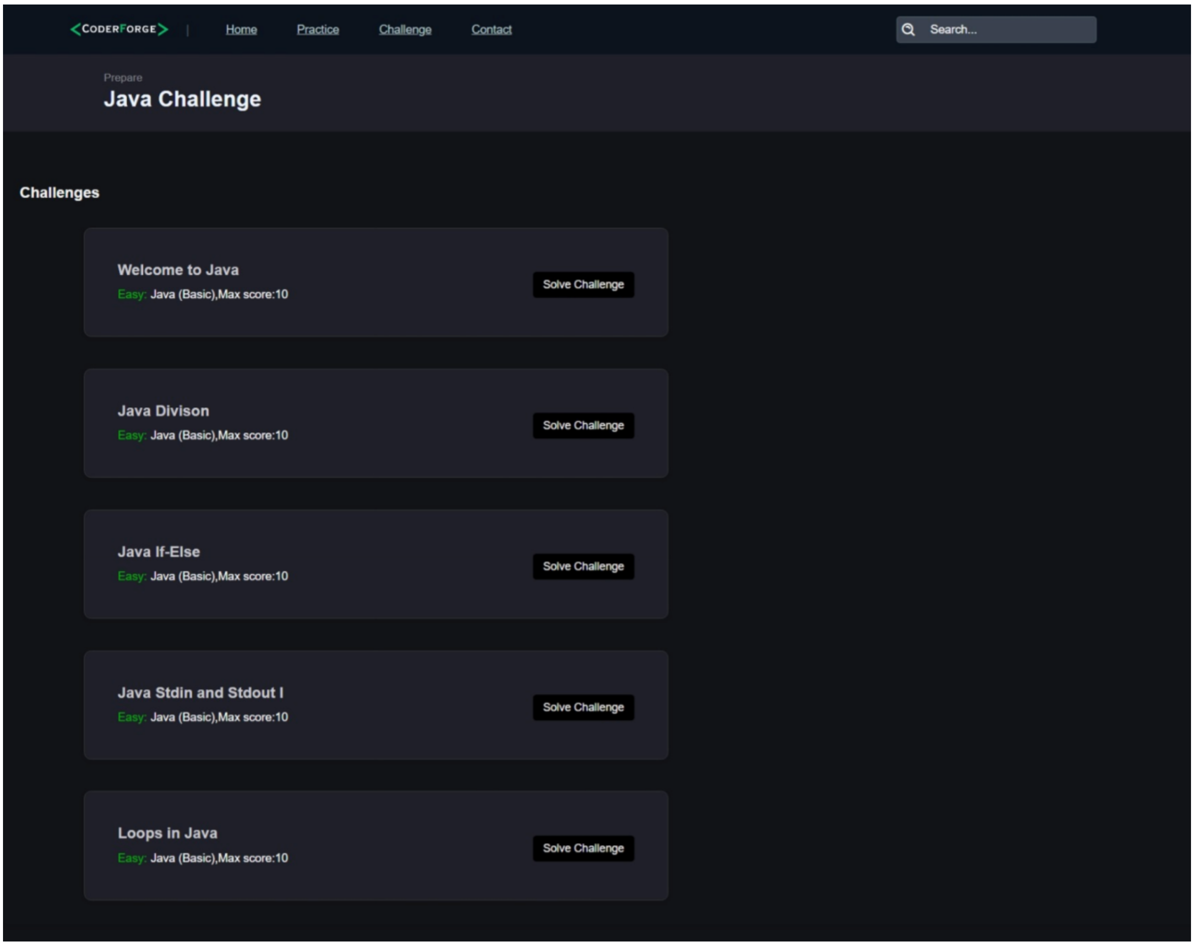
Python

Ruby

SQL

- **Java Challenge** : Java Challenge section:

Fig 4.8.4 Java Challenge section



Prepare

Java Challenge

Challenges

Welcome to Java
Easy: Java (Basic),Max score:10
Solve Challenge

Java Divison
Easy: Java (Basic),Max score:10
Solve Challenge

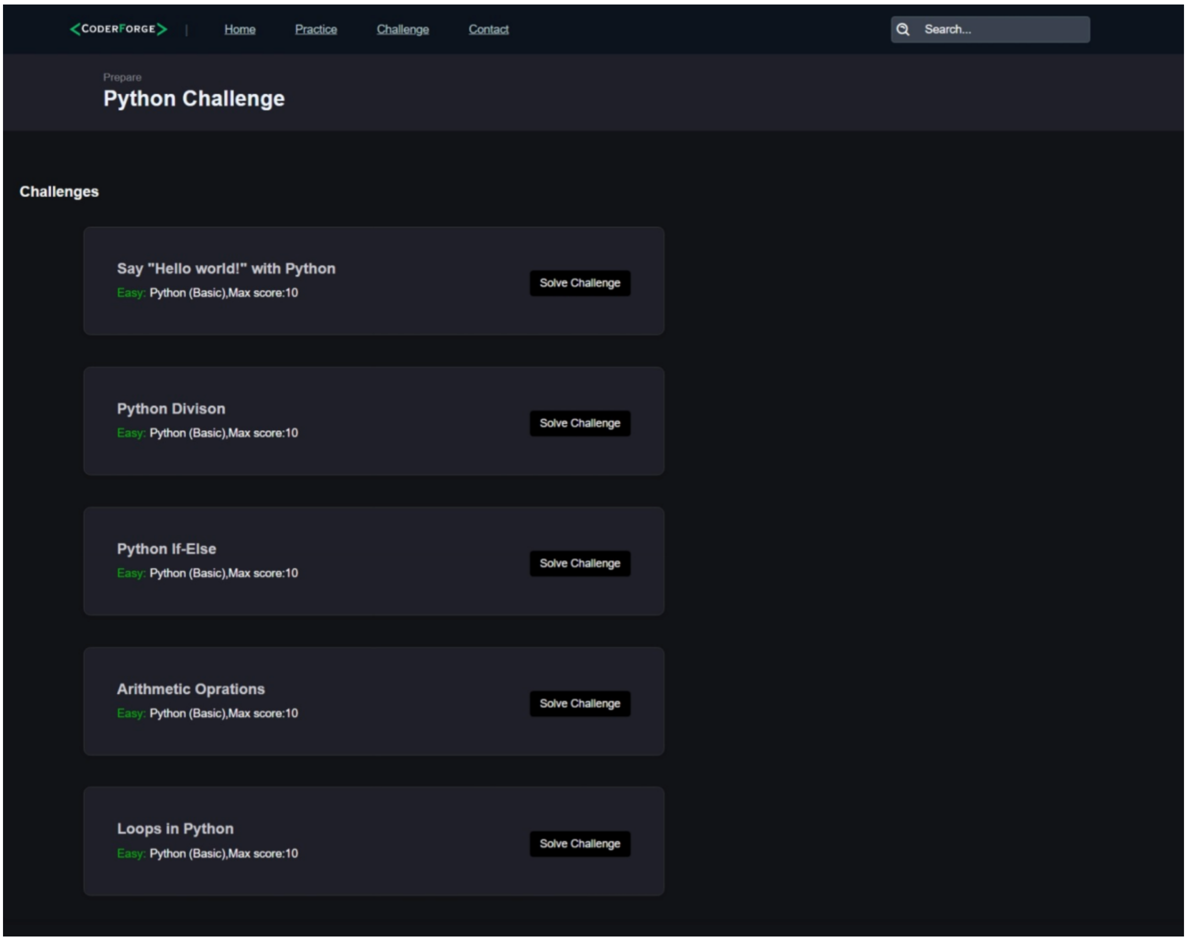
Java If-Else
Easy: Java (Basic),Max score:10
Solve Challenge

Java Stdin and Stdout I
Easy: Java (Basic),Max score:10
Solve Challenge

Loops in Java
Easy: Java (Basic),Max score:10
Solve Challenge

- **Python Challenge:** Python Challenge section:

Fig 4.8.5 Python Challenge section



Prepare

Python Challenge

Challenges

Say "Hello world!" with Python
Easy: Python (Basic),Max score:10

Python Divison
Easy: Python (Basic),Max score:10

Python If-Else
Easy: Python (Basic),Max score:10

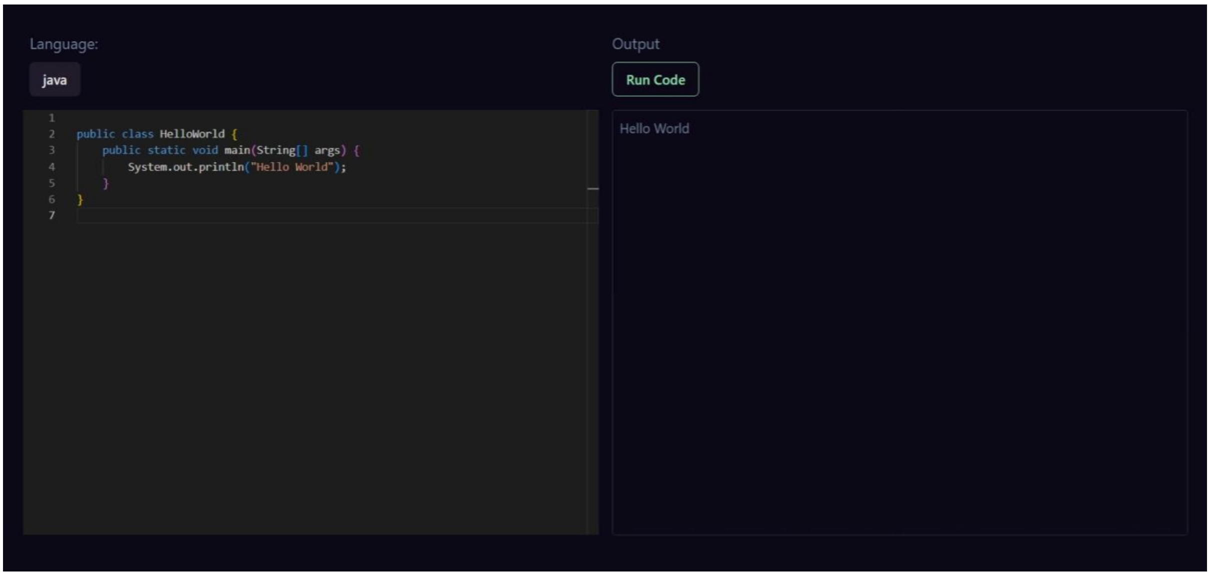
Arithmetic Oprations
Easy: Python (Basic),Max score:10

Loops in Python
Easy: Python (Basic),Max score:10

4.9 Output/ Screenshot

- **Output** : When Output is correct:

Fig 4.9.1 Correct output

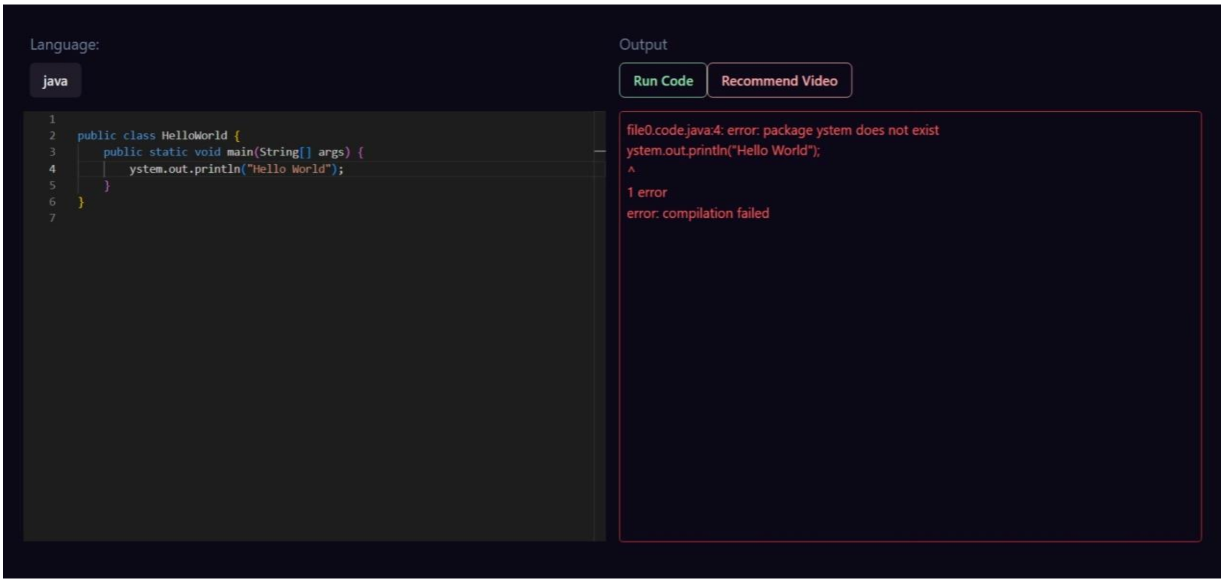


```java

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}

