IntelliJudge Problem Solving Platform (An Open Source Problem Solving Platform)

By

Md. Shahidul Bhuiyan	ID:20211203004
Ridoy Chandra Sarker	ID:20211203005
Tanvir Hossen	ID:20211203008
Sakhawat Hossain	ID:20211203018
Kazi Atik Foysal	ID:20211203054

Submitted in partial fulfillment of the requirements of CSE 498

Bachelor of Science in

Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Business and Technology

Contents

List of Figures

Declaration

We do hereby declare that the project works presented here with entitled as, "Intelli Judge : Development of Problem Solving Platform " are the results of our own works. We further declare that the project has been compiled and written by us and no part of this project has been submitted elsewhere for the requirements of any degree, award or diploma or any other purposes except for this project. The materials that are obtained from other sources are duly acknowledged in this project.

Signature of Authors

Md. Shahidul Bhuiyan
ID: 20211203004

Ridoy Chandra Sarker
ID: 20211203005

Tanvir Hossen
ID: 20211203008

Sakhawat Hossain
ID: 20211203018

Kazi Atik Foysal
ID: 20211203054

Approval

This certificate acknowledges that Md. Shahidul Bhuiyan (ID: 20211203004), Ridoy Chandra Sarker (ID: 20211203005), Tanvir Hossen (ID: 20211203008), Sakhawat Hossain (ID: 20211203018), and Kazi Atik Foysal (ID: 20211203054) are students of Bangladesh University of Business and Technology (BUBT), we have successfully completed a project titled ”Intelli Judge : Development of Problem Solving Platform”. The project involved creating a super-advanced learning platform. It's designed very carefully to help job seekers learn really fast and smart. We know that learning is super important for engineers because the job market is very competitive. So, our project provides a solution that makes learning easier and helps people succeed.

Mr. Md. Ashiqur Rahman

Assistant Professor

Department of Computer Science & Engineering

Bangladesh University of Business and Technology

Chairman

Md. Saifur Rahman

Assistant Professor & Chairman(Acting)

Department of Computer Science & Engineering

Bangladesh University of Business and Technology

Dedication

We would like to dedicate this project to our loving parents. Words cannot express how grateful we are for your love, support, and encouragement throughout our academic journey. Your unwavering faith in us has been our source of strength, and we dedicate this project work to you as a token of our love and appreciation. Thank you for always being there, for your unwavering love and support, and for instilling in us the values of perseverance, dedication, and hard work. We hope to make you proud and repay you for all that you have done for us.

Acknowledgement

We begin by expressing our heartfelt gratitude to Almighty Allah for His countless blessings and boundless mercy upon us. Without His divine guidance, we would not have been able to achieve what we have today. We extend our sincere appreciation to Bangladesh University of Business and Technology (BUBT) for providing us with an exceptional environment to pursue our project. Our special thanks go to Md. Ashiqur Rahman, Assistant Professor in the Department of Computer Science and Engineering (CSE) at BUBT. His invaluable guidance and mentorship have been instrumental in shaping our project journey. He generously provided us with a wealth of academic resources in this field. His patience, dedication, and unwavering support have been indispensable. Even in the busiest of times, he always made time to engage in discussions and address our queries. We also wish to acknowledge our fellow team members, whose unwavering support and cooperation enriched our collaborative efforts. Lastly, we extend our gratitude to our university for affording us this remarkable opportunity. It is through their support that we were able to undertake this project.

Abstract

In an ever-evolving digital age, education is undergoing a profound transformation. This project explores the integration of digital learning platforms to revolutionize the educational landscape. The aim is to improve access to quality education for learners of all ages and backgrounds. One of our primary objectives is to create a dynamic online learning platform that hosts a wealth of educational content. This platform will include interactive courses on a wide array of subjects, from technology. Learners will have the opportunity to explore these subjects at their own pace, promoting self-directed learning. Furthermore, our project emphasizes the importance of practical actions. We provide participants with actionable steps and resources to reduce their carbon footprint, conserve natural resources, and advocate for sustainable practices within their communities. Ultimately, our project seeks to bridge the digital divide in education, making knowledge more accessible and engaging for learners worldwide. By leveraging the potential of digital learning platforms, we aspire to create a brighter future where education knows no boundaries and empowers learners to thrive in an increasingly complex and interconnected world. We hope this project will show that learning can be a great adventure, and anyone can be a part of it. Let's learn together and have some fun along the way!

Chapter 1

1 Introduction

1.1 Introduction

In a rapidly evolving digital age, access to quality education has become more crucial than ever. The traditional boundaries of classrooms are disappearing, and the world of learning is expanding beyond imagination. As we step into an era characterized by rapid change and ever-expanding opportunities, the importance of education becomes more pronounced than ever before. Nowadays, almost 95% of developers can use online coding test platforms. Many companies use these platforms in their hiring process to assess candidates' problem-solving skills and coding proficiency. Skill Assessment Educational institutions and coding bootcamps use these platforms for skill assessments, providing insights into students' coding abilities. Developers use these platforms for self-paced learning and practice, solving diverse coding challenges. Online judge platforms foster communities where developers discuss problems, share solutions, and collaborate.

This platform is not just a problem solving platform like HackerRank, LeetCode, Codeforces etc. it's a learning experience that encourages active participation, critical thinking, and skill development with a guidance and ML generated path for growing in this field.

1.2 Problem Statement

Nowadays, as the number of problem sovlers continues to increase, new users often struggle to find the right path, facing challenges in identifying which problems to solve next. This issue results in spending a considerable amount of time navigating through similar problems. Our recommendation is an online judge platform that uses a recommendation-based coding test system. This system not only suggests suitable problems for users but also offers quiz tests. Additionally, users have the option to create their own problems.

This platform brings about a significant change by streamlining the user experience. The recommendation system guides users towards problems that align with their skill levels and learning objectives. This not only saves time but also ensures that users are consistently challenged at an appropriate difficulty level.

The inclusion of quiz tests provides a structured way for users to assess their knowledge and understanding of various concepts. This feature adds a dynamic element to the learning process, allowing users to gauge their progress and identify areas that require further attention.

Moreover, empowering users to create their own problems is a unique and valuable feature. When users decide to create a problem, they are not only challenged to solve it themselves but also prompted to think critically about the problem's structure, difficulty, and potential solutions. This dual role of problem creator and solver contributes to a deeper understanding of programming concepts.

This recommendation-based online judge coding test platform addresses the challenges faced by new users. It offers tailored problem recommendations, quiz tests for self-assessment, and a creative outlet for users to contribute by crafting their own challenges. This holistic approach fosters a more engaging and effective learning experience.

1.3 Research Background

Research on online judge coding test platforms covers diverse areas, including the effectiveness of adaptive learning algorithms, impact on learning outcomes and skill development, user experience and engagement, gamification and motivation, community dynamics, assessment methods, remote assessment, inclusivity, user-created content, integration with educational curricula, and ethical considerations. This research aims to provide insights for developers, educators, and policymakers to enhance the effectiveness and fairness of these platforms.

1.4 Research Objectives

The following analysis objectives achieved from this analysis area unit are given below:

This platform offers problem-solving challenges across various subjects and difficulty levels. These challenges promote critical thinking, problem-solving skills, and real-world application of knowledge.

ML-driven recommendation system for next problem, so the problem solver can have a guidance for growing in a great way.

1.5 Motivations

The motivation behind Intelli Judge : Development of Problem Solving Platform With ML-Driven Recommendation System is to create an innovative online learning platform. We aim to address the limitations of traditional quiz systems, enhance user engagement through gamification, and bridge the gap between theory and practice with real-world problem-solving opportunities. Additionally, we strive to empower instructors by offering ML-powered tools for seamless course content management. Our goal is to provide an enriching and user-centric learning experience, ultimately democratizing education and making it more accessible and effective for all.

