Placement Preparation Module Focusing on DSA

Project report submitted to
Dr. Babasaheb Ambedkar Technological University, Maharashtra
in partial fulfilment of the requirements for the award of
the degree

Bachelor of Technology
In
Computer Engineering

By

Suyash Patalbansi (2046491245052)
Shreya Raut (2046491245047)
Shreya Chinchmalatpure (2046491245048)
Susmita Kujur (2046491245051)

under the guidance of

Prof. Sheetal Kale
Professor and Head of the Department

Department of Computer Engineering
Bajaj Institute of Technology
Wardha-442 001 (India)
2023-24

BAJAJ INSTITUTE OF TECHNOLOGY, WARDHA

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project report titled

PLACEMENT PREPARATION MODULE FOCUSING ON DSA

has been successfully completed

by

Suyash Patalbansi, (2046491245052)

Shreya Raut, (2046491245047)

Shreya Chinchmalatpure, (2046491245048)

Susmita Kujur, (2046491245051)

in partial fulfilment of the requirements for the award of

the degree in Bachelor of Technology, Computer Engineering

Prof. Sheetal Kale
Guide
Department of Computer Engineering
Bajaj Institute of Technology, Wardha

Prof. Sheetal Kale
Head of the Department
Department of Computer Engineering
Bajaj Institute of Technology, Wardha

Dr. N. M. Kanhe
Principal
Bajaj Institute of Technology, Wardha

Date: 28/11/2023
Place: Wardha

DECLARATION

We, hereby declare that the project report titled “Placement Preparation Module Focusing On DSA” submitted by us to the Bajaj Institute of Technology, Wardha, in partial fulfilment of the requirement for the award of Degree of B. Tech. in Computer Engineering discipline is a record of bonafide project work carried out by us under the guidance of Prof. Sheetal Kale Professor and Head of the Department.

We, further declare that this submission by the undersigned represents our original work and We have quoted the references where others ideas/words have been included. We understand any violation of the above will levy a disciplinary action on us.

We, further declare that the work reported in this project report has not been submitted either in-part or in-full for the award of any other degree in any other Institute or University.

Date:- 28/11/2023

Place:- Wardha

S. No.	Name of Student	PRN Number	Signature
1.	Suyash Patalbansi	2046491245052	
2.	Shreya Raut	2046491245047	
3.	Shreya Chinchmalatpure	2046491245048	
4.	Susmita Kujur	2046491245051	

ACKNOWLEDGEMENT

We wish to extend my sincere gratitude to project guide to Prof. Sheetal Kale, Professor and Head of the Department for having made available her valuable time, support and guiding us throughout this project with timely inputs and suggestions.

We are extremely grateful to Dr Narendra Kanhe, the Principal, Bajaj Institute of Technology, Wardha for providing required resources for the successful completion of our project.

Date: 28/11/2023

Place: Wardha

S. No.	Name of Student	PRN Number	Signature
1.	Suyash Patalbansi	2046491245052	
2.	Shreya Raut	2046491245047	
3.	Shreya Chinchmalatpure	2046491245048	
4.	Susmita Kujur	2046491245051	

ABSTRACT

Data Structure and Algorithms is an important basic course for computer science and other information related majors. It carries a tremendous responsibility for cultivating students' computer program design capabilities and computational thinking skills. Data Structures is a course that combines theory with practice. Students are not only required to possess a solid foundation of higher mathematics and data structures, but also a strong ability for abstract thinking and computer programming. The traditional teaching of data structures pays most attention to mathematical reasoning, proving correctness and the complexity analysis of algorithms. Students need regular practice and programming skills to improve in their professional aspects.

To address this issue, we have developed a web application engineered on the MERN stack—MongoDB, Express.js, React.js, and Node.js. that provide students with DSA content and quizzes and related programming questions to improve user's skills and knowledge in DSA. The web application helps user prepare for placements and improve their knowledge and skills so that they perform better in exams. User Authentication utilizes secure authentication mechanisms for user account creation and login. Offers structured lessons on DSA concepts and provides interactive quizzes based on difficulty levels.

Website have integrated coding practice section featuring a versatile code editor supporting multiple programming languages and also implemented Q&A system for users to engage in discussions related to DSA topics. Implemented visual representations of performance metrics for enhanced user engagement. Chatbot feature is incorporated which utilizes ChatGPT API for user assistance that enhances user interactions through real-time responses and guidance. There is notification system which provide users reminders and Cheatsheets are provided for summarizing and quick revision of key time and space complexities for common algorithms and data structures.

Keywords – DSA, MERN, Code Editor, Quizzes, Programming, Placements.

TABLE OF CONTENTS

ABBREVIATIONS

DSA	Data Structures and Algorithms
CS	Computer Science
IT	Information Technology
LMS	Learning Management Systems
API	Application Programming Interface
Q&A	Question and Answer
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets

CHAPTER 1
INTRODUCTION

1.1 Introduction

Data structures and Algorithms are really important courses for every concept used in computer science and software engineering. There are many concepts involved in the subjects of Data structures and Algorithms. There will be many questions for educators like how to learn Algorithms and Data structures, as there are many concepts involved without getting confused during attending the course.

Learning problems in the data structures course have always been the topic of discussion among lecturers in computer science [1] because this course has a high number of failures. The lecturers have used various approaches, methods, strategies, patterns, and technologies, but there has been no appropriate solution to be applied. Different levels of acceptance, understanding or learning capacity for each student are a challenge for lecturers. The failure to learn the course data structure for students occurs because this course requires abstract thinking. Many theories require research and deep understanding. Also, the density of teaching materials and its contents and methods are different from one subject to another. This requires high thinking skills in order to understand and learn the data structures course [9].

Data Structures is a course that combines theory with practice. Students are not only required to possess a solid foundation of higher mathematics and data structures, but also a strong ability for abstract thinking and computer programming. The traditional teaching of data structures pays most attention to mathematical reasoning, proving correctness and the complexity analysis of algorithms. Developing computing skills at higher learning levels, including the ability to apply the learned concepts in practice, requires using active learning and motivational strategies. Active learning is especially important in introductory Data Structures courses

In order to improve the educational process, it is necessary to motivate and engage the student and make them practice regularly. The placement preparation DSA module is a dynamic website for students to build a strong foundation in data structures and algorithms, providing them with structured learning resources, visualized learning

materials, practice quizzes and challenging programming questions. The placement preparation DSA module website provides a comprehensive and supportive learning environment to the students to prepare better for the placements, and ultimately helping students master the essentials DSA concepts.

1.2 Motivation

Traditionally, instructors in lecture-based courses in introductory computer science have an expectation that the material presented in class is reinforced via textbook reading, lab exercises, assigned homeworks or quizzes. Given the rapid increases in the CS population over the past few years [7], these courses are large, and monitoring the performance of all students and reaching out to at-risk students is a challenge. While students may be given practice problems to reinforce material learned in class, the time-consuming nature of many of these assignments, both for students and for graders, mean they can only cover a small amount of the material that students are expected to understand. Students require feedback on their performance to either verify their comprehension or identify where they made mistakes. Identifying and addressing student's misconceptions early is vital to ensuring students have a strong foundation on which to build future knowledge and will help reduce their frustration. Another limitation is the finite set of examples and practice problems available to students under this model. Some students rely heavily on examples to learn material and the limited number presented in class or in the textbook may not be sufficient for students to gain comprehension, especially when instructors use the same textbook examples in their lectures. Likewise, whether homework problems come from the instructor or a textbook, the limited number of problems covering a particular topic in turn limits the amount a student is able to practice.

This has resulted in new ways of teaching such courses, broadly classified as active learning techniques, that can include any combination of lab-based instruction, flipped classroom settings, gamification, peer-learning, or use of multimedia content all of which attempt to better engage students. Our focus in this work is towards building new interactive learning modules and analyzing student performance in data structures and algorithms courses. These modules can help reinforce lecture content outside of class.

The nature of computing and the generation of students have changed drastically in the past years. Still, most higher education computing courses are taught in traditional ways that may not be adequate to keep up with the changing educational reality and may not support the necessary learning. Developing computing skills at higher learning levels, including the ability to apply the learned concepts in practice, requires using active learning and motivational strategies. Active learning is especially important in introductory Data Structures courses that teach fundamental conceptual and programming constructs [8]. Improving the learning of these concepts is crucial to enhance the formation of professionals in the area [3].

1.3 Objectives

- Create a robust module for Data Structures and Algorithms (DSA) to aid in placement preparation.
- To implement topic-wise and level-wise quizzes along with challenging coding questions to enhance problem-solving skills.
- To generate detailed graphical reports to provide users with insights into their performance and learning activities.
- To provide users with concise study material covering the entire DSA syllabus for effective learning.
- To facilitate a user-driven question and answer system to encourage interaction and resolve doubts among users.
- Include a cheatsheet feature for quick and summarized revision of the DSA syllabus.
- To design a structured DSA learning roadmap to guide users in their preparation journey effectively.
- To allow users to view content without login but require authentication for quizzes, coding questions, discussions, and dashboard access. Implement access control for administrators.
- Provide administrators with tools to control and manage content, quizzes, and coding questions for effective platform administration.
- To integrate a code editor with support for multiple programming languages to cater to diverse user preferences.

- To implement a notification system to keep users informed about updates, reminders, and relevant activities on the platform.

1.4 Scope of Work

The implementation of a web-based platform to aid in placement preparation by providing a space for student to improve their knowledge and skills in Data Structures and Algorithms. Our focus in this work is towards building new interactive learning modules and analyzing student performance in data structures and algorithms courses by providing them with structured learning resources, visualized learning materials, practice quizzes and challenging programming questions. These modules can help reinforce lecture content outside of class.

By leveraging the capabilities of Web and Internet, the web application can be accessed through internet at anytime and anywhere. We have provided comprehensive study material covering the entire DSA syllabus. Also developed challenging programming questions and quizzes for different DSA topics and levels to reinforce and practice the knowledge gained through contents. programming questions will help users to develop programming skills and improve problem solving ability. The platform's design focuses on an intuitive and user-friendly interface for seamless navigation. Set up a secure user authentication system to implement access controls to restrict certain features to authenticated users. These will help in tracking user’s progress. We have developed a user-driven question and answer system to facilitate doubt resolution. Include a cheatsheet feature for users to quickly revise and summarize key concepts. Platform include a structured DSA learning roadmap to guide users through their preparation. Platform have integrated code editor with support for multiple programming languages. Users informed about updates, reminders, and activities through notification system to keep them engaged and motivated. Users are provided graphical reports with insights into their performance and learning activities. Administrators are given tools and features for to manage and control content on the platform.

1.5 Organization of Report

The report is organized in six chapters

CHAPTER 1: Introduction. The chapter gives a brief introduction to Data Structures and Algorithms and its importance in computer science. The chapter also discusses the motivation and objective behind the project it also contains the chapter organization.

CHAPTER 2: Literature Review. The chapter gives an overview on some of the prior research done which are directly related to the project. The chapter also contains a summary of literature review.

CHAPTER 3: Methodology/System requirements and Specification The chapter explains the working and the methodology of developing web application for learning DSA. It also contains the technology, frameworks and tools used in project. And also, the hardware and software requirements of the project.

CHAPTER 4: Implementation of Project. The chapter contains implementation of project work. The chapter gives overview of implementation of software, tools and framework used in this project.

CHAPTER 4: Results and Discussions. The chapter contains results obtained by the project. The chapter gives explanation of results obtained through our web application.

CHAPTER 6: Conclusion and Future Scope. The chapter gives the conclusion to the project work done in this report. The chapter discusses the scope of The Placement Preparation DSA Module. The potential of Placement Preparation DSA Module is also mentioned in this chapter.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Review

2.1.1 Learning Management System for Data Structures and Algorithm

Year of Publication: 2021

Author: Marco Paulo J. Burgos

Objective: To develop of Learning Management System (LMS) for Data Structures and Algorithms with the main feature that allows students to take modular online learning.

Contribution: The study addressed the adjustments of academic institutions to online class and modular learning caused by the Covid-19 pandemic. The research and development approach includes (i) stages of system development using the Waterfall Method, (ii) level of acceptability of the developed system based on the ISO 25010 standard, (iii) difference in the evaluation of the three groups of respondents, (iv) challenges encountered while using the system, and (v) implementation plan. The respondents chosen through convenience sampling were 60 students, 15 faculty members, and 15 Information Technology (IT) experts. The checklist-format questionnaire was based on the ISO 25010 which determined acceptability using functional suitability, performance efficiency, usability, and reliability criteria. An interview was also conducted to evaluate respondents' experiences using the system. Based on the respondents' evaluation, the developed system was 'acceptable' as reflected by the obtained weighted means. Further results showed no significant difference in the evaluation of the three groups of respondents in terms of functional suitability, performance efficiency, usability, and reliability.