```

```
Hello World
```

- **Output** : When Output is incorrect it will recommend video for that error:

Fig 4.9.2 Incorrect output



```java

public class HelloWorld {
    public static void main(String[] args) {
        ystem.out.println("Hello World");
    }
}

```

```
file0.code.java:4: error: package ystem does not exist
ystem.out.println("Hello World");
^
1 error
error: compilation failed
```

Chapter 5: Conclusion

5.1 Conclusion

The development of the **Coder Forge** platform has been a challenging yet rewarding endeavor. The platform successfully provides a comprehensive and user-friendly environment for individuals to learn, practice, and master different programming languages. By offering a diverse array of programming language tasks and assessments, immediate and constructive feedback, personalized learning paths, comprehensive video lectures, and a supportive community forum, **Coder Forge** has managed to address many of the limitations observed in existing educational platforms.

5.2 Limitations

Despite its many features, **Coder Forge** is not without its limitations. For instance, while the platform does offer a wide range of programming tasks, it may not cover all possible programming concepts or languages. Additionally, while the platform aims to provide personalized learning paths, the effectiveness of these paths may vary depending on the individual's learning style and pace. Furthermore, while the platform does provide immediate feedback on tasks, this feedback is based on predefined criteria and may not capture all possible errors or areas for improvement.

5.3 Future Enhancements

Looking ahead, there are several enhancements that could further improve the **Coder Forge** platform. For instance, the platform could benefit from the addition of more advanced programming tasks and assessments, as well as support for more programming languages. The feedback mechanism could also be improved to provide more detailed and personalized feedback. Additionally, the platform could incorporate more advanced machine learning algorithms to better personalize the learning paths and recommendations. Finally, the platform could also benefit from the addition of more social features, such as the ability for users to collaborate on tasks or share their progress with others. These enhancements would not only improve the user experience but also further the platform's goal of democratizing access to programming education.

Chapter 6: Bibliography

1. "Development of a new model on utilizing online learning platforms to improve students' academic achievements and satisfaction"[1]: This research explores potential factors influencing students' academic achievements and satisfaction with using online learning platforms. The study was conducted on 243 students using online learning platforms in higher education.
2. "Design of an Online Programming Platform and a Study on Learners' Testing Ability"[2]: This study discusses the design of an online judge system called Pytutor that integrates software-testing concepts to assess learners' programming abilities and proficiency by exploring students' test cases and learning behaviors.
3. "Systematic research of e-learning platforms for solving challenges"[4]: This holistic research begins with a comprehensive comparative study about the available e-learning platforms, followed by a primary data analysis through an online survey of 364 engineering students from various colleges and branches

4. Connect to a MongoDB Database Using Node.js | MongoDB: This guide walks you through the basics of how to get started using MongoDB with Node.js. It covers connecting to a MongoDB database from a Node.js script, retrieving a list of databases, and printing the results to your console.