1.6 Significance of the Research

Research on online judge coding test platforms is important for several reasons. It helps to improve the evaluation of coding and problem-solving skills, which is crucial for hiring talented individuals globally. The findings also assist in streamlining recruitment processes by ensuring fairness through standardized skill assessments. Additionally, research insights contribute to enhancing learning environments for coding education, benefiting both educators and learners. The impact of online judge platforms on building coding communities is explored, emphasizing their positive influence on learning and collaboration. Understanding how these platforms align with industry needs informs strategies for talent development. The significance of online judge platforms in promoting accessibility and inclusivity in coding education is particularly important for supporting diverse learners. Research also guides data-driven decision-making for platform enhancements, ensuring continuous improvement. Insights into innovative assessment techniques contribute to the ongoing evolution of coding assessments. The implications of the research for educational policies further highlight the broader significance of online judge coding test platforms in preparing the workforce for the demands of the digital job market. Here is the section with both the main items and sub-items properly formatted with bullet points:

- **Problem Solving and Code Assessment:**
  - Users can solve programming problems in an integrated code editor.
  - An internal compiler automatically compiles and executes the submitted code.
  - Real-time feedback is provided on the code's correctness and performance.
  - Error messages and performance metrics guide users to improve their solutions.
- **Problem Upload:**
  - Users can contribute by uploading their own programming problems and test cases.
  - The system automatically compiles and validates the uploaded problems and test cases.
  - This feature encourages users to create and share innovative problems for others to solve.
- **Contest Management:**
  - Tools for managing programming contests, including setting up problems and participants.
  - Contest organizers can easily schedule and host coding contests.

  - Real-time leaderboards display participant progress and performance.

- **User Dashboard:**
  - Personalized dashboards show user activity such as solved problems, ongoing contests, and rank.
  - Performance metrics and progress tracking encourage skill improvement.
  - Users can track contest participation, problem-solving statistics, and overall rankings.

- **User Management:**
  - Administrators can manage user roles (e.g., regular users, problem setters, contest organizers).
  - User permissions are securely managed to maintain platform integrity.
  - Customizable user profiles allow users to view their progress and achievements.

1.7 Summary

This chapter comprises a broad overview of the problem, such as what we are we specifically targeting, what are the purposes of our thesis work are, and the motivation of the output of the thesis work.An online judge coding platform is a digital environment designed for efficient evaluation of coding and problem-solving skills. It provides a standardized and scalable method for assessing individuals' programming proficiency. These platforms support global talent acquisition, enabling organizations to evaluate candidates remotely. They contribute to continuous learning by offering diverse coding challenges, fostering community engagement through forums and contests. The platforms align with industry needs, preparing individuals for real-world scenarios. Educational institutions leverage them for skill assessments, and the platforms often integrate innovative features, ensuring relevance in the dynamic field of coding education and recruitment.

2 Literature Review or Background Study

2.1 Introduction

Problem solving is the most important practice of programmers over the world for learning programming. But over the years, beginners faced difficulties for practicing them because of lacking of guidance. Using ML we think we can solve this problem by recommending programmers next problem based on their preferences and previous solved problems and experience.

2.2 Related work

We have read many research papers and written down of main procedure of their articles.

Y. Watanobe et al. [1] proposed a system for evaluating program code called Online Judge System (OJS). OJSs are playing a key role in both academia and industry to evaluate the correctness of programs submitted by the learners. OJS accepts solution codes from arbitrary users, compiles and executes them in a shared environment, validates the behavior of them using specific test input/output datasets, and then reports the results of the evalu-ation and resource usage to all associated users

S. Wasik et al. [2] proposed Online judges are systems designed for the reliable evaluation of algorithm source code submitted by users, which is then compiled and tested in a homogeneous environment.Use heuristic algorithm is a technique for solving problems when traditional algorithms fail.

S. Wenfei Chen et al.[3] implemented Three layer recommendation model and LDA model in their online judge platform for research. LDA is a generative probability model, which can generate a latent layer of interest which will guide the user's choice according to the behavior of others in the past. Finally, their recommendation method for OJ based on the three-layer collaborative filtering algorithm helps the users locate the suitable problems according to their levels and may be a potential substitute for the traditional way of suggestions[3]

Md Mahim Anjum et al.[4] cutting-edge language models, trained on coding languages, as our reference point and juxtapose them using match-based (essentially token similarity) and execution-based (focusing on functional assessment) criteria. Used Transformer-based language models as foundational methods. Accuracy 90.1 which depends on difficulty levels.

Xudong Lu et al.[5] demonstrate Data Driven Analysis on the Effect of Online Judge System. Online judge system has been proved to be an effective means to improve students' programming abilities. Use a data-driven analysis method called Dimension Extraction algorithm (DECA) to answer the question of what are the concepts in introductory programming that are easy/hard for students.

Raciel Yera Toledo et al.[6] identifies a significant problem on online judges: users often struggle to find appropriate problems to solve, leading to frustration and decreased engagement. The author adopts a basic collaborative filtering approach that initially considers just problems solved or not solved. They also propose several ways to incorporate specific judge's information in the recommendation process, like problems that the students try to solve but they fail, and problems where students usually need several failed intents before accepting them.

Adrian Kosowski et al.[7] presentes that online judge and contestant system, used for E-Learning of programming. where the students learn the C/C++ programming languages and study data structures (heaps, binary search trees and graphs) and dozens of algorithms (sorting, set operations, hashing, graph algorithms, pattern recognition, geometry algorithms, etc...). . The students have to choose the appropriate data structure and design an efficient algorithm with a bounded computational complexity to solve each of about 50 easy and hard problems with a strict specification and i/o examples.

W. X. Zhao et al.[8] prepared a dataset from top three Online Judge platforms Timus(Russia), POJ(Peking University Online Judge, China) and HDU(Hangzhou Dianzi University, China). There are two different learning patterns to consider, either volume- or topic-oriented modes. After describing the two kinds of learning patterns separately, they integrated them in a unified model The Two-Mode Markov Topic Model. For POJ, Timus and HDU datasets the accuracy was respectively 81.7, 70.7 and 78.8

Andy Kurnia et al. [9] describe and evaluate the implementation and applicability of an automatic programming assignment grading system we named the online judge. They have implemented that automatic grading that would overcome some of the deficiencies of manual grading. However, this requires careful implementation of the system, there are still many security holes that will need to be patched before we actually start using it. Based on their testing, we conclude that automatic grading is indeed applicable to modules that emphasize programming.

Gang Huang et al. [10]studied an Online Judge platform, identifying 3,700 bugs in false positive solutions. It evaluates test suite quality and underscores the need for ongoing efforts to enhance effectiveness. The author has crucial implications and challenges across three key domains. For OJ platform developers, it emphasizes the limitations of testing in ensuring correctness, advocating for cautious reliance on assessment results and the adoption of routine test suite updates with random testing for improved quality..

Zhang et al.[11] developed a matrix factorization method for explainable recommendation, which provides an explanation sentence for the recommended item. As a result, it falls into the category of matrix factorization with textual explanation. Also they introduced the Item-based CF method, and further described its application in the Amazon product recommendation system. Item-based CF takes each item as a vector of ratings, and predicts the similar items based on the previous purchase and search data. The research shows some simple statistical information as explanations, such as 70 percent of recommended products were bought.

Yongheng Mu et al. [12] proposes a multimodal movie recommendation system that leverages deep learning to address the information overload problem and provide users with personalized content and services. The research addresses information overload, focusing on movie recommendation systems. Traditional algorithms face challenges, prompting the introduction of a personalized multimodal system. Combining deep learning and multimodal data analysis, the system results in an increase in accuracy, particularly with MovieLens datasets.

Wu Da-Qin,Yan Hui et al.[13] proposes a novel CDIO-based online judge system, aiming to address deficiencies in personalized learning pathways. Leveraging adaptive challenges, collaborative tools, and innovative evaluation methods, it enhances learning efficiency and community engagement in programming education

Md. Mostafizer Rahman,Yutaka Watanobe and Keita Nakamura et al.[14] propose an intelligent support model for program code completion in software engineering and programming education. Using a combination of LSTM and an attention mechanism, the model achieves high accuracy in error detection (62) and source code classification (96). It aims to enhance source code debugging and refactoring processes, particularly for lengthy code sequences, and is positioned as an efficient solution with promising applications in the field.