Conclusion: The Waterfall Model was used in the development of LMS for Data Structures and Algorithms. This method encompassed several methods which are requirements analysis, system design, implementation, testing, deployment, and maintenance phases. The overall assessment indicated that the system is 'acceptable' as supported by the weighted means of 4.49 for the students, 4.43 for the faculty members, and 4.38 for the IT experts. Furthermore, the weighted means for each component are 4.51 for functionality, 4.38 for performance, 4.45 for usability, and 4.39 for reliability. This numerical evidence proves that the LMS for Data Structures and Algorithms serves its purpose and greatly benefits the end users especially in terms of functionality, performance, usability, and reliability in accordance with the standards of ISO 25010. An implementation plan was also developed for the use of the LMS. It shows comprehensive procedures that elucidate the distinct functionalities of the system. A Gantt chart was created to have an idea of the timeline for the implementation. The methods in the user manual have been evaluated and tested to facilitate the understanding of the detailed usage of the LMS.

2.1.2 Some ways of increasing the efficiency of teaching data structures

Year of Publication: 2022

Author: Zarema S. Seidametova.

Objective: To present some ways of increasing the efficiency of teaching data structures (hashing, trees) during the Algorithms and data structures course.

Contribution: Nowadays computer modelling and simulation are widely used in computer science education. Many concepts of computer science and software engineering are difficult for students to understand. Algorithms and Data structures are one of the most important subjects in Computer Science, which is required by all programmers. Programmers need to know how to handle the data in huge quantities, how the data should be stored in the memory and how the memory should be used efficiently. There are several ways of increasing the efficiency of the educational process. One of such ways is visualization, the other way is flipped classroom concept. They conducted two experiments with four study groups of two subjects of the Algorithms and Data Structures – (1) Hashing, (2) Trees (BST, RBT, AVL), – taught second-year bachelor students. In the first experiment, study groups were prepared on the subjects (1) and (2) by the same level of teaching technique with or without visualization tools. Unlike the first experiment, in the second experiment, they used flipped classroom concepts for one study group and a standard way of teaching for another group.

Conclusion: There are a lot of approaches to the implementation of some ways of increasing the efficiency of teaching data structures (hashing, trees). They conducted two experiments with four study groups of two subjects of the Algorithms and Data Structures – (1) Hashing, (2) Trees (BST, RBT, AVL), – taught second-year bachelor students. In the first experiment, study groups were prepared on the subjects (1) and (2) by the same level of teaching technique with or without visualization tools. Unlike the first experiment, in the second experiment, they used flipped classroom concept for one study group and a standard way of teaching for another group. Flipped classroom and visualization as pedagogical tools can be used to teach students the necessary skills and competencies in algorithms and data structures.

2.1.3 Interactive Data Structure Learning Platform

Year of Publication: 2014

Author: Estevan B. Costa1, Armando M. Toda, Marcell A. A. Mesquita,

Fabio T. Matsunaga, Jacques D. Brancher

Objective: To develop and implement an interactive web learning environment, called DSLEP (Data Structure Learning Platform), to support students in higher education IT courses.

Contribution: The advent in technology in the past few years allowed an improvement in the educational area, as the increasing in the development of educational system. One of the techniques that emerged in this lapse is called Gamification, defined as the utilization of video game mechanics outside its bounds. Researchers in this area found

positive results in the application of these concepts in several areas from marketing to education. Among higher education, focusing on IT courses, Data Structures can be considered an important subject to be taught, as they are base for many systems. The system includes basic concepts taught on this discipline as stacks, queues, lists, arrays, trees. The system is also implemented with gamification concepts, as points, levels, and leader boards, to motivate students in the learning process and stimulate self-learning.

Conclusion: This work presents the development and implementation of an Interactive Learning Environment for Data Structures which was also proposed by other authors. To improve the gamification core of the system, 33 achievements and 9 leader boards were made to be used within the system and its 7 activities. The objective of this work was achieved since the platform is finished and already being tested with students from local colleges. Also, it is implemented with the gamification concepts to engage and stimulate their motivation. To secure and guarantee the information stored within the system, we used a tool (Clay.IO) to manage the gamification concept. This also allowed the user to exchange personal information with others, as a social network system. As future works, new activities and new concepts will be implemented. Also, it is in development a profile analyser to work within the system so the teachers may create a 'class' to monitor the student's development and progress within the lessons. Those analysis also may determine which activities may be improved or balanced.

2.1.4 Research on Data Structure Course Teaching Based on Open Teaching System Model

Year of Publication: 2019

Author: Lasheng Yu, Xiaopeng Zheng and Ye Biao

Objective: To develop students' learning initiative, thinking innovation and independent thinking ability in the process of data structure teaching which may be useful to the other instructors in the universities.

Contribution: Data Structure is an important basic course for computer science and other information related majors. It carries a tremendous responsibility for cultivating students' computer program design capabilities and computational thinking skills. Data Structure is the core course in the computer science curriculum system, and corresponds to the theories, abstractions, and design methodology of problem solving in the discipline. This article mainly discusses how to develop students' learning initiative, thinking innovation and independent thinking ability in the process of data structure teaching which may be useful to the other instructors in the universities.

Conclusion: This paper takes the data structure as an example to illustrate the application of open teaching in the data structure curriculum teaching system, from the curriculum syllabus, teaching content arrangement, teaching process principles and methods, coursework settings, course assessment mechanism and other teaching process. Starting from the requirements of bringing student's subjective initiative into cultivating students' innovative awareness and ability to innovate, we will grasp the main problems of students' ability to solve complex problems, and do a good job of continuously improving the quality of course teaching and the mastery of students'

ability, and promote the achievement of graduation requirements, which in turn supports the realization of training goals. Through the reform of the data structure course teaching model, significant results have been achieved, teaching quality and teaching levels have been improved, and the teaching reform has been demonstrated as a model, and students' ability in innovation and entrepreneurship has been improved and local economic development has been served.

2.1.5 Studied Questions in Data Structures and Algorithms Assessments

Year of Publication: 2023

Author: Iris Gaber, Amir Kirsh and David Sttater

Objective: To argue that a good exam should contain questions that students have seen during the semester, and that the grading of those questions should be strict.

Contribution: Designing a proper exam that accurately evaluates students' knowledge and skills is one of the important tasks of every teacher. The format of the exams affects the way students learn throughout the course, and a well-designed exam can enhance meaningful learning. We describe a case study which, over three semesters, supports the claim that answering these questions require the "Understand" level of Bloom's taxonomy, and that this strategy fosters more meaningful learning and better assesses students' knowledge.

Conclusion: They showed that having "studied questions" in the exam does not impair the assessment process. Solving studied questions is not trivial, and, in our case study, still produced variance within the grades, so these questions can still differentiate between students who understood the material and those who did not. Studied questions are also perceived as fair. Students do not consider something they saw as tricky or difficult. It also promotes their self-efficacy and reduces their anxiety, as presented in the questionnaire that we distributed. There are several ways to present "Studied Questions" in an exam. One of the ways is to create a pool of questions that would be shared with the students, then have the exam draw questions from that pool. Our experience is that when a large percentage of the exam (between 50% and 70%) is based on studied questions, it enhances meaningful learning, improves the fairness of the exam and reduces students' anxiety.

2.1.6 Learning Management System (LMS) among University Students: Does It Work?

Year of Publication: 2013

Author: Nor Azura Adzharuddin and Lee Hwei Ling

Objective: To provide several insights of the LMS phenomenon.

Contribution: The Learning Management System (LMS) has been established in a number of universities worldwide to help connect students and lecturers without the confines of the traditional classroom. It is an environment with digital software which is designed to manage user learning interventions as well as deliver learning content and resources to students. Since the LMS system has already been implemented and it has also been made compulsory for the lecturers to apply in their daily lectures, it is

vital to identify feedback of students as users of LMS. Previous studies have shown various findings in relation to the impact of using LMS in the higher learning environment in various universities worldwide.

Conclusion: In this modern world where information is disseminated quickly via the internet, the LMS is an essential tool for university students as not they can keep updated with their coursework, but get instant notifications pertaining to their daily assignments. In turn, lecturers have an easier time reaching out to their students out of class hours and can instantly update them over the LMS about issues regarding their coursework. Although those using the LMS might encounter some problems, it's all part and parcel of learning and using a whole new system altogether. Universities should provide proper training and guidance for students and lecturers using the LMS, as well as have a team which is on-call at all times to solve any problems that may arise. Nevertheless, most university students have access to their university's LMS or similar systems that help to enhance their learning process. Many have also expressed positive views about LMS, therefore proving that LMS is a necessary implementation in all universities worldwide.

2.1.7 Visualization, Assessment and Analytics in Data Structures Learning Modules

Year of Publication: 2018

Author: Matthew Mcquaigue, David Burlinson, Kalpathi Subramanian, Erik Saule, Jamie Payton

Objective: To build interactive teaching modules for data structures and algorithms courses.

Contribution: In recent years, interactive textbooks have gained prominence in an effort to overcome student reluctance to routinely read textbooks, complete assigned homeworks, and to better engage students to keep up with lecture content. Interactive textbooks are more structured, contain smaller amounts of textual material, and integrate media and assessment content. They build interactive teaching modules for data structures and algorithms courses with the following characteristics, (1) the modules are highly visual and interactive, (2) training and assessment are tightly integrated within the same module, with sufficient variability in the exercises to make it next to impossible to violate academic integrity, (3) a data logging and analytic system that provides instantaneous student feedback and assessment, and (4) an interactive visual analytic system for the instructor to see students' performance at the individual, sub-group or class level, allowing timely intervention and support for selected students.

Conclusion: Their current implementation demonstrates a prototype of the parts of the system. They have presented new learning modules that can be used as a part of an interactive textbook in the data structures and algorithms courses. Their module are built to work with the OpenDSA system, using JSAV library for generating the module visualizations that have been customized to provide a highly interactive, visual and engaging experience for students. Much work remains to be done in connecting the various components and integrating the data into the visualization modules, prior to

routine use as part of the LMS. This will be followed by formal user studies in data structures and algorithms courses. Finally, more work is needed in developing the visual analytics system to better understand the needs of the instructors as well as identification of at-risk students for timely intervention and support.

2.1.8 Active Learning through Game Play in a Data Structures Course

Year of Publication: 2018

Author: Darina Dicheva, Austin Hodge.

Objective: To present an educational game intended to explicate several features hindering student's understanding of the data structure Stack on conceptual and practical level.

Contribution: Data Structures is a fundamental Computer Science discipline, challenging students' abstract thinking, problem solving and programming skills. In this paper, they present an educational game intended to explicate several features hindering students' understanding of the data structure Stack on conceptual and practical level. The game targets all three aspects of teaching data structures: conceptualization, application and implementation. These aspects are embodied as three parts of the game tied together through a meaningful storyline. The application part targets the use of stacks to solve problems, such as converting arithmetic expressions from infix to postfix notation and evaluating postfix and infix expressions. The implementation part involves solving Parson's problems and writing Java code for implementing the methods of the Stack class. The results of the conducted evaluation of the game show statistically significant learning gains for the students and a strong positive attitude towards this type of active learning.

Conclusion: Due to the difficulties that many students face in learning data structures, it is often problematic to impart working knowledge of their creation and operation by using only traditional teaching methods. As a response to these challenges, we designed the Stack Game. The objective was to enable students to acquire a correct cognitive model of the stack concept and behaviour through gameplay. The Stack Game was incorporated as part of the learning activities in a regular Data Structures course, to promote deeper and more active learning. Our experience with designing and integrating the game into the classroom experience shows the importance of embedding sound instructional design into games in order to create effective learning tools. In particular, offering a game that includes all aspects of the targeted topic helps students to reflect on the learned concepts in a holistic way. The manner in which learning experiences from games are incorporated into the classroom is another important factor to its effectiveness. Instructors can further facilitate the transfer of skills by discussions connecting the game experience with relevant concepts students are learning in class.

2.2 Summary of Literature Review

Above mentioned research papers and official documentation explains the architecture of Learning Management System for Data Structures and Algorithm and its implementation. Also, there is some documentation that discusses its implementation through visualizations and game-playing. It highlighted the importance of active learning in learning data structures and algorithms. It also provided various possible solutions using various methods of learning data structures and algorithms. These papers discuss the various techniques to improve programming skills of the user. Some documentation discussed about assessment and analytics of users and its importance in improving data structures and algorithms knowledge in user. There is also problem with quality of exam questions by which study and retention of data structure and algorithms. The nature of computing and the generation of students have changed drastically in the past years. Still, most higher education computing courses are taught in traditional ways that may not be adequate to keep up with the changing educational reality and may not support the necessary learning. Developing computing skills at higher learning levels, including the ability to apply the learned concepts in practice, requires using active learning and motivational strategies. Improving the learning of these concepts is crucial to enhance the formation of professionals in the area

CHAPTER 3
METHODOLOGY / SYSTEM REQUIREMENTS AND SPECIFICATIONS

3.1 Web Development

Web development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites. Web development is closely related to the job of designing the features and functionality of apps (web design). The term development is usually reserved for the actual construction of these things (that is to say, the programming of sites).