Yutaka Watanobe et al.[15] proposes a next-generation programming learning platform architecture using an online judge system, aiming to generate learning objects from user and judge activities. It emphasizes the lifecycle of data and suggests Apache Hadoop and Fluentd for data proliferation. Despite not detailing specific experiments, the platform envisions providing tasks, solution codes, and evaluations while acknowledging challenges in its implementation. The review outcome is not explicitly discussed, and the paper concludes by highlighting the model's potential application in programming education with suggestions for clarity improvement.

References	Research Purpose	Algorithm	Research Gap
1	Evaluating system for programming problems, called Online Judge System	N/A	Codes from arbitrary users, compiling and executing them in a shared environment and also validating the behavior
2	Systems designed for reliable evaluation algorithm source code submitted by users, which is then compiled and tested in Homogeneous environment	heuristic algorithm	Include scalability of online judges, security in code evaluation, diversity of test cases, adaptability to programming paradigms, and human-centric aspects, while potential research gaps may involve heuristic algorithm design and generalizability of findings
3	Three-layer recommendation model, incorporating LDA for generating a latent layer of interest, and proposing a three-layer collaborative filtering algorithm for personalized problem suggestions based on user skill levels	content-based filtering	Personalized recommendations, dynamic user behavior, data privacy, evaluation metrics, adaptability to diverse problem domains, and user acceptance in integrating a Three-layer recommendation model and LDA in their online judge platform.
4	Aimed to assess cutting-edge language models trained on coding languages, using match-based and execution-based criteria.	N/A	Robustness to ambiguity in evaluating cutting-edge language models for coding tasks, with a focus on real-world applicability and interpretability.

References	Research Purpose	Algorithm	Challenges or Research Gap
5	Determine the effectiveness of the Online Judge System in enhancing students' programming abilities and identifies easy/hard introductory programming concepts	Dimension Extraction	Generalizability student engagement, long-term impact, bias, adaptability to varied concepts, and the integration of online judge systems within educational pedagogy in the context of data-driven analysis on the effects of online judge systems on students' programming abilities.
6	Online judges by implementing a collaborative filtering approach, considering problems solved or not solved, and incorporating specific judge's information to enhance problem recommendation accuracy	N/A	The challenge of user difficulty in finding appropriate problems on online judges, aiming to bridge the gap by implementing a collaborative filtering approach and enhancing recommendation accuracy with specific judge's information.
7	E-Learning in programming by implementing an online judge and contestant system	N/A	Challenges related to assessing students' proficiency in algorithmic problem-solving and identifying potential research gaps in effective online learning methodologies for programming education
8	Develop a unified model, the Two-Mode Markov Topic Model	N/A	Effectively integrating learning patterns from diverse Online Judge platforms and exploring potential research gaps in unified models for analyzing volume or topic-oriented modes of learning.

References	Research Purpose	Algorithm	Challenges or Research Gap
9	Implementation of an automatic programming assignment grading system	N/A	In the secure implementation of an automatic programming assignment grading system, emphasizing the need to address security holes for broader adoption
10	Improve Online Judge platforms by identifying 3,700 bugs, evaluating test suite quality, and emphasizing the need for ongoing efforts in routine updates for enhanced effectiveness.	N/A	Challenges in ensuring the correctness of Online Judge platforms, emphasizing the need for caution in relying on assessment results and suggesting ongoing efforts to address limitations and enhance quality
11	Explainable recommendation system, utilizing matrix factorization with textual explanations and emphasizing Item-based Collaborative Filtering in Amazon product recommendations	N/A	Challenges in achieving effective and interpretable explanations in recommendation systems, emphasizing potential research gaps in enhancing interpretability and accuracy
12	A deep learning-based multimodal movie recommendation system to tackle information overload and enhance accuracy, with a focus on MovieLens datasets	N/A	Challenges in movie recommendation algorithms and explores gaps in achieving enhanced personalization and accuracy with deep learning and multimodal data.

References	Research Purpose	Algorithm	Challenges or Research Gap
13	Proposed a Programming Assignment Management System with a focus on the assignment submission and evaluation process	N/A	Addressing the complexity and scalability of programming assignment submissions in an educational setting, with potential research gaps in improving the system's adaptability and automation
14	Aiming to develop a framework for adaptive online programming education platforms	N/A	Exploring adaptive learning paths for students, focusing on individual needs and offering personalized problem recommendations with the possibility of further research into improving the adaptability of online education systems
15	Introduced an Automatic Grading System using machine learning techniques to evaluate code submissions	machine learning	Challenges in accurately grading diverse types of code submissions, specifically addressing the limitations in current machine learning techniques for automated code evaluation

Chapter 3

3 Proposed Model

3.1 Introduction

In an era characterized by rapid technological advancements, the landscape of education is evolving at an unprecedented pace. Educational Technology, or EdTech, has emerged as a transformative force in the realm of teaching and learning. The integration of technology into education promises to enhance access, improve engagement, and elevate the quality of educational experiences. However, it is imperative that this integration be guided by a well-considered framework to maximize its potential and mitigate potential challenges.

3.2 Feasibility analysis

This project took four months to complete and involved a team of five researchers along with one supervisor. The development and testing phases required both hardware and software resources, including computational power for code evaluation and system integration.

The research team also created and assessed the necessary datasets for testing the platform's functionality, ensuring that the collection and handling of data adhered to legal and ethical standards. However, no financial support was provided by either the supervisor or the institution, which added an extra layer of responsibility for the researchers in managing resources effectively.

3.3 Requirement Analysis

3.3.1 Backend

- Node.js
- Express.js

- MongoDB
- TensorFlow.js
- Machine Learning Model (trained for code suggestions/evaluation)

3.3.2 Frontend

- React.js
- Code Editor

3.3.3 Additional Tools

- Docker
- CI/CD tools (Jenkins)

3.3.4 Key Considerations

- Scalability
- Security
- User Experience
- Model Maintenance and Updates

3.4 The Research Methodology

This proposed framework represents our institution's commitment to leveraging the power of technology to enhance the educational journey for all stakeholders - students, educators, administrators, and parents. By embarking on this strategic journey, we seek to address the pressing educational needs of the 21st century, equipping our learners with the skills, knowledge, and competencies they need to thrive in an increasingly digital and interconnected world.

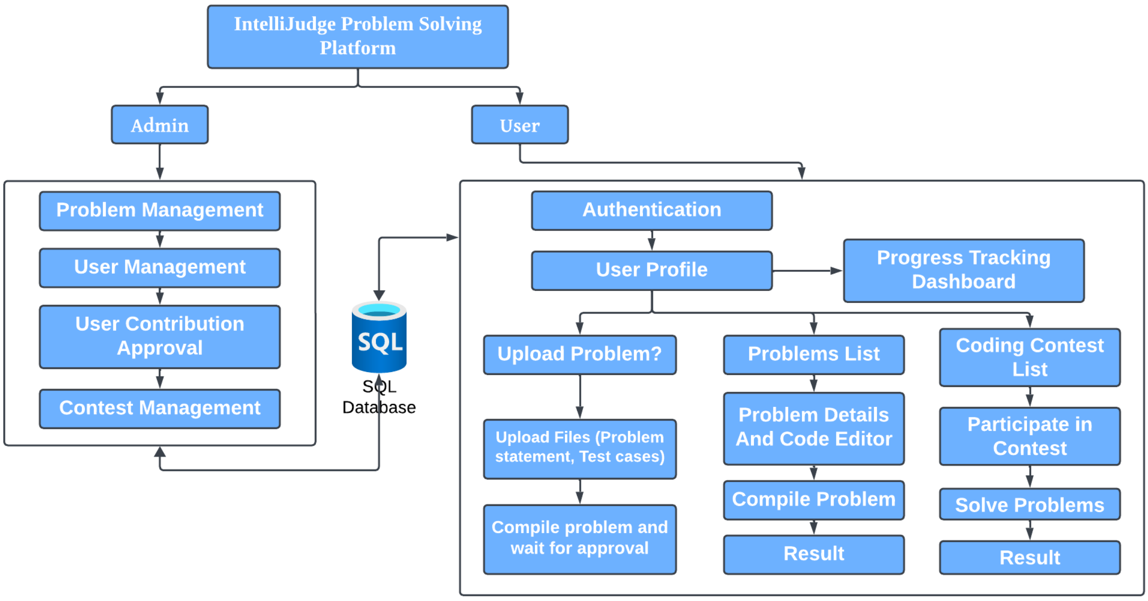


Figure 3.1: Proposed model .