The basic tools involved in web development are programming languages called HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript. There are, however, a number of other programs used to "manage" or facilitate the construction of sites that would otherwise have to be done "from scratch" by writing code. A number of content management systems (CMS) fall into this category, including WordPress, Joomla!, Drupal, TYPO3, and Adobe Experience Manager, among others.

There are two main types of web development: front-end development, back-end development. Front-end development involves the "client-facing" side of web development. That is to say usually, front-end web development refers to the portion of the site, app, or digital product that users will see and interact with. A Back-End Developer creates the basic framework of a website before maintaining it and ensuring it performs the way it should, including database interactions, user authentication, server, network and hosting configuration, and business logic.

3.2 MERN Stack

The MERN stack is a web development framework made up of the stack of MongoDB, Express.js, React.js, and Nodejs. It is one of the several variants of the MEAN stack. MERN Stack is a JavaScript Stack that is used for easier and faster deployment of full-stack web applications. MERN Stack comprises of 4 technologies namely: MongoDB,

Express, React and Node.js. It is designed to make the development process smoother and easier.

- MongoDB: Non-Relational Database
- Express: Node.js web server
- React: JavaScript Frontend Framework
- Node: JavaScript Web Server

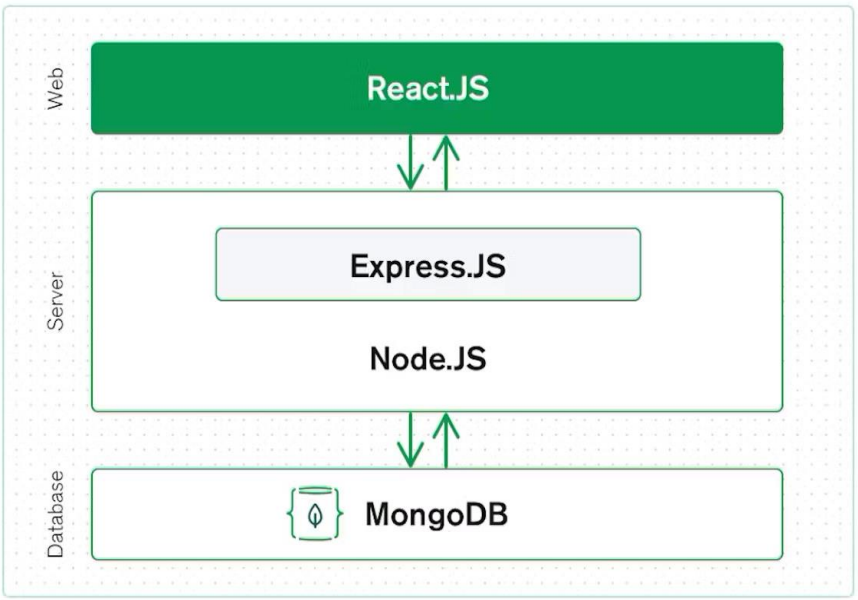

Fig. 3.1 Architecture of MERN Stack

When you use the MERN stack, you work with React to implement the presentation layer, Express and Node to make up the middle or application layer, and MongoDB to create the database layer. We will utilize these four technologies to develop a basic application.

3.2.1 MongoDB: Cross-platform Document-Oriented Database

MongoDB is a NoSQL database where each record is a document comprising of key-value pairs that are similar to JSON (JavaScript Object Notation) objects. MongoDB is flexible and allows its users to create schema, databases, tables, etc. Documents that are identifiable by a primary key make up the basic unit of MongoDB. Once MongoDB is installed, users can make use of Mongo shell as well. Mongo shell

provides a JavaScript interface through which the users can interact and carry out operations (Eg: querying, updating records, deleting records).

Reasons to use MongoDB:

- Fast – Being a document-oriented database, easy to index documents. Therefore, a faster response.
- Scalability – Large data can be handled by dividing it into several machines.
- Use of JavaScript – MongoDB uses JavaScript which is the biggest advantage.
- Schema Less – Any type of data in a separate document.
- Data stored in the form of JSON –
- Objects, Object Members, Arrays, Values, and Strings
- JSON syntax is very easy to use.
- JSON has a wide range of browser compatibility.
- Sharing Data: Data of any size and type (video, audio) can be shared easily.
- Simple Environment Setup – It's really simple to set up MongoDB.
- Flexible Document Model – MongoDB supports document-model (tables, schemas, columns & SQL) which is faster and easier.

3.2.2 Express: Back-End Framework

Express is a Node.js framework. Rather than writing the code using Node.js and creating loads of Node modules, Express makes it simpler and easier to write the back-end code. Express helps in designing great web applications and APIs. Express supports many middlewares which makes the code shorter and easier to write.

Reasons to use Express:

- Asynchronous and Single-threaded.
- Efficient, fast & scalable
- Has the biggest community for Node.js
- Express promotes code reusability with its built-in router.
- Robust API

3.2.3 React: Front-End Library

React is a JavaScript library that is used for building user interfaces. React is used for the development of single-page applications and mobile applications because of its

ability to handle rapidly changing data. React allows users to code in JavaScript and create UI components.

Reasons to use React:

- Virtual DOM – A virtual DOM object is a representation of a DOM object. Virtual DOM is actually a copy of the original DOM. Any modification in the web application causes the entire UI to re-render the virtual DOM. Then the difference between the original DOM and this virtual DOM is compared and the changes are made accordingly to the original DOM.
- JSX – Stands for JavaScript XML. It is an HTML/XML JavaScript Extension which is used in React. Makes it easier and simpler to write React components.
- Components – ReactJS supports Components. Components are the building blocks of UI wherein each component has a logic and contributes to the overall UI. These components also promote code reusability and make the overall web application easier to understand.
- High Performance – Features like Virtual DOM, JSX and Components makes it much faster than the rest of the frameworks out there.
- Developing Android/Ios Apps – With React Native you can easily code Android-based or IOS-Based apps with just the knowledge of JavaScript and ReactJS.

3.2.4 Node.js: JS Runtime Environment

Node.js provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node pack manager i.e. npm allows the user to choose from thousands of free packages (node modules) to download.

Reason to use Node.JS:

- Open-source JavaScript Runtime Environment
- Single threading – Follows a single-threaded model.
- Data Streaming
- Fast – Built on Google Chrome's JavaScript Engine, Node.js has a fast code execution.
- Highly Scalable

3.3 Methodology

3.3.1. Project Inception

- Conduct an extensive market analysis to identify the need for a DSA preparation module.
- Define project goals, objectives, and success criteria based on user requirements.
- Formulate a detailed project charter outlining scope, stakeholders, and deliverables.

3.3.2. Requirement Elicitation and Analysis

- Engage with stakeholders, including potential users and subject matter experts, to gather comprehensive requirements.
- Prioritize requirements based on their impact and relevance to the project goals.
- Develop user personas and use cases to guide the design process.

3.3.3. System Architecture Design

- Define the overall system architecture, including backend and frontend components.
- Choose appropriate technologies and frameworks based on scalability, security, and performance requirements.
- Design database schema and relationships to efficiently store and retrieve data.

3.3.4. Content Development

- Collaborate with subject matter experts to curate high-quality content covering the entire DSA syllabus.
- Ensure content is concise, clear, and aligned with the learning goals.

3.3.5. Quiz and Question Design

- Develop a systematic approach to create topic-wise and level-wise quizzes.
- Design challenging programming questions to assess and enhance problem-solving skills.

3.3.6. Performance Tracking System

- Implement a robust system for tracking user performance.

- Generate graphical reports to provide users with detailed insights into their progress.

3.3.7. Community Interaction Features

- Integrate a user-friendly Q&A platform to facilitate collaborative learning.
- Encourage user participation in asking and answering questions to build a supportive community.

3.3.8. Learning Resources

- Provide concise study materials for each DSA topic to supplement learning.
- Develop cheatsheets for quick revision of key concepts.

3.3.9. Learning Roadmap Development

- Create a structured learning roadmap to guide users through the DSA syllabus.
- Break down the syllabus into milestones for effective learning progression.

3.3.10. User Authentication and Accessibility

- Allow users to access content without login for convenience.
- Implement a secure login system for quizzes, coding questions, discussions, and dashboard interaction.

3.3.11. Admin Control Panel

- Design an admin interface for content management, ensuring admins can update quizzes, manage coding questions, and monitor platform activity.
- Design a comprehensive admin control panel for content management, user analytics, and system configuration.
- Ensure real-time monitoring of platform performance and user engagement metrics.

3.3.12. Code Editor Integration

- Integrate a versatile code editor that support multiple programming languages and frameworks to accommodate diverse user preferences.
- Ensure seamless integration with learning resources and quizzes.

- Ensure a seamless coding experience for users with features like syntax highlighting and code completion.

3.3.13. Notification System

- Develop a notification system to keep users informed about updates, activities, and upcoming events.
- Implement reminders for quizzes and other important milestones.

3.3.14. Testing and Quality Assurance:

- Conduct rigorous testing to identify and address any bugs or usability issues.
- Ensure the platform is user-friendly, responsive, and functions seamlessly across devices.

3.4 Data Structures and Algorithms

Data structures and Algorithms are really important courses for every concept used in computer science and software engineering. There are many concepts involved in the subjects of Algorithms and Data structures. There will be many questions for educators like how to learn Algorithms and Data structures, as there are many concepts involved without getting confused during attending the course. Data Structures is a course that combines theory with practice. Students are not only required to possess a solid foundation of higher mathematics and data structures, but also a strong ability for abstract thinking and computer programming.

Data structures topics:

- Linear data structures – array, stack, queue, linked list;
- Non-linear data structures – hashing (direct hash tables, collisions, open addressing, double hashing, perfect hashing), trees (BST, RBT, AVL), graphs, heaps;
- Augmented data structures.

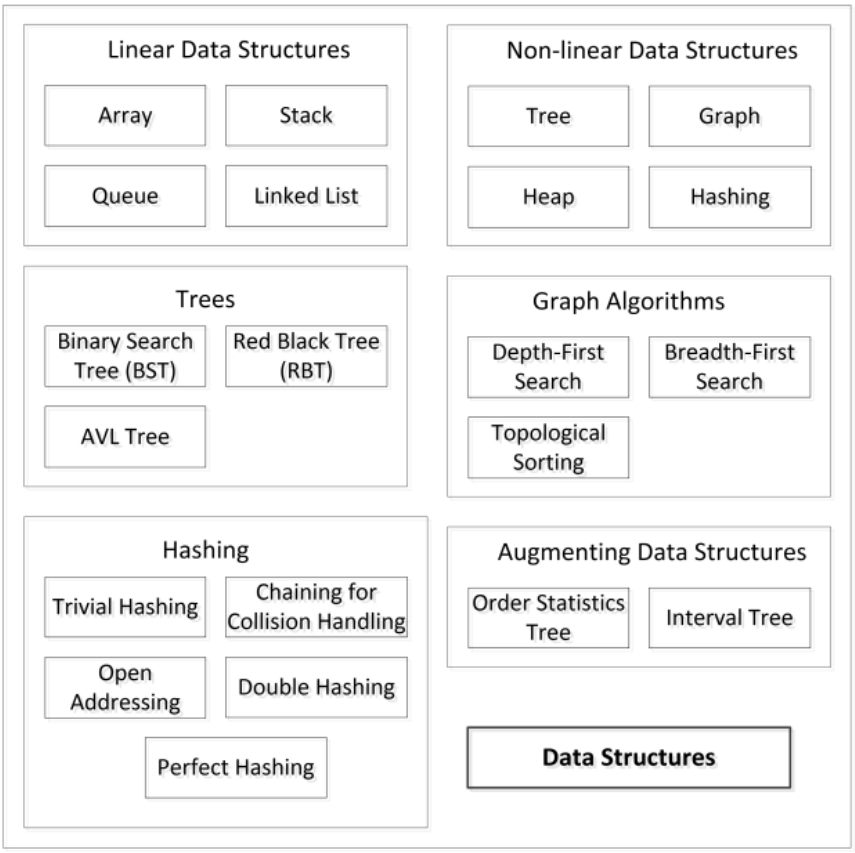

Fig. 3.2 Data structures

Algorithms and Data structures are used to design an efficient algorithm, organize data access, and economize the execution time of programs. Data structures are widely used in operating systems, database management systems, statistical soft-ware, compiler design, numerical analysis, and artificial intelligence simulations. Algorithms and Data structures play a significant role in the computer science and soft-ware engineering domain. The course of algorithms and data structure introduces the basic data type, such as byte and character string, and presents collective data types, linked lists, stacks, arrays, queues, trees, heaps, and priority queues and complex data structures and, and focuses on programming methodologies, problem-solving strategies, and algorithm development.