- **Enhancing Learning Outcomes:** To improve student achievement and engagement by leveraging technology to create dynamic, personalized, and interactive learning experiences.

- **Equity and Access:** To bridge the digital divide by ensuring equitable access to technology and digital resources for all learners, regardless of their background or circumstances.

- **Data Privacy and Security:** To prioritize the protection of student data and privacy, ensuring that all technology implementations comply with relevant regulations and standards.

- **Sustainability:** To develop a sustainable and scalable ecosystem that can adapt to evolving technological trends and remain financially viable over the long term.

3.5 System Overview

Problem Solving is an ambitious online learning platform designed to offer a comprehensive and interactive educational experience. The system encompasses a multifaceted set of features to cater to both learners and instructors. Overall, ”Problem Solving With ML-Driven Recommendation System” aspires to redefine online education, providing a dynamic and inclusive learning environment that empowers users to acquire knowledge, solve problems, and connect with instructors and peers, all facilitated by cutting-edge ML technology. The following is a high-level description of the system architecture:

3.5.1 User Registration and Authentication:

- **User Registration:** Users can create accounts on the platform, providing basic information such as name, email, and password.
- **Authentication:** Secure authentication mechanisms ensure that only authorized users can access the platform.

3.5.2 User Profile Management:

- **User Registration:** Users can create accounts on the platform, providing basic information such as name, email, and password.
- **Authentication:** Secure authentication mechanisms ensure that only authorized users can access the platform.

3.5.3 Problem-Solving Activities:

Problem-solving exercises are a key component of our project. These exercises challenge users to apply their knowledge and skills to real-world problems, helping them develop critical thinking and analytical abilities. Users will work through a variety of exercises that cover topics such as algorithms, data structures, etc.

One example of a problem-solving exercise is the ’Tower of Hanoi’ puzzle. This puzzle involves moving a stack of disks from one peg to another, following specific rules. The exercise requires users to think strategically and logically, as they must plan each move carefully to avoid making mistakes. Other exercises include coding challenges and logic puzzles, which help users develop their programming and problem-solving skills.

- **Admin-Provided Problems:** Administrators curate a repository of problems and exercises across different subjects and difficulty levels.
- **User Participation:** Users can engage in problem-solving activities to enhance their critical thinking skills.
- **Quiz Creation:** Instructors are provided with ML-powered content creation tools to develop course materials efficiently.
- **Participation and Scoring:** Users can participate in quizzes, earn points for correct answers, and track their quiz scores.

3.5.4 Contest management system:

- **Contest Creation and Scheduling:**

  Contest organizers can create contests with a defined start and end time, set rules, and assign problem sets. Contests can be scheduled in advance, with the platform automatically opening and closing the contest at the designated times.

- **Problem Set Management:**

  Contest organizers can select from existing problem sets or upload new problems for the contest. Organizers can set problem difficulty levels and assign different point values based on complexity.

- **Real-time Monitoring and Submissions:**

  Participants submit their solutions through the platform, where submissions are compiled and executed in real time. Organizers and participants can monitor live submissions and see real-time results and standings.

- **Leaderboards and Scoring:**

  The platform generates real-time leaderboards that display participant rankings based on performance. Rankings are determined by factors such as number of problems solved, time taken, and accuracy of solutions.

- **Participant Management:**

  Organizers can manage participants by adding them manually or allowing open registration. Participants are grouped by contest type, such as individuals or teams, with options to assign specific roles within teams. The system supports user authentication to ensure that only authorized participants join the contest.

- **Post-contest Analysis:**

  After the contest, participants can review their submissions, test cases, and where their solutions failed. Organizers can generate detailed contest reports, including problem-solving statistics, participant performance, and any issues encountered during the contest. The platform stores historical contest data, allowing participants to access past contest results and improve for future competitions.

3.6 User Interface

Users can take part in competitions, submit answers, and work on a variety of coding problems. We provide a smooth experience with our user-friendly interface, which includes features like a powerful coding environment and instant feedback on solutions that are submitted. Our platform is made to accommodate a range of skill levels, so whether you're a beginner programmer or a seasoned developer, it encourages ongoing learning and skill improvement. The following activities are part of it:

3.6.1 User Request

Through the user interface, Users start by creating an account on the platform. Once registered, users can set up their profiles, including customizing avatars, adding personal information, and setting preferences. Users browse a list of problems categorized by difficulty. They choose a problem, view its details, and access the coding environment to submit a solution. In the coding environment, users write and test their code. Users can participate in scheduled competitions or create their own. They join competitions, solve problems within a specified time frame, and compete against others.

3.7 Conclusion

This proposed framework serves as a guiding document, providing a roadmap to navigate the evolving landscape of education in the digital age. Through strategic planning, collective effort, and a commitment to our educational mission, we aim to usher in a new era of educational excellence and inclusivity.

Chapter 4

4 Implementation and Testing

4.1 Introduction

The successful implementation and testing of the Online Judge System involve a structured approach to software development, ensuring that the system meets both functional and non-functional requirements. This chapter outlines the process of setting up the system, designing and integrating its modules, and conducting rigorous testing to validate its performance and reliability. The system is built using React.js for the front end, Express.js for the backend framework, Node.js as the server environment, and MongoDB as the database. These technologies together enable features for users to contribute programming problems, solve challenges, and participate in competitions. Each section delves into the specifics of how these components were developed, tested, and refined to achieve a robust and user-friendly platform.

4.2 System Setup

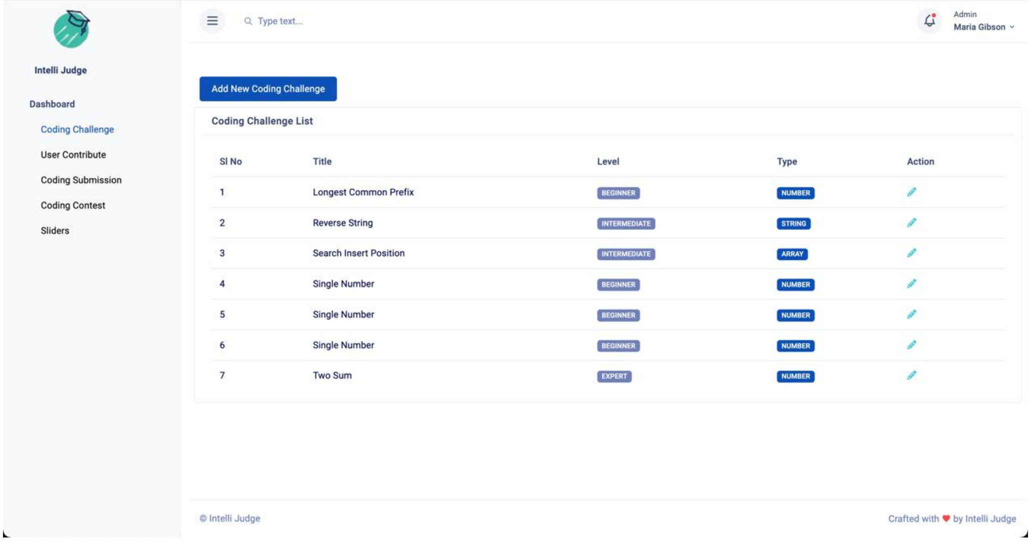
The system setup involves several key steps to configure the development environment and deploy the Online Judge System. The necessary software, including Node.js and MongoDB, was installed and configured. The development environment was set up to facilitate smooth coding, debugging, and testing processes. A local server environment was configured using Node.js to simulate the production environment, while MongoDB was set up to handle data storage and retrieval. Once the environment was prepared, the project files were organized, and database schemas were created to support the application's functionalities. Configuration files were adjusted to match the server settings, ensuring that all components interact seamlessly.

4.3 System Modules

The Online Judge System is composed of several integral modules, each designed to handle specific functionalities:

4.3.1 Admin Panel:

- **Coding Challenge Management:** This module allows administrators to create, edit, and manage coding challenges. It includes tools for monitoring challenge performance and user engagement, utilizing React.js for the interface and Express.js for backend operations [4.1].



Coding Challenge List

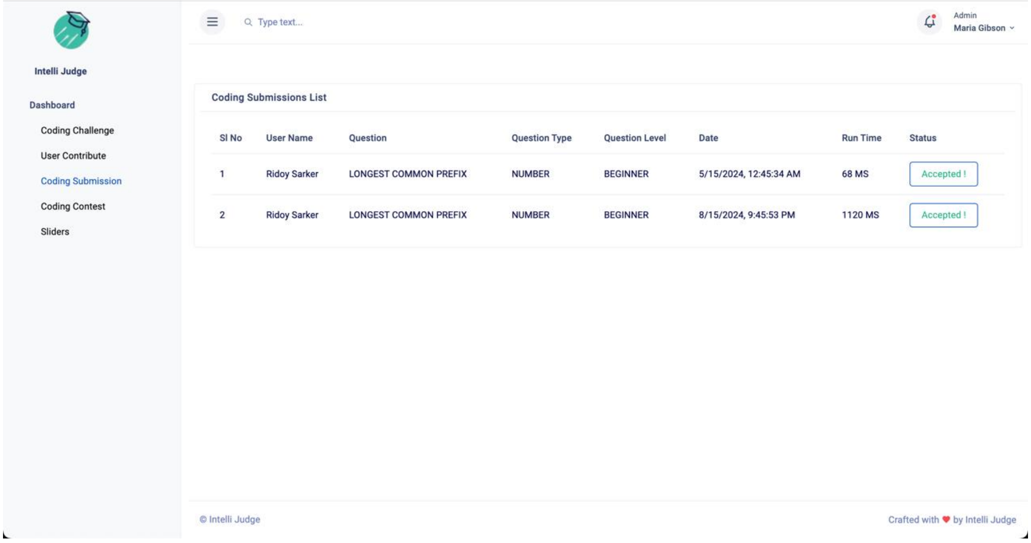
Sl No	Title	Level	Type	Action
1	Longest Common Prefix	BEGINNER	NUMBER	✎
2	Reverse String	INTERMEDIATE	STRING	✎
3	Search Insert Position	INTERMEDIATE	ARRAY	✎
4	Single Number	BEGINNER	NUMBER	✎
5	Single Number	BEGINNER	NUMBER	✎
6	Single Number	BEGINNER	NUMBER	✎
7	Two Sum	EXPERT	NUMBER	✎

Figure 4.1: Coding Challenge Management.

- **Coding Challenge Submissions:** Administrators can review and manage submissions for coding challenges. This module ensures that all submissions are evaluated against predefined criteria, with data stored in MySQL [4.2].

- **User Contribution Management:** This module enables administrators to oversee user contributions, including problem submissions and code contributions. It provides insights into user activity and contribution quality, leveraging React.js for the frontend and Express.js for backend management [4.3].

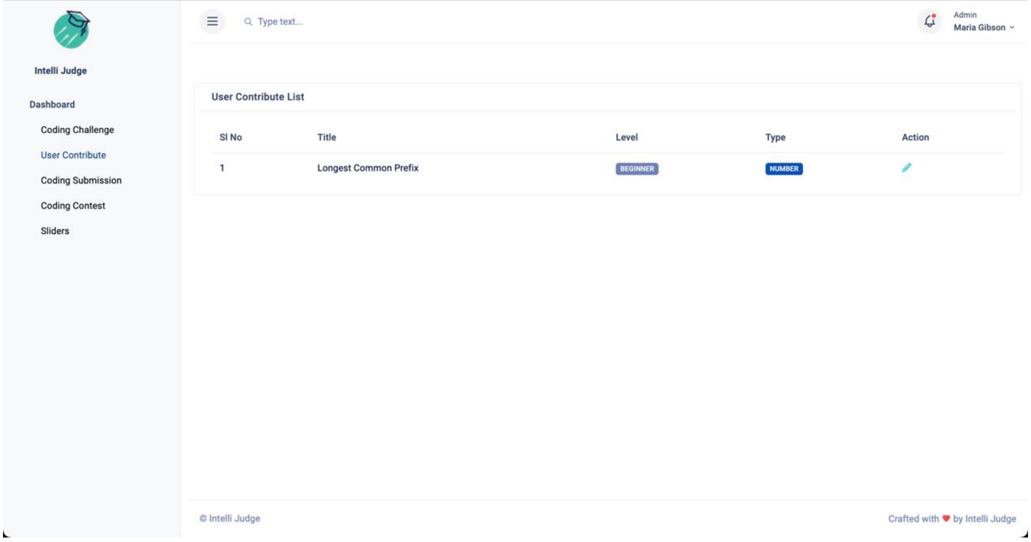
- **Coding Contest Management:** This module facilitates the organization and management of coding contests. It includes features for contest schedul-



Coding Submissions List

Sl No	User Name	Question	Question Type	Question Level	Date	Run Time	Status
1	Ridoy Sarker	LONGEST COMMON PREFIX	NUMBER	BEGINNER	5/15/2024, 12:45:34 AM	68 MS	Accepted !
2	Ridoy Sarker	LONGEST COMMON PREFIX	NUMBER	BEGINNER	8/15/2024, 9:45:53 PM	1120 MS	Accepted !

Figure 4.2: Coding Challenge Submissions.



User Contribute List

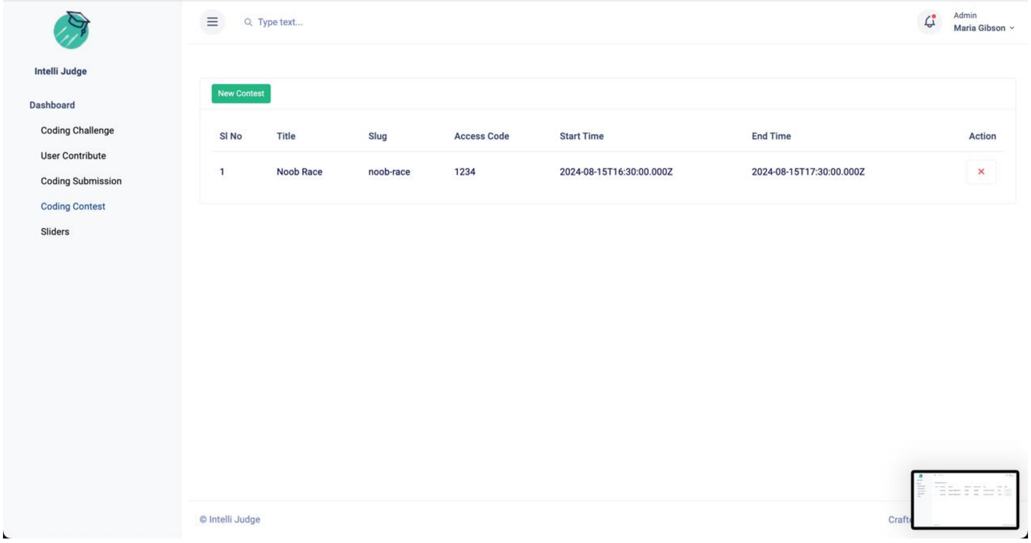
Sl No	Title	Level	Type	Action
1	Longest Common Prefix	BEGINNER	NUMBER	✎

Figure 4.3: User Contribution Management.

ing, participant registration, and result tracking, all managed via React.js and Express.js, with data stored in MongoDB [4.4].