Algorithms includes such topics as:

- Algorithms analysis technics – asymptotic analysis and notations, Big-Oh, complexities of algorithms, worst, average, best cases of algorithm performing, recurrences, master theorem for solving some types of recurrence equations;

- Sorting algorithms with comparisons – bubble sort, selection sort, insertion sort, merge sort, heap sort, quicksort;
- Sorting algorithms without comparisons (in linear-time) – bucket sort, counting sort, radix sort;
- Searcher algorithms – linear search, binary search, jump search.

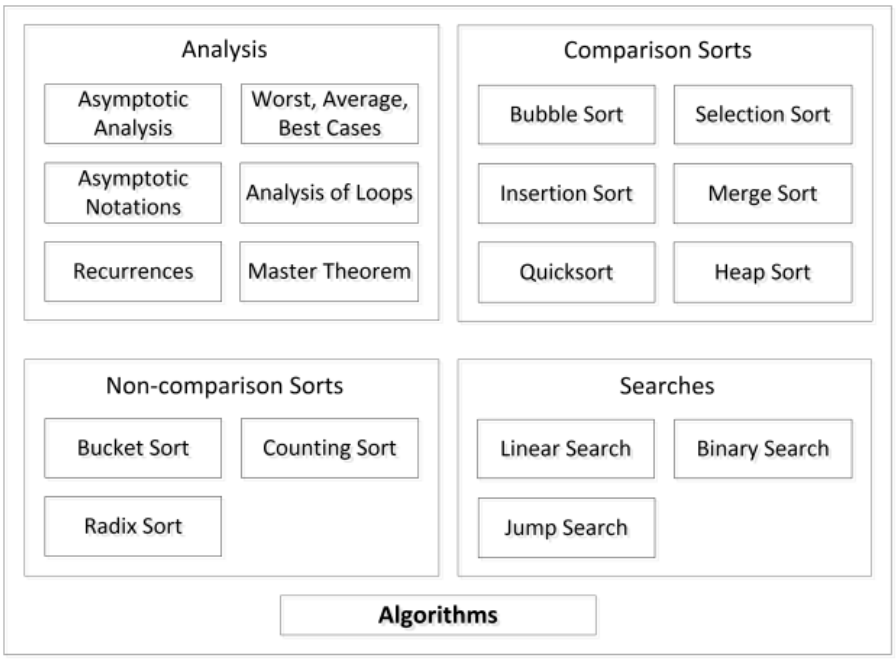

Fig. 3.3 Algorithms

3.5 Methods/Algorithm

3.5.1 System Flow Chart

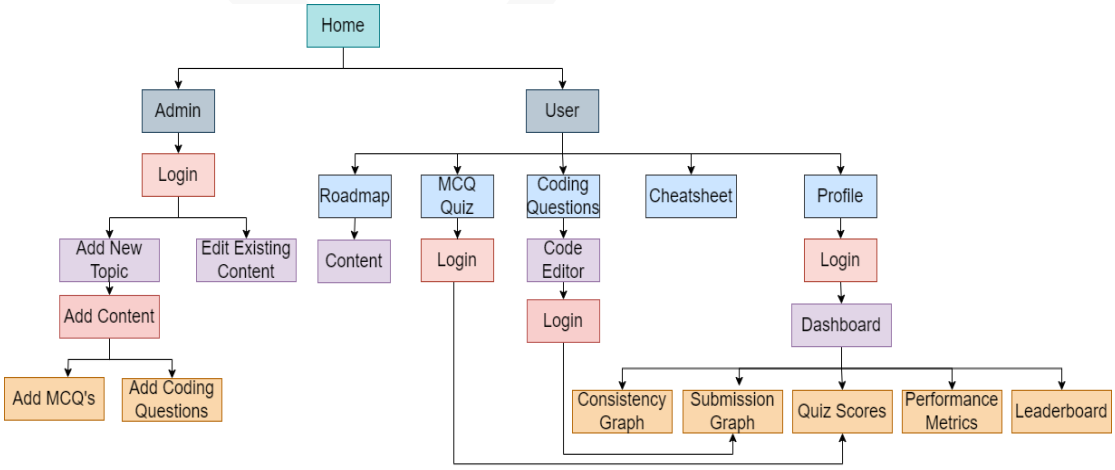

Fig. 3.4 Flow Chart for Placement Preparation Module Focusing on DSA

3.6 System Design

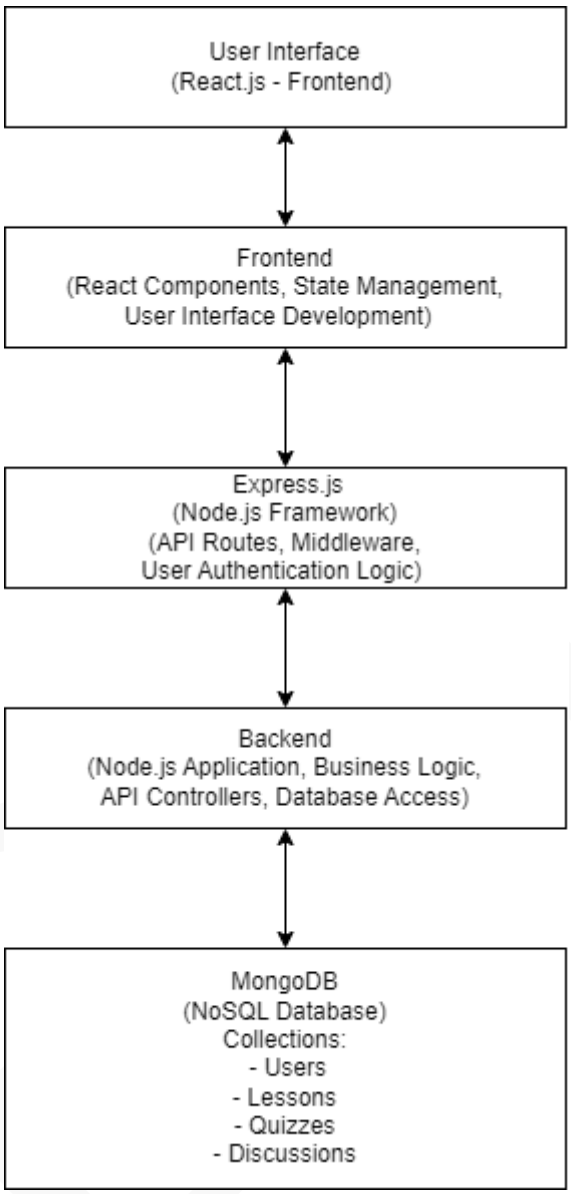

Fig. 3.5 System Architecture for Placement Preparation Module Focusing on DSA

3.7 Software Requirement

3.7.1 VS Code

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust, and Julia. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code

employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called "Visual Studio Online" and "Visual Studio Team Services").

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Visual Studio Code can be extended via extensions available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where users can access version control settings and view changes made to the current project. To use the feature, Visual Studio Code must be linked to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows users to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

3.7.2 Postman

Postman is one of the most popular software testing tools which is used for API testing. With the help of this tool, developers can easily create, test, share, and document APIs. Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API. In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET,

POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

Postman is based on a wide range of extremely user-friendly power tools. For more than 8 million users, Postman has become a tool of convenience. Following are the reasons why Postman is used:

- Accessibility- One can use it anywhere after installing Postman into the device by simply logging in to the account.
- Use Collections-Postman allows users to build collections for their API-calls. Every set can create multiple requests and subfolders. It will help to organize the test suites.
- Test development- To test checkpoints, verification of successful HTTP response status shall be added to every API- calls.
- Automation Testing-Tests can be performed in several repetitions or iterations by using the Collection Runner or Newman, which saves time for repeated tests.
- Creating Environments- The design of multiple environments results in less replication of tests as one can use the same collection but for a different setting.
- Debugging- To effectively debug the tests, the postman console helps to track what data is being retrieved.
- Collaboration- You can import or export collections and environments to enhance the sharing of files. You may also use a direct connection to share the collections.
- Continuous integration-It can support continuous integration.

3.7.3 Web Browser (Chrome, Firefox, Internet Explorer, Safari)

A web browser is an application for accessing websites and the Internet.[1] When a user requests a web page from a particular website, the browser retrieves its files from a web server and then displays the page on the user's screen. Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. Versions were later released for Linux, macOS, iOS, and also for Android, where it is the default browser. The browser is also the main component of ChromeOS, where it serves as the platform for web applications.

3.7.4 APIs

ChatGPT

We have used this API for chatbot feature in our project. OpenAI's text generation models (often referred to as generative pre-trained transformers or "GPT" models for short), like GPT-4 and GPT-3.5, have been trained to understand natural and formal language. Models like GPT-4 allows text outputs in response to their inputs. Assistants refer to entities, which in the case of the OpenAI API are powered by large language models like GPT-4, that are capable of performing tasks for users. These assistants operate based on the instructions embedded within the context window of the model.

Judge0

Judge0 is a robust, scalable, and open-source online code execution system. You can use it to build a wide range of applications that need online code execution features.

- Rich and verbose API documentation
- Scalable architecture
- Sandboxed compilation and execution
- Support for 60+ languages
- Compilation and execution of multi-file programs
- Support for additional files alongside the user's program
- Support for custom user-defined compiler options, command-line arguments, and time and memory limits
- Detailed execution results
- Webhooks (HTTP callbacks)

3.8 Functional Requirements

Functional requirements for an air quality monitoring system using IoT include the following:

3.9.1. User Authentication

- Users can create accounts with unique usernames and passwords.

- The system securely stores user credentials.
- Users can log in using their credentials to access personalized features.

3.9.1. Learning Resources

- The platform provides structured lessons on various Data Structures and Algorithms (DSA) concepts.
- Learning resources include textual content, visual aids, and examples.
- Questions are categorized based on difficulty levels (easy, medium, hard) and topics (arrays, linked lists, sorting algorithms, etc.).

3.9.1. Quizzes and Practice

- Users can access quizzes covering different DSA topics and difficulty levels.
- Quizzes are interactive with multiple-choice questions and coding challenges.
- A coding practice section allows users to solve algorithmic problems.

3.9.1. Discussion Forum

- The platform features a Q&A section where users can ask and answer questions.
- Discussion boards are organized around DSA topics for community engagement.

3.9.1. Roadmap

- A personalized learning roadmap is designed for each user.
- The roadmap tracks completed lessons, quiz scores, and coding practice achievements.

3.9.1. Content Access

- Users can browse and view website content, including lessons and non-interactive resources without logging in.
- Login is required for interactive activities such as quizzes, coding practice, and discussions.

3.9.1. Cheatsheet

- Cheatsheets are provided for summarizing key time and space complexities for common algorithms and data structures.

3.9.1. Profile Management

- Users can create, edit, and manage their profiles.
- Profile information includes personal details, academic history, and areas of interest.

3.9.1. Visual Progress Tracking

- Visual representations, such as charts or graphs, depict users' performance and progress over time.
- Progress is visually displayed for completed lessons, quiz scores, and coding practice achievements.

3.9.1. Code Editor

- The platform integrates a code editor that supports multiple programming languages.
- Users can write, test, and submit code directly within the platform.

3.9.1. Accessibility

- The user interface is designed to be accessible and user-friendly for users with diverse needs.
- Navigation elements are intuitive, and the platform adheres to accessibility standards.

3.9.1. Feedback Mechanism

- Users can provide feedback on lessons, quizzes, and overall platform experience.
- A mechanism for reporting issues or suggesting improvements is implemented.

3.9.1. Responsive Design

- The platform is responsive, ensuring optimal user experience across various devices and screen sizes.

3.9.1. Security Measures

- Secure protocols, such as encryption, are implemented to protect user data and privacy.
- Measures are in place to prevent unauthorized access and ensure data integrity.

3.9.1. Chatbot

- The platform includes a chatbot feature to assist users with queries related to DSA concepts, quizzes, and platform navigation.
- The chatbot is accessible through a designated interface, providing users with a conversational way to seek information.

3.9 Non-Functional Requirements

3.10.1. Performance

- Response Time: The system should respond to user interactions within 2 seconds for all major functionalities.
- Scalability: The platform should handle a simultaneous user load of at least 1000 users without significant degradation in performance.

3.10.1. Reliability

- The system should have an uptime of at least 99.5% for continuous availability.
- In the event of a server failure, the system should recover within minutes.

3.10.1. Availability:

- The platform should be accessible 24/7, with scheduled maintenance communicated in advance.

3.10.1. Scalability

- The system should scale horizontally to accommodate an increasing number of users and data.

3.10.1. Security

- User data, including personal information and performance metrics, should be encrypted during transmission and storage.
- The platform should implement secure user authentication and authorization mechanisms.
- Regular security audits and vulnerability assessments should be conducted.

3.10.1. Usability

- The user interface should be intuitive and require minimal training for new users.