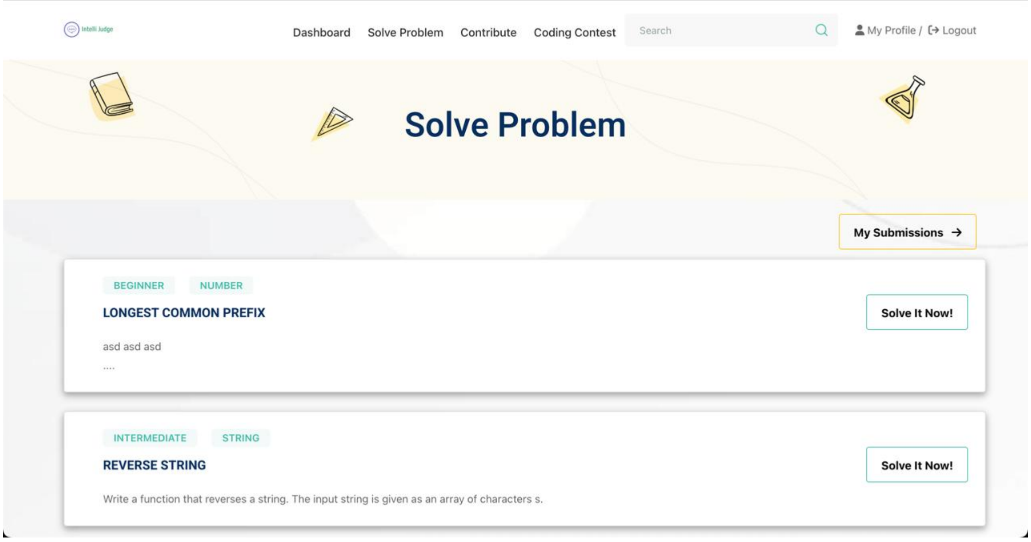
4.3.2 User Panel:

- **Coding Challenges:** Users can browse and participate in various coding challenges. This module provides an intuitive interface built with React.js, allowing users to view challenge details and submit their solutions [4.5].
- **Code Submissions:** Users can submit their code solutions for challenges. This module handles code submission, execution, and output verification against



Sl No	Title	Slug	Access Code	Start Time	End Time	Action
1	Noob Race	noob-race	1234	2024-08-15T16:30:00.000Z	2024-08-15T17:30:00.000Z	×

Figure 4.4: Coding Contest Management.



LONGEST COMMON PREFIX

asd asd asd

....

REVERSE STRING

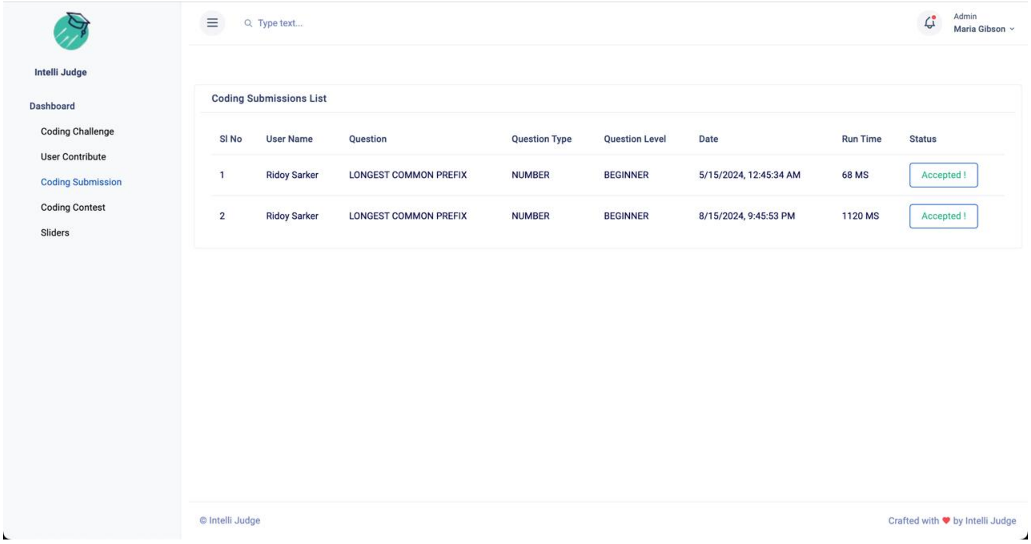
Write a function that reverses a string. The input string is given as an array of characters s.

Figure 4.5: Coding Challenges.

predefined test cases, with Node.js executing the backend processes [4.6].

- **Code Contributions:** Users can contribute new programming problems and associated test cases. This module includes interfaces for problem description, input/output specifications, and constraints, with MongoDB storing the problem data [4.7].

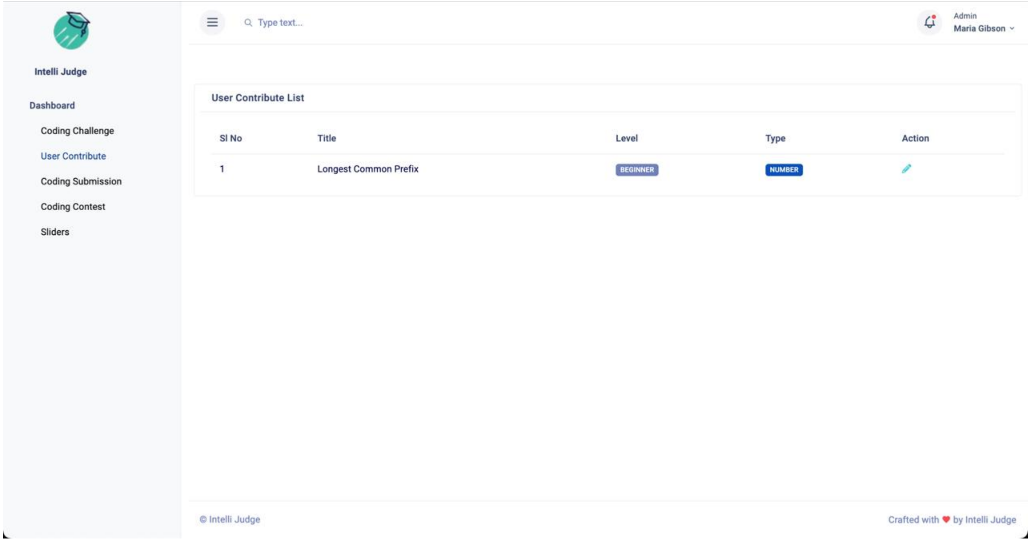
- **Dashboard:** Users have access to a personalized dashboard that displays their activity, including submitted solutions, challenges participated in, and contributions made. This dashboard is built using React.js for a seamless user expe-



Coding Submissions List

Sl No	User Name	Question	Question Type	Question Level	Date	Run Time	Status
1	Ridoy Sarker	LONGEST COMMON PREFIX	NUMBER	BEGINNER	5/15/2024, 12:45:34 AM	68 MS	Accepted !
2	Ridoy Sarker	LONGEST COMMON PREFIX	NUMBER	BEGINNER	8/15/2024, 9:45:53 PM	1120 MS	Accepted !

Figure 4.6: Code Submissions.