3.10.1. Maintainability

- Codebase should be well-documented for ease of maintenance and future enhancements.
- System updates and patches should be deployable without significant downtime.

3.10.1. Compatibility

- The platform should be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.
- Mobile responsiveness should be maintained across various devices and operating systems.

3.10.1. Portability

- The system should be deployable on cloud platforms like AWS, Azure, or Google Cloud.
- Code should be modular and portable for easy migration between different environments.

3.10.1. Interoperability

- The platform should integrate seamlessly with external services such as email and SMS for notifications.
- APIs should be available for potential future integrations with other educational platforms.

3.10.1. Capacity

- Database capacity should accommodate a growing volume of lessons, quizzes, and user data.

3.10 UML Diagram

3.10.2 Use Case Diagram

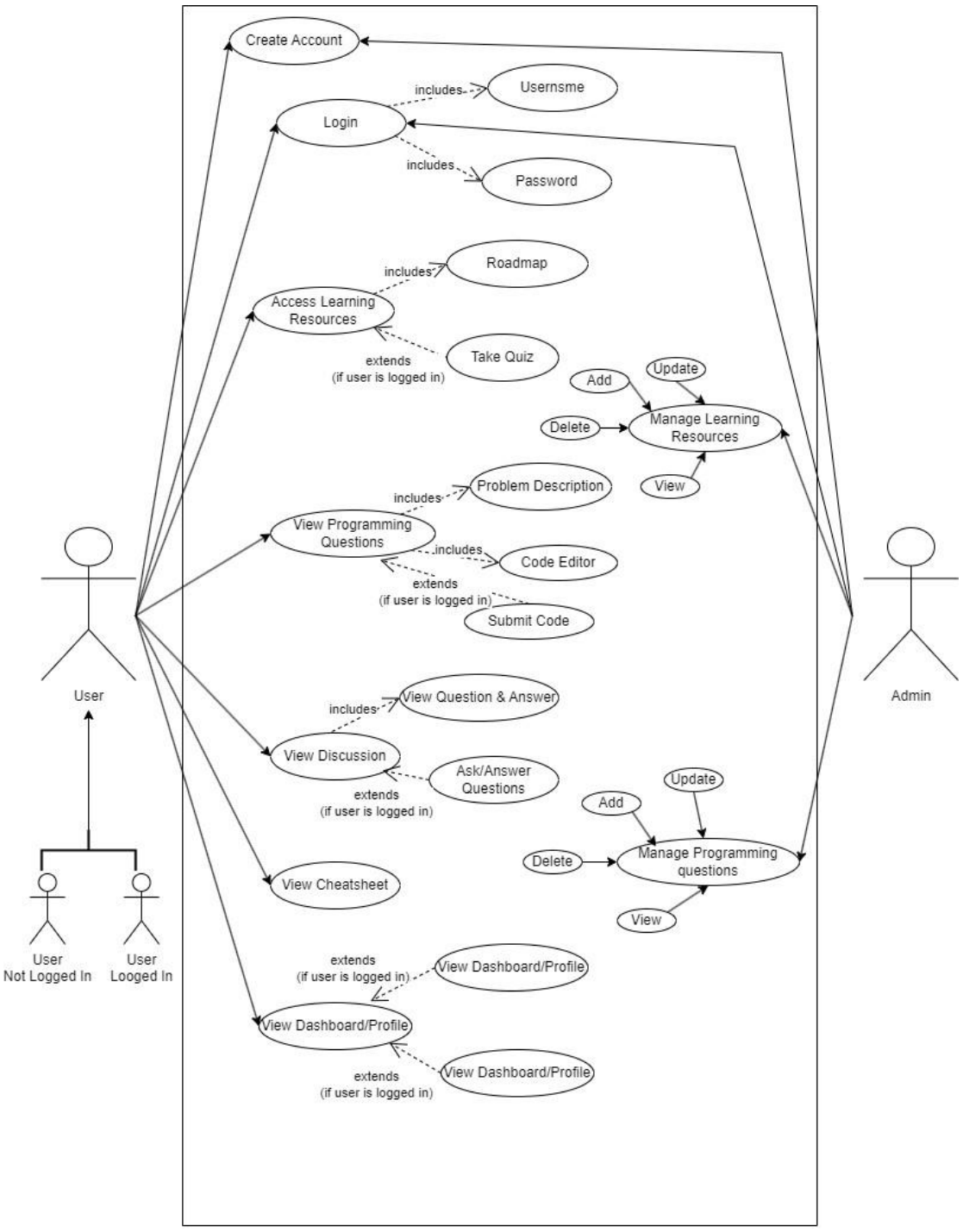

Fig. 3.6 Use Case Diagram for Placement Preparation Module Focusing on DSA

3.10.1 Sequence Diagram

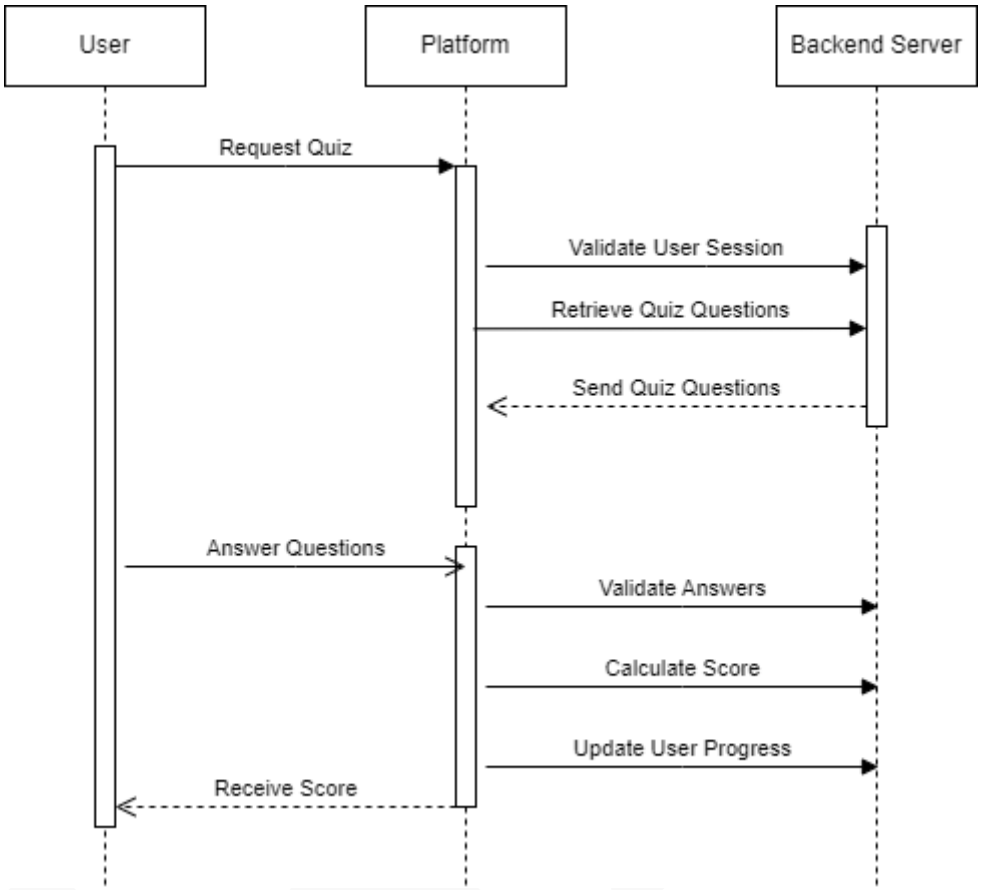

Fig. 3.7 Sequence Diagram for Placement Preparation Module Focusing on DSA.

CHAPTER 4

IMPLEMENTATION OF PROJECT

This chapter contains implementation of project. In this project we have implemented a Placement Preparation Module for DSA. web application engineered on the MERN stack—MongoDB, Express.js, React.js, and Node.js. We set up the development environment with Node.js, MongoDB, and a code editor. Initialize a React.js project for the frontend and a Node.js/Express.js project for the backend.

Implement user registration and login functionalities. Created React components for displaying lessons on DSA concepts. Develop interactive quizzes with React components, handling real-time user responses. Implemented a coding practice section with a custom code editor supporting multiple programming languages using API Judge0 CE that has logic to execute and evaluate user-submitted code. Created API endpoints for posting and fetching discussion threads and Implemented React components for rendering discussion threads and facilitating user interactions.

Designed React components to visualize the user's learning roadmap. Implemented visual progress tracking using charts or graphs based on user performance metrics. Integrate a chatbot using ChatGPT API. We made responsive design using CSS frameworks and tested the platform on various devices to ensure compatibility.

Designed the MongoDB database schema to accommodate user data, lessons, quizzes, and study material. Established relationships between collections for efficient data retrieval. Conducted unit testing for individual components and performed end-to-end testing to ensure the seamless integration of features.

4.1 App.js

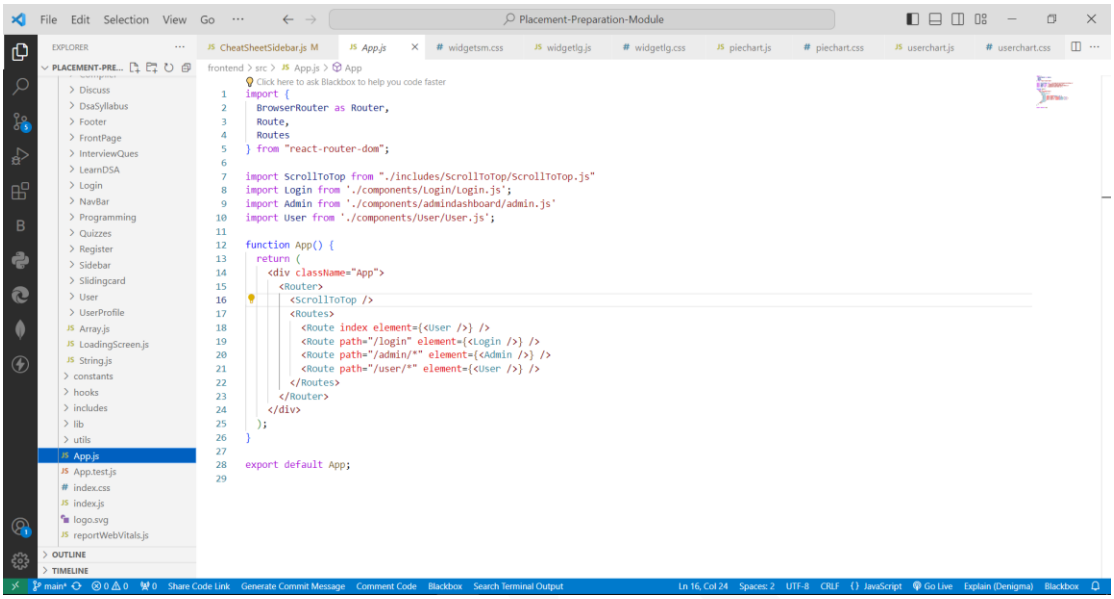

Fig. 4.1 App.js file of Placement Preparation Module Focusing on DSA.

4.2 User.js

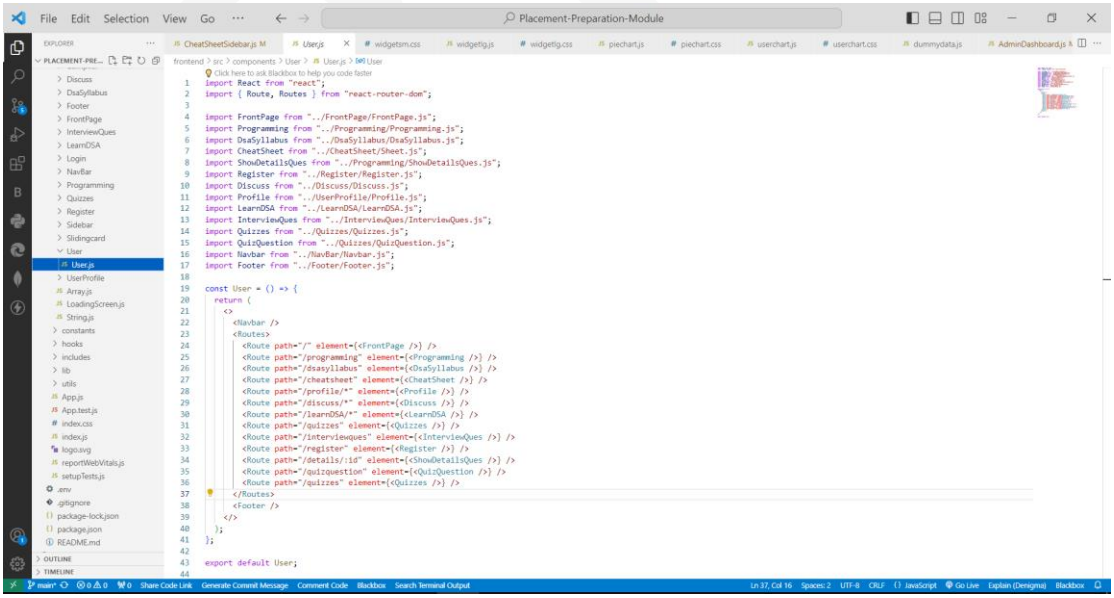

Fig. 4.2 User.js file of Placement Preparation Module Focusing on DSA.

4.3 Admin Dashboard

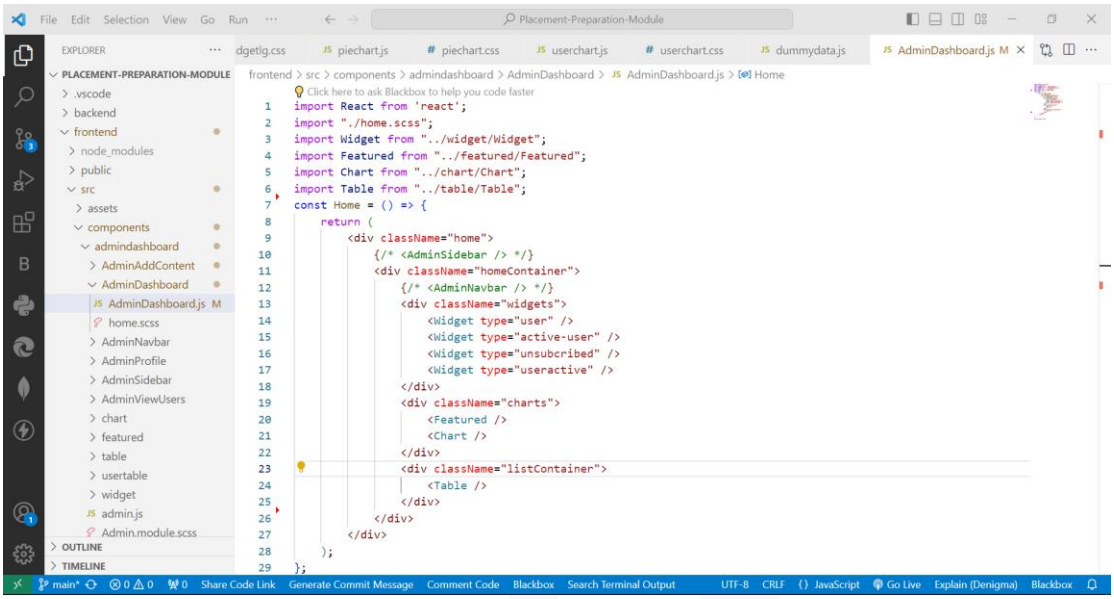

Fig. 4.3 AdminDashboard.js file of Placement Preparation Module Focusing on DSA.

4.4 Chatbot

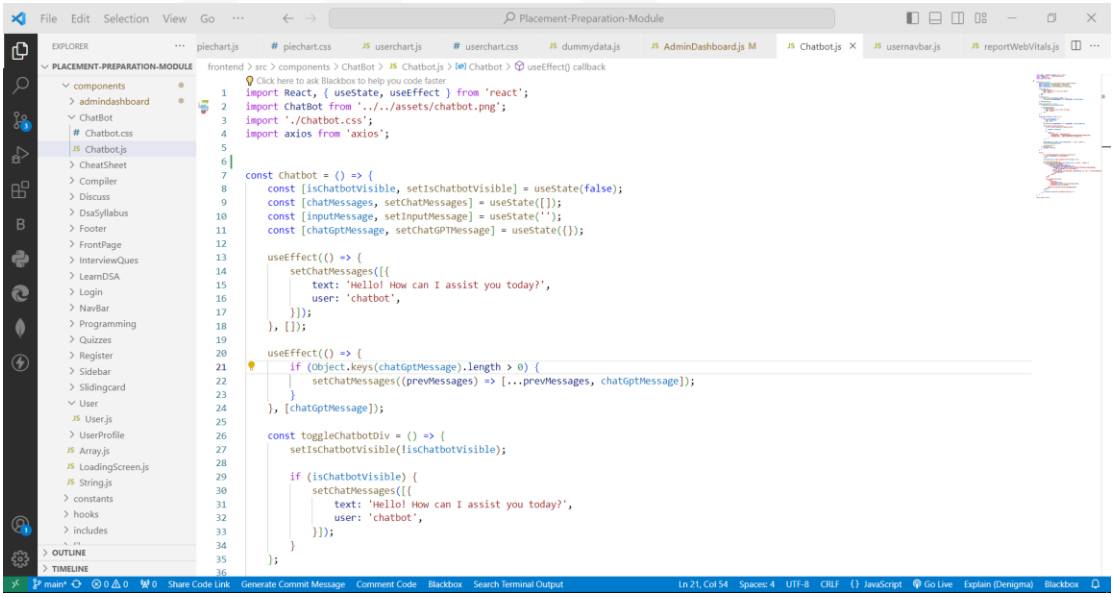

Fig. 4.4 Chatbot.js file of Placement Preparation Module Focusing on DSA.

4.5 Cheatsheet

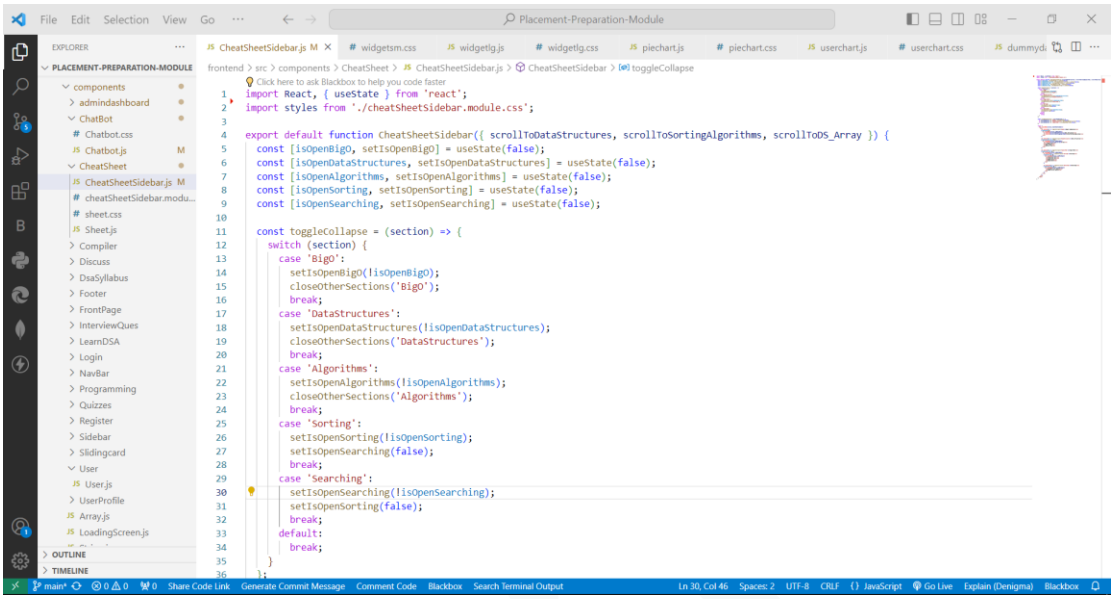

Fig. 4.5 Cheatsheet.js file of Placement Preparation Module Focusing on DSA.

4.6 Discussion Forum

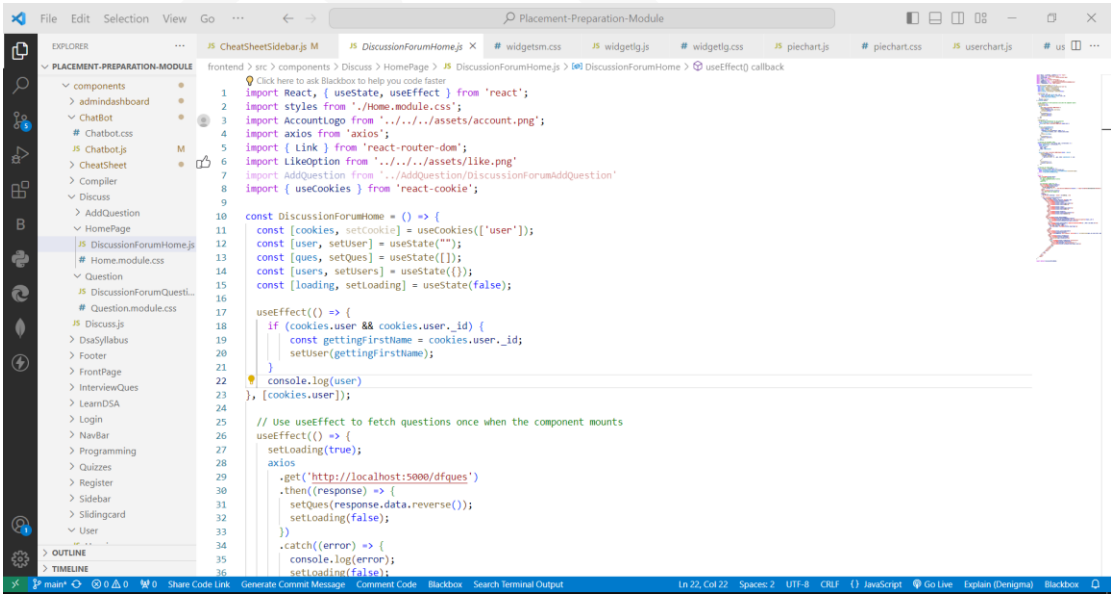

Fig. 4.6 DiscussionForum.js file of Placement Preparation Module Focusing on DSA.

4.7 Learn DSA

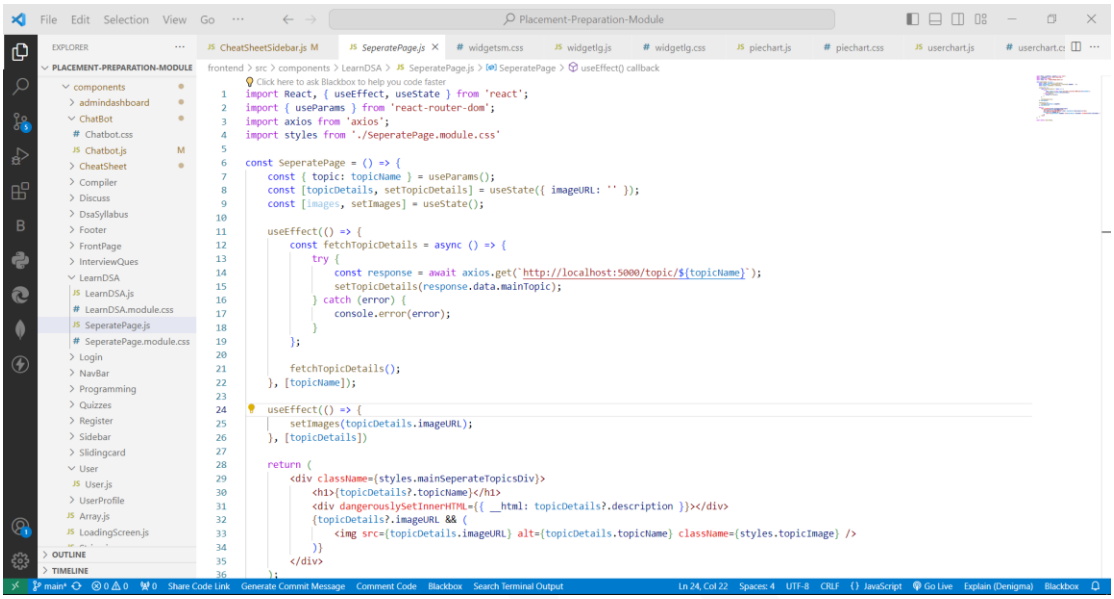

Fig. 4.7 SeparatePage.js file of Placement Preparation Module Focusing on DSA.