User Contribute List

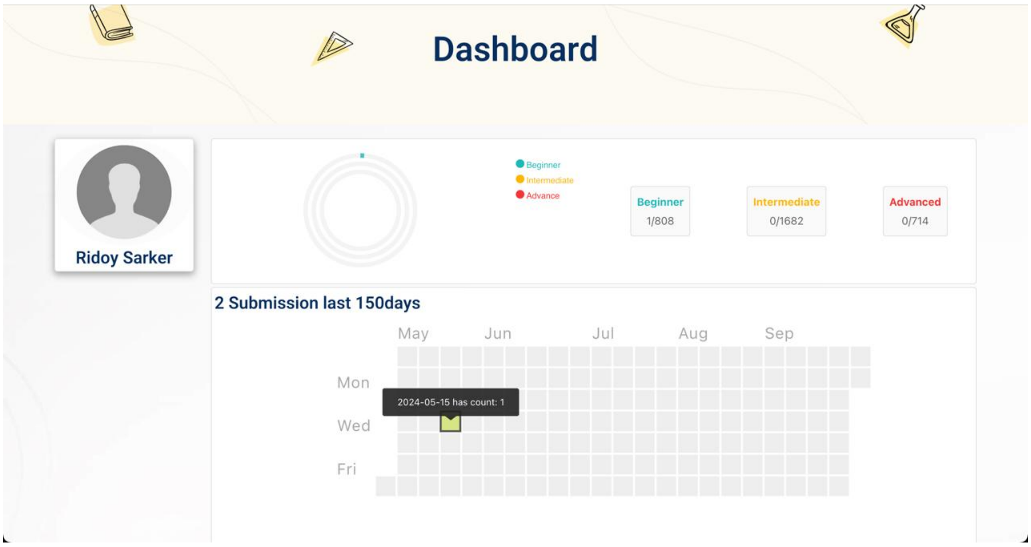
Sl No	Title	Level	Type	Action
1	Longest Common Prefix	BEGINNER	NUMBER	✎

Figure 4.7: Code Contributions.

rience [4.8].

- **Coding Contest:** Users can participate in coding contests organized by the platform. This module provides information on upcoming contests, registration options, and results tracking, all managed through a user-friendly interface [4.9].

Each module was developed separately and then integrated to ensure that all components function cohesively. The modular design allowed for iterative development and testing, improving the overall system quality.



Dashboard

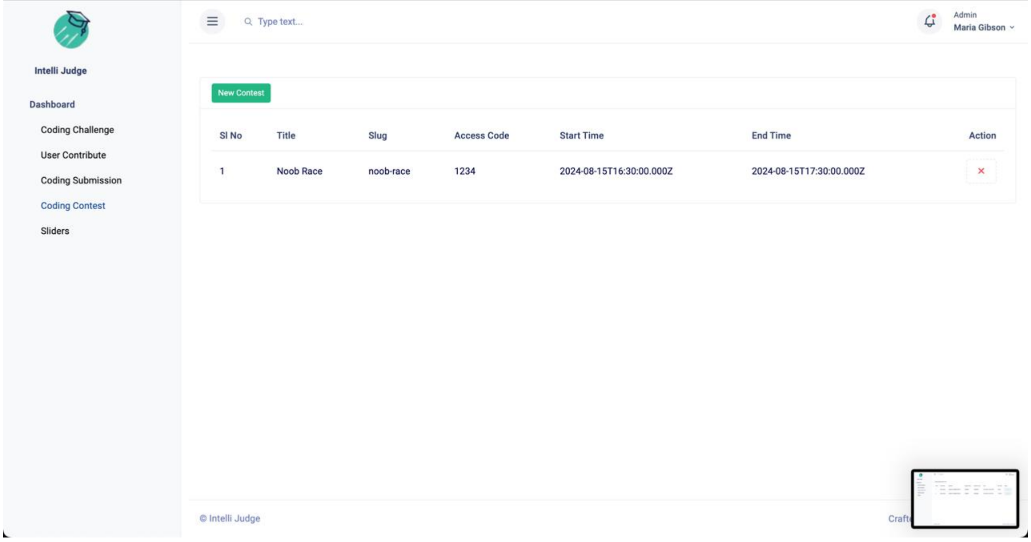
Ridoy Sarker

Beginner	Intermediate	Advanced
1/808	0/1682	0/714

2 Submission last 150days

2024-05-15 has count: 1

Figure 4.8: User Dashboard.



Sl No	Title	Slug	Access Code	Start Time	End Time	Action
1	Noob Race	noob-race	1234	2024-08-15T16:30:00.000Z	2024-08-15T17:30:00.000Z	×

Figure 4.9: Coding Contest.

4.4 Prototype Design

The prototype design of the Online Judge System was developed to visualize and refine the system's user interface and functionality before full-scale implementation. Wireframes and mockups were created using design tools and translated into React.js components representing key interfaces, including user dashboards, problem submission forms, and competition management pages. User feedback was gathered through usability testing to identify areas for improvement. The prototype was iteratively refined based on feedback to ensure an intuitive and user-friendly experience. This design phase was crucial in validating the system's concept and layout, paving the way for more detailed development and integration.

4.5 System Requirements for End-User

For the Online Judge System to function effectively, end-users need to meet certain system requirements:

- **Hardware Requirements:** A modern computer or laptop with at least 4GB of RAM and a stable internet connection.
- **Software Requirements:** A recent version of a web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge. Users should also have a text editor or IDE for coding purposes.
- **Account Requirements:** Users must create an account to participate in problem-solving and competitions. The registration process requires a valid email address.

Each module was developed separately and then integrated to ensure that all components function cohesively. The modular design allowed for iterative development and testing, improving the overall system quality.

4.6 Prototype Design

The prototype design of the Online Judge System was developed to visualize and refine the system's user interface and functionality before full-scale implementation. Wireframes and mockups were created using design tools and translated into React.js components representing key interfaces, including user dashboards, problem submission forms, and competition management pages. User feedback was gathered through usability testing to identify areas for improvement. The prototype was iteratively refined based on feedback to ensure an intuitive and user-friendly experience. This design phase was crucial in validating the system's concept and layout, paving the way for more detailed development and integration.

4.7 System Requirements for End-User

For the Online Judge System to function effectively, end-users need to meet certain system requirements:

- **Hardware Requirements:** A modern computer or laptop with at least 4GB of RAM and a stable internet connection.
- **Software Requirements:** A recent version of a web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge. Users should also have a text editor or IDE for coding purposes.
- **Account Requirements:** Users must create an account to participate in problem-solving and competitions. The registration process requires a valid email address.

These requirements ensure that users can interact with the platform smoothly and efficiently.

Table 1: Comparison of Features in Various Systems

Paper No.	**Code Contribution**	**Contest**	**User Tracking Dashboard**
[3]	No	No	No
[5]	No	No	No
[7]	No	No	No
Proposed Method	Yes	Yes	Yes

4.8 Testing & Quality Assurance

Testing and quality assurance were integral to the development of the Online Judge System. The system underwent multiple testing phases:

- **Unit Testing:** Individual modules were tested in isolation to verify their functionality and ensure they met design specifications. Unit tests for the backend were conducted using tools like Mocha and Chai, while React Testing Library was used for the front end. Below is an example of unit tests written for a compiler using Jest.

- **Integration Testing:** Integrated modules were tested together to confirm that they interacted correctly and that data flowed seamlessly between them. Express.js routes were tested to ensure proper API interactions with the front end.
- **System Testing:** The entire system was tested as a whole to validate end-to-end functionality and ensure that all requirements were met. System testing ensured that React.js components, Express.js routes, and MongoDB interactions were working together seamlessly.
- **Performance Testing:** The system was tested under various load conditions to assess its performance, scalability, and response times. Tools like Apache JMeter were used to simulate load on the Node.js server and MongoDB.

Quality assurance practices included code reviews, adherence to coding standards, and continuous integration to maintain high code quality and system stability.

4.9 Summary

This chapter detailed the implementation and testing processes of the Online Judge System. The system setup involved configuring the development environment and deploying essential software components. System modules were designed and integrated to provide a comprehensive platform for problem-solving and competitions. Prototype design helped refine user interfaces, while end-user requirements ensured compatibility and usability. Rigorous testing and quality assurance practices validated the system's functionality and performance. Together, these efforts culminated in a robust and user-centric online judge platform.

Chapter 5

5 Standards, Constraints, and Milestones

5.1 Introduction

In the development of the Online Judge Recommendation System, adherence to standards and managing constraints are critical to ensuring the system's success and sustainability. This chapter outlines the various standards followed, the societal impacts of the system, ethical considerations, and the challenges encountered during development. Additionally, it discusses constraints such as design, component, and budget constraints, and presents a timeline with milestones to track progress. By addressing these aspects, we aim to provide a comprehensive understanding of the project's framework and the factors influencing its implementation and success.