4.8 Login

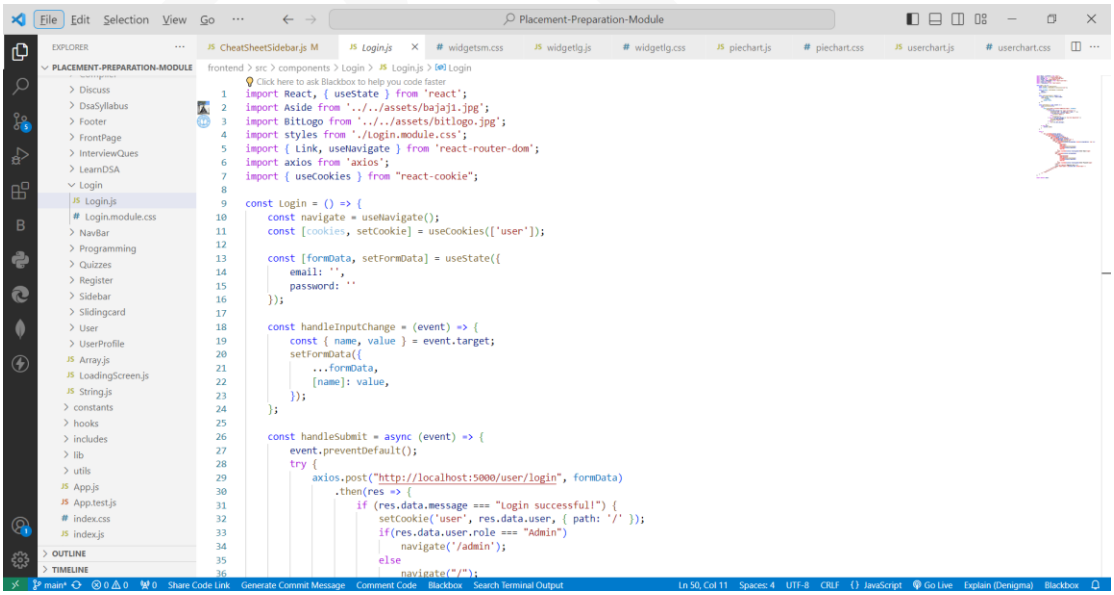

Fig. 4.8 Login.js file of Placement Preparation Module Focusing on DSA.

4.9 Programming

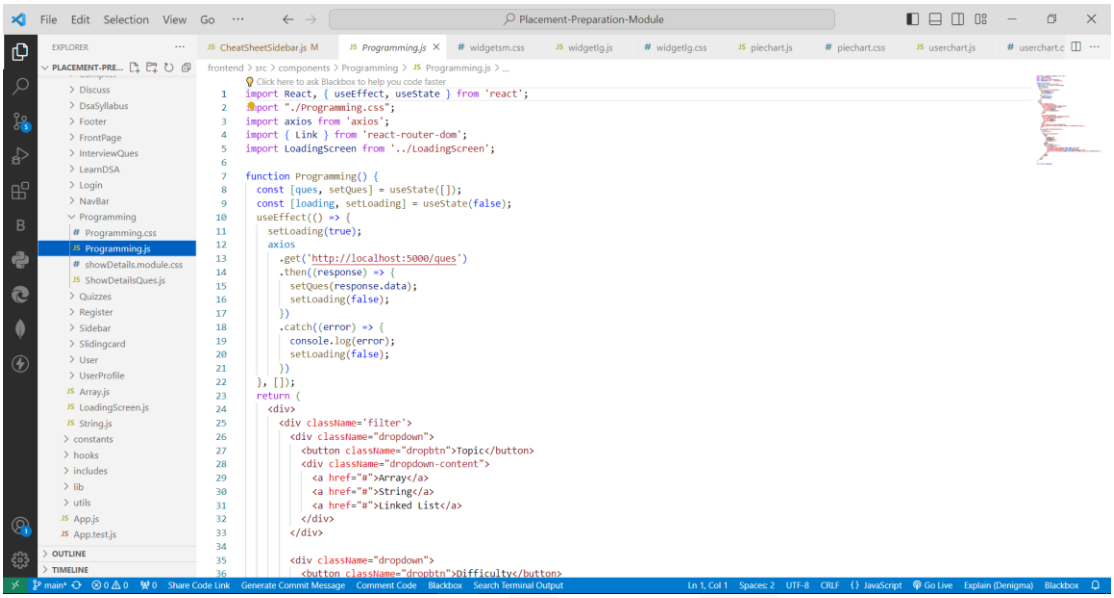

Fig. 4.9 Programming.js file of Placement Preparation Module Focusing on DSA.

4.10 Quiz Questions

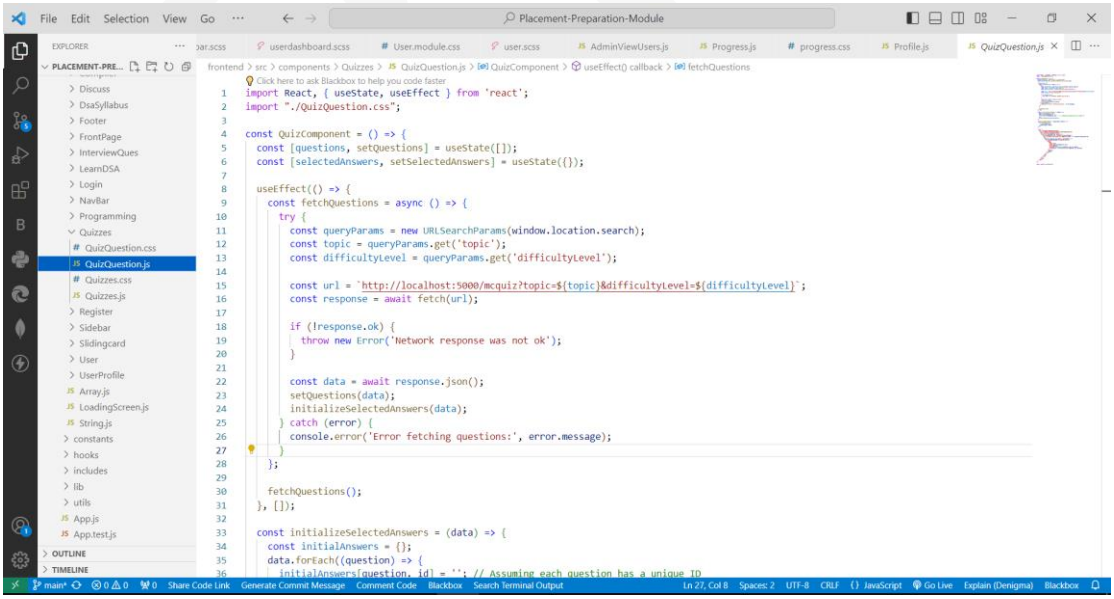

Fig. 4.10 QuizQuestions.js file of Placement Preparation Module Focusing on DSA.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Home Page

The homepage interface enables user to navigate through the concept and purpose of learning DSA and learn about features of our website. There is also contact us sections with option of newsletters.

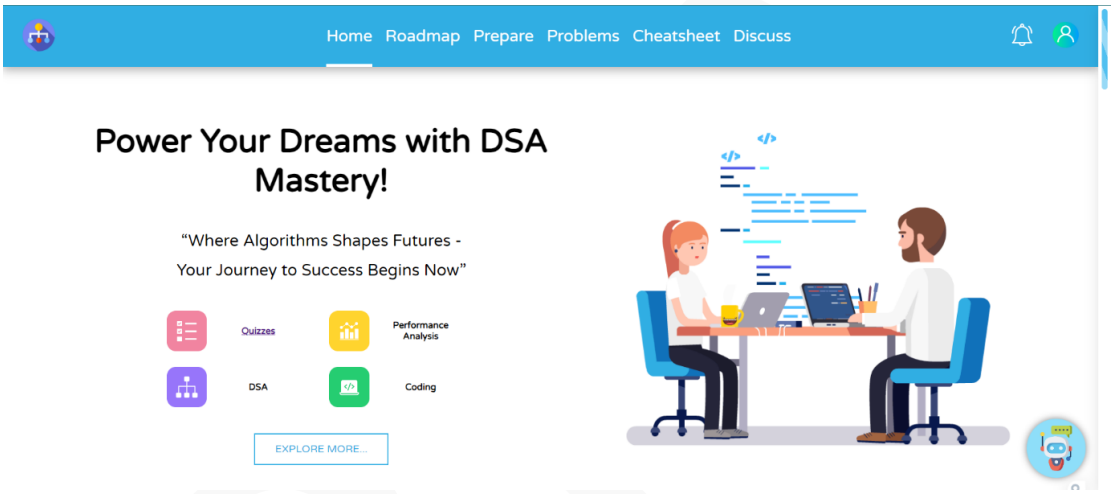

Fig. 5.1 Home Page of Placement Preparation Module Focusing on DSA.

5.1.1 Chatbot

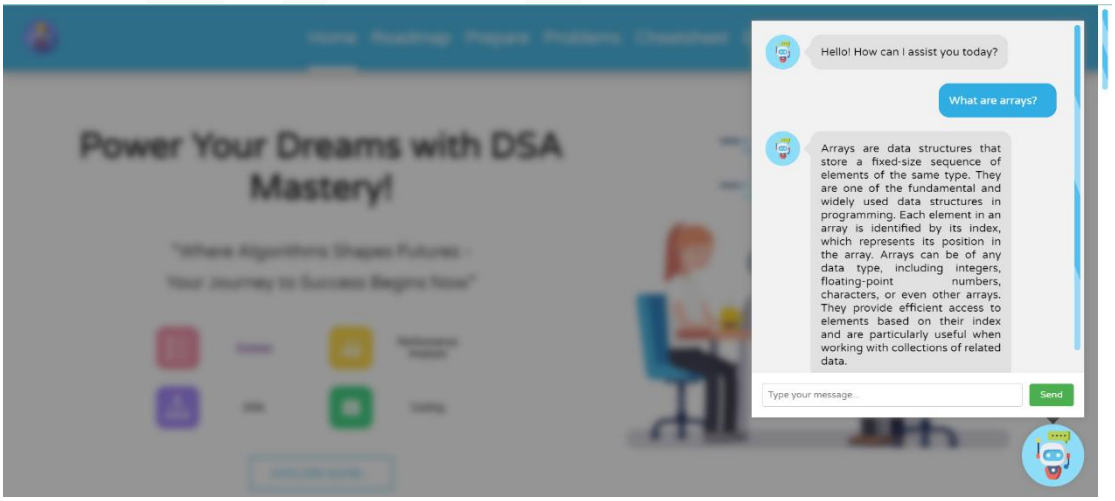

Fig. 5.2 Chatbot of Placement Preparation Module Focusing on DSA.

Fig. 5.3 Home Page of Placement Preparation Module Focusing on DSA.

5.2 Roadmap Page

The homepage interface enables user to navigate through the concept and purpose of learning DSA and learn about features of our website. There is also contact us sections with option of newsletters.

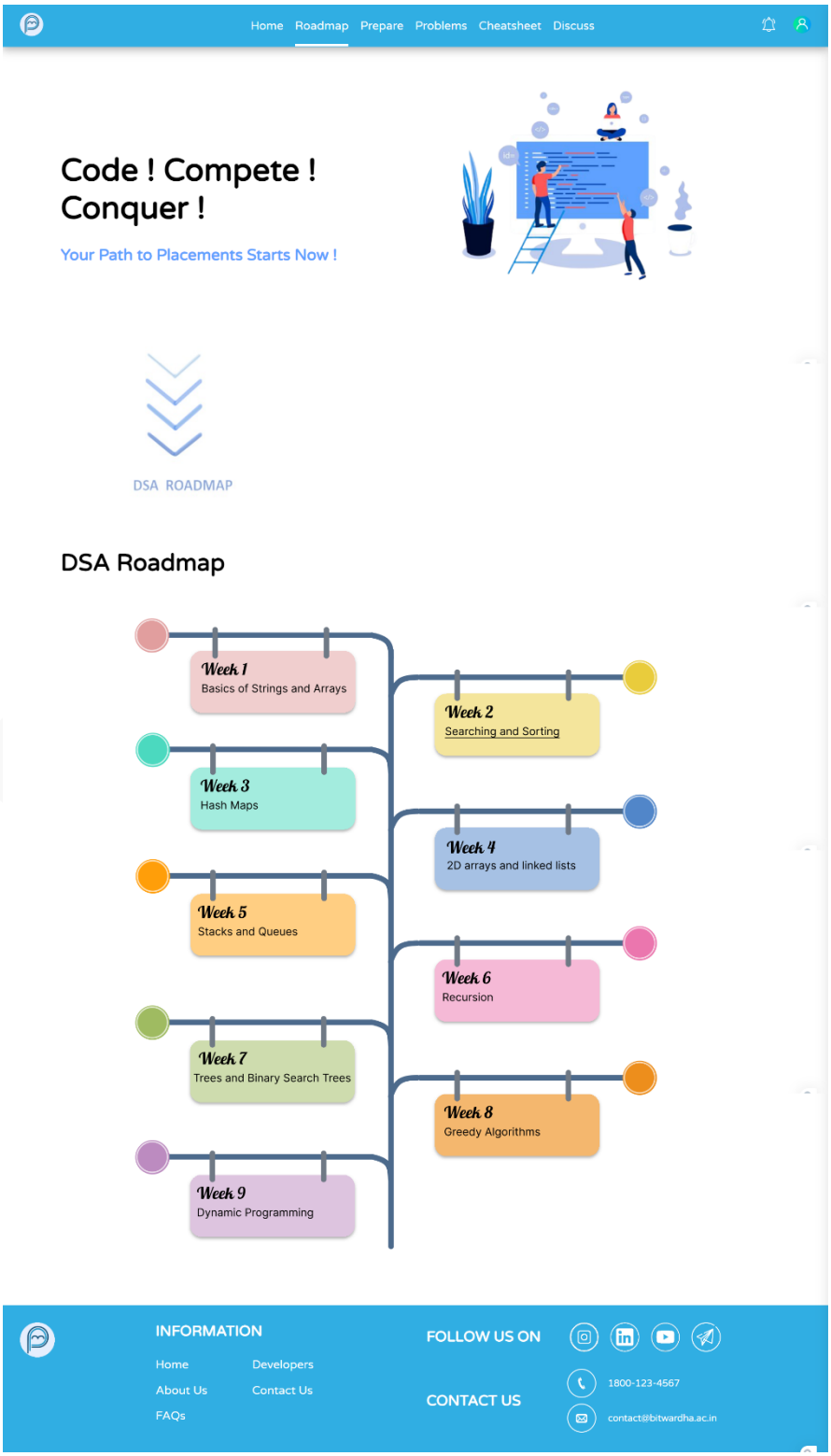

Fig. 5.4 Roadmap Page of Placement Preparation Module Focusing on DSA.