5.2 Sustainability Standards

Sustainability standards are crucial for ensuring that the Online Judge Recommendation System remains viable and environmentally friendly throughout its lifecycle. Key sustainability standards include:

- **Energy Efficiency:** Implementing efficient coding practices and optimizing server operations to reduce energy consumption.
- **Resource Management:** Utilizing server resources effectively to minimize waste and enhance performance.
- **Long-Term Maintainability:** Designing the system with modularity and scalability in mind to facilitate future updates and maintenance.
- **Compliance with Green IT Policies:** Adhering to industry standards and best practices for sustainable IT operations and reducing the carbon footprint.

These standards help ensure that the system not only meets current needs but also contributes positively to long-term environmental and operational sustainability.

5.3 Impacts on Society

The Online Judge Recommendation System has several societal impacts, including:

- **Educational Benefits:** Enhances learning opportunities for students and professionals by providing a platform for practicing programming skills and participating in competitions.
- **Community Building:** Fosters a global community of problem solvers and developers, encouraging collaboration and knowledge sharing.
- **Economic Impact:** Supports the growth of the tech industry by developing a skilled workforce and providing a venue for competitive programming that can lead to job opportunities.
- **Accessibility:** Provides an open platform for users from various backgrounds to contribute and solve problems, promoting inclusivity and diversity in the tech field.

These impacts highlight the system's role in advancing education, fostering community engagement, and supporting economic and social development.

5.4 Ethics

Ethical considerations are fundamental in the development and deployment of the Online Judge Recommendation System. Key ethical aspects include:

- **Data Privacy:** Ensuring user data is protected through robust encryption and compliance with data protection regulations.
- **Fairness:** Implementing unbiased algorithms for problem recommendations and competition scoring to ensure a level playing field.
- **Intellectual Property:** Respecting and managing intellectual property rights related to user-contributed problems and solutions.
- **Transparency:** Providing clear information about system functionalities, data usage, and any potential biases in recommendations.

Addressing these ethical considerations helps build trust with users and stakeholders, ensuring the system operates in a responsible and transparent manner.

5.5 Challenges

The development of the Online Judge Recommendation System encountered several challenges:

- **Technical Complexity:** Integrating diverse system modules and ensuring seamless interaction between them posed significant technical challenges.
- **Scalability:** Designing a system capable of handling a growing number of users and problem submissions required careful planning and resource management.
- **User Engagement:** Attracting and retaining active users and contributors demanded effective marketing and user experience strategies.
- **Security Concerns:** Protecting the system from potential security threats and ensuring data integrity required ongoing vigilance and robust security measures.

Addressing these challenges was essential for the successful implementation and operation of the system.

5.6 Constraints

The development of the Online Judge Recommendation System faced several constraints:

- **Design Constraints:** The system design needed to balance functionality with usability, ensuring that the interface is intuitive while supporting a wide range of features.
- **Component Constraints:** Limited by the capabilities of existing technologies and libraries, requiring careful selection and integration of components to meet system requirements.

- **Budget Constraints:** Financial limitations impacted the scope of development, necessitating prioritization of features and efficient resource allocation.

These constraints influenced the design and implementation of the system, requiring careful management to achieve project goals within the given limitations.

5.7 Timeline and Gantt Chart

The project timeline outlines the key phases and milestones of the Online Judge Recommendation System's development. The Gantt chart provides a visual representation of the project schedule, highlighting critical tasks and their dependencies. Key milestones5.1 include:

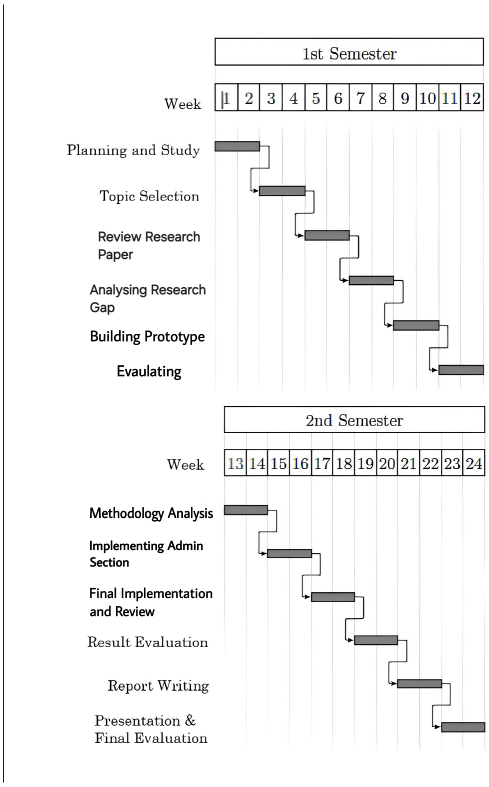


Figure 5.1: Gantt Chart.

- **Project Initiation:** Setting up the development environment and defining project scope.
- **Module Development:** Completing the development of individual system modules.

- **Integration and Testing:** Integrating modules and conducting comprehensive testing.
- **Prototype Design and Feedback:** Designing and refining the prototype based on user feedback.
- **Final Deployment:** Launching the system and monitoring initial performance.

The Gantt chart illustrates the timeline for each milestone, providing a roadmap for project management and tracking progress.

5.8 Summary

This chapter has explored the standards, constraints, and milestones associated with the development of the Online Judge Recommendation System. Sustainability standards ensure the system's long-term viability and environmental responsibility. The societal impacts highlight the system's contributions to education, community building, and economic development. Ethical considerations focus on data privacy, fairness, and transparency. Challenges such as technical complexity and scalability were addressed throughout the development process. Constraints related to design, components, and budget influenced project decisions and outcomes. The timeline and Gantt chart provide a structured overview of the project's progression, ensuring that key milestones are met. Together, these elements form a comprehensive framework for understanding the project's development and success.

Chapter 6

6 Conclusion

The Online Judge Recommendation System represents a significant advancement in the field of competitive programming and problem-solving platforms. By integrating robust features for problem contribution, solution submission, and competitive programming, the system offers a comprehensive and user-centric solution. The development process involved meticulous planning, adherence to standards, and addressing various challenges and constraints. The system's design and functionality aim to foster a global community of problem solvers, enhance learning opportunities, and support the growth of the tech industry. This chapter summarizes the project's key achievements, reflects on its limitations, and outlines future directions for further development and enhancement.

6.1 Limitations

Despite the system's success in providing a functional and engaging platform, several limitations were identified:

- **Scalability Issues:** While the system was designed to handle a moderate number of users, further optimization may be needed to accommodate a significantly larger user base and increased problem submissions.
- **Performance Overheads:** Some modules, particularly those involving real-time problem-solving and competition management, may experience performance bottlenecks under high load conditions.
- **User Experience Variability:** The diversity of user preferences and skill levels may lead to varied experiences, requiring ongoing adjustments to ensure a consistently positive user experience.
- **Limited Integration with External Tools:** The current system may lack

seamless integration with some external development tools and platforms, which could enhance user productivity and engagement.

- **Security Vulnerabilities:** While robust security measures were implemented, no system is entirely immune to potential vulnerabilities, necessitating continual security assessments and updates.

These limitations highlight areas for improvement and provide a foundation for addressing future challenges.

6.2 Future Works and Direction

To enhance the Online Judge Recommendation System and extend its capabilities, several avenues for future work have been identified:

- **Scalability Enhancements:** Implementing advanced optimization techniques and cloud-based solutions to improve scalability and manage increasing user activity more effectively.
- **Performance Optimization:** Refining system components and algorithms to reduce latency and enhance performance, especially during peak usage times.
- **User Experience Improvements:** Conducting further research into user needs and preferences to tailor the interface and features, ensuring a more personalized and engaging experience.
- **Integration with External Tools:** Exploring integration possibilities with popular coding environments, version control systems, and other development tools to streamline workflows for users.
- **Enhanced Security Measures:** Continuously updating security protocols and performing regular vulnerability assessments to protect user data and maintain system integrity.
- **Expansion of Features:** Introducing new features such as advanced analytics, collaborative problem-solving options, and expanded competition formats to increase user engagement and functionality.

These future directions aim to build upon the current system's success and address identified limitations, driving continuous improvement and innovation in the platform.