5.3 Prepare Page

There is a side navigation bar which contains topics of DSA and the page contains roadmap which displays users learning path. It contains topic wise short contents with short and precise information. Also interview questions are provided.

5.3.2 Quizzes

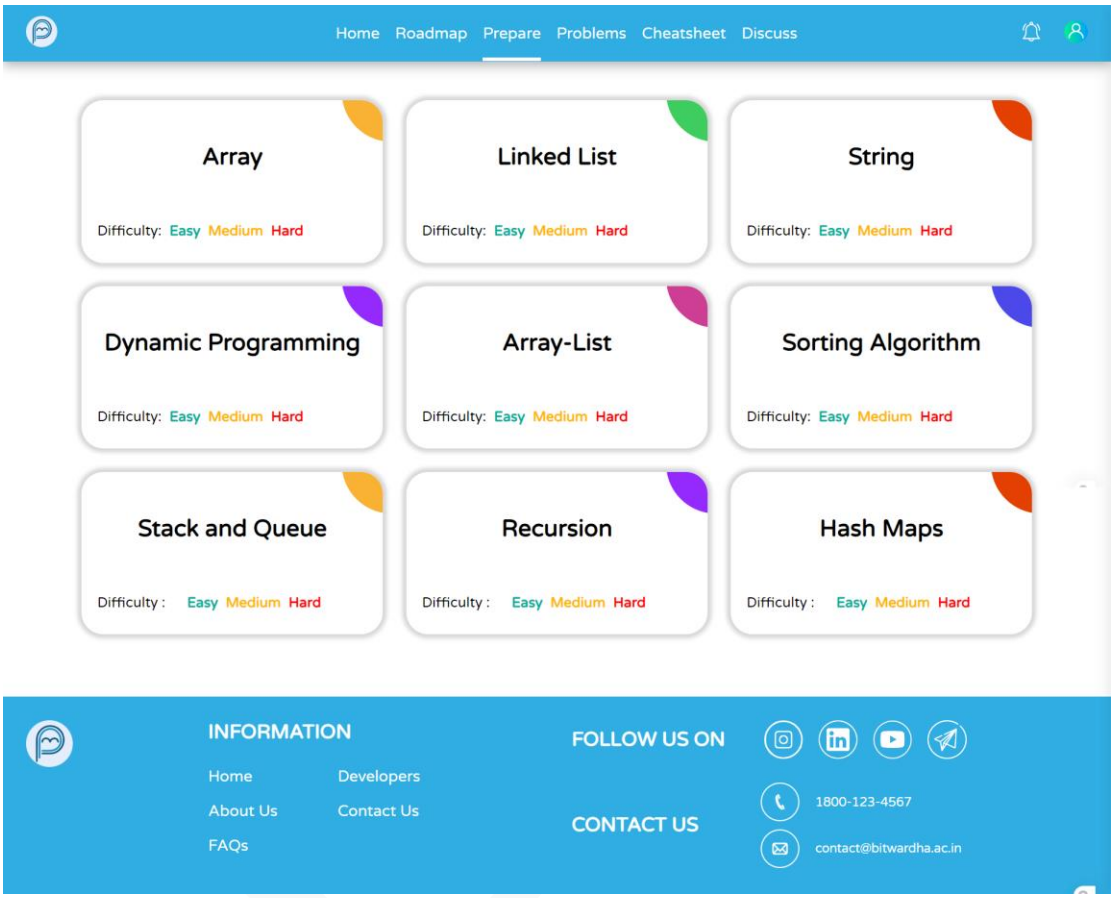

Fig. 5.5 Quizzes Page of Placement Preparation Module Focusing on DSA.

Fig. 5.6 Quizzes Form of Placement Preparation Module Focusing on DSA.

5.3.1 Learn DSA

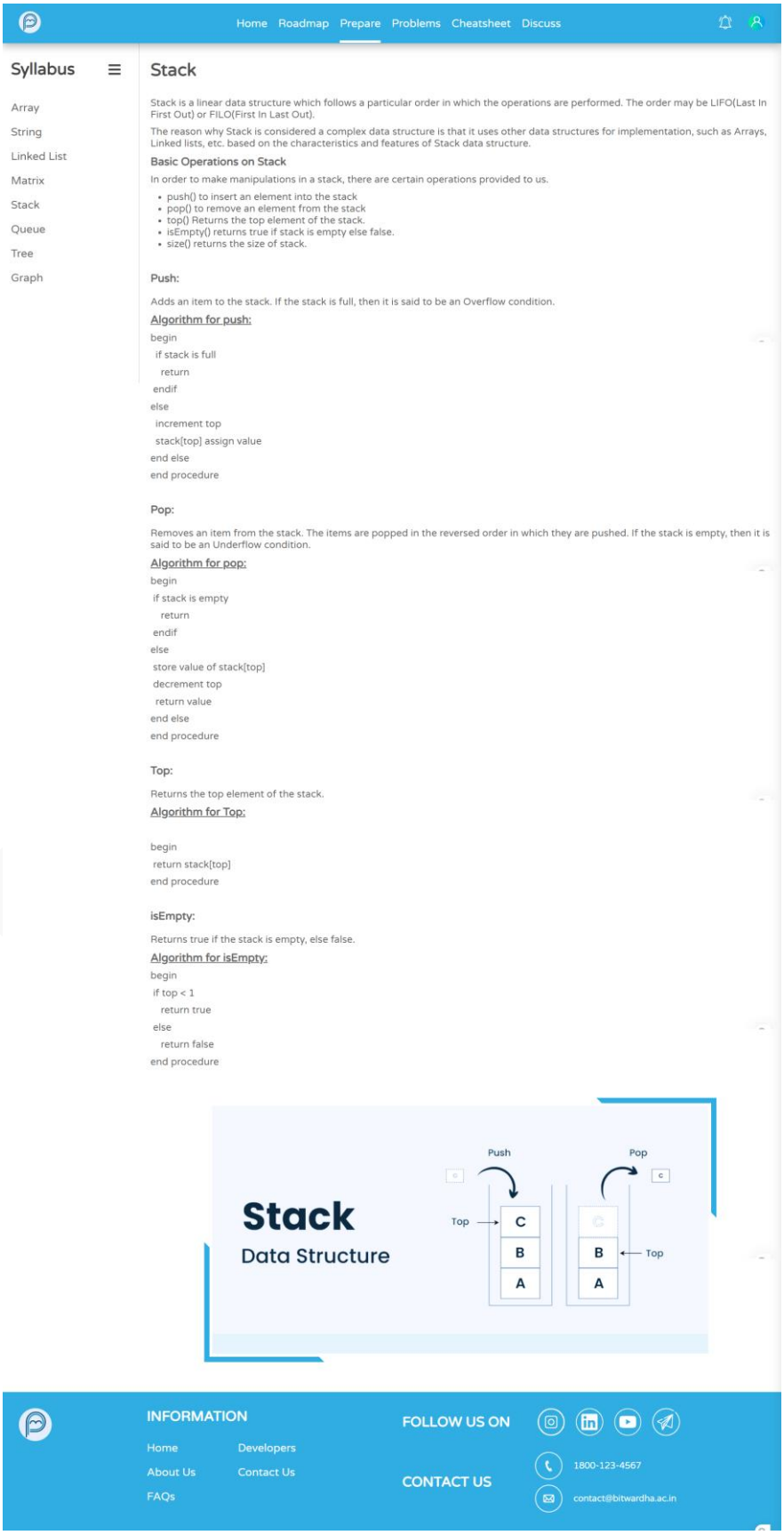

Fig. 5.7 DSA Content Page of Placement Preparation Module Focusing on DSA.

5.3.3 Interview Questions

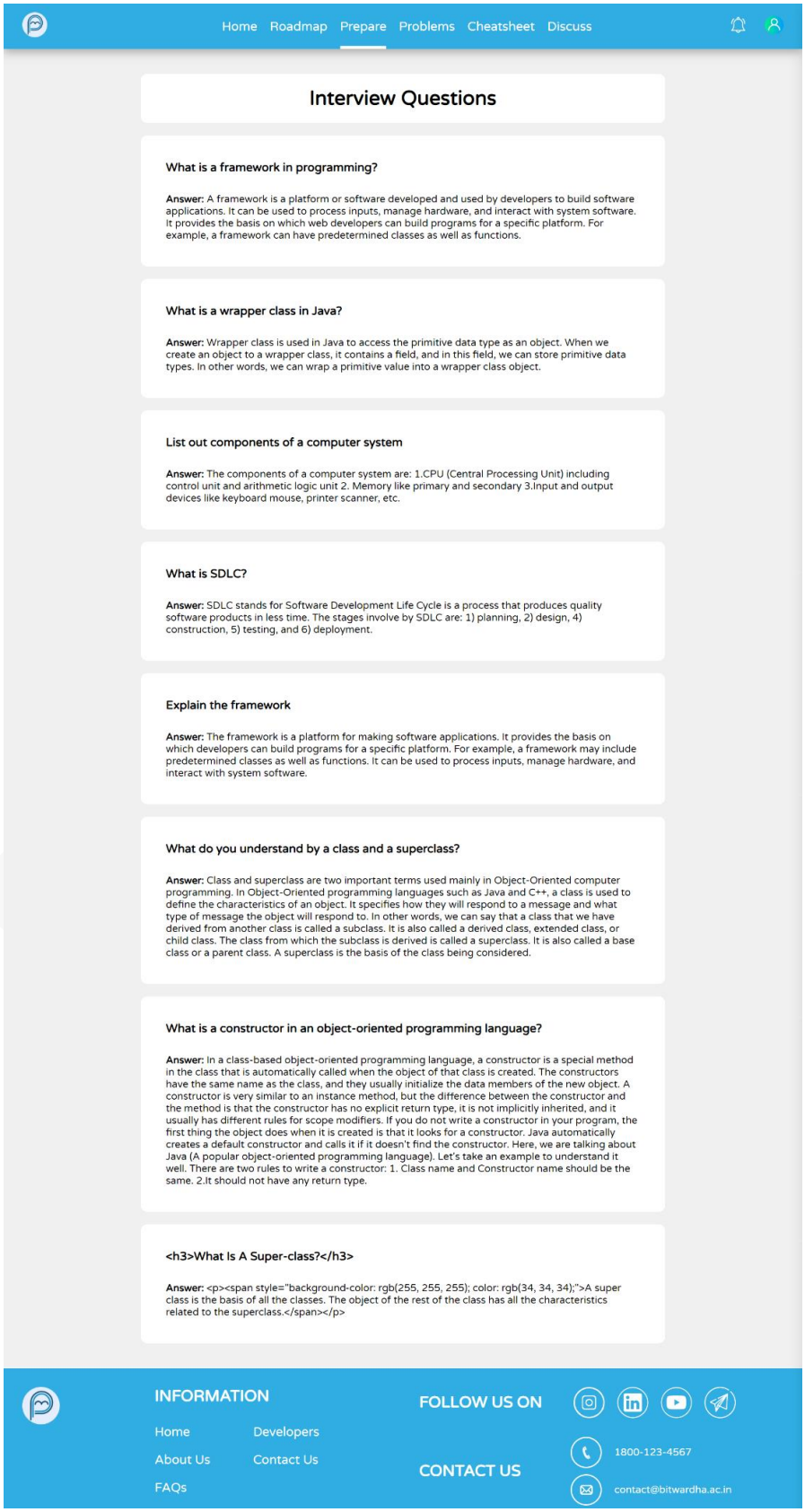

Fig. 5.8 Interview Questions Page of Placement Preparation Module for DSA.

5.4 Problems Page

This interface contains all the coding questions will be displayed d based on topic wise and difficulty level wise. The coding interface will have Problem Description on left half side and code editor will be on right side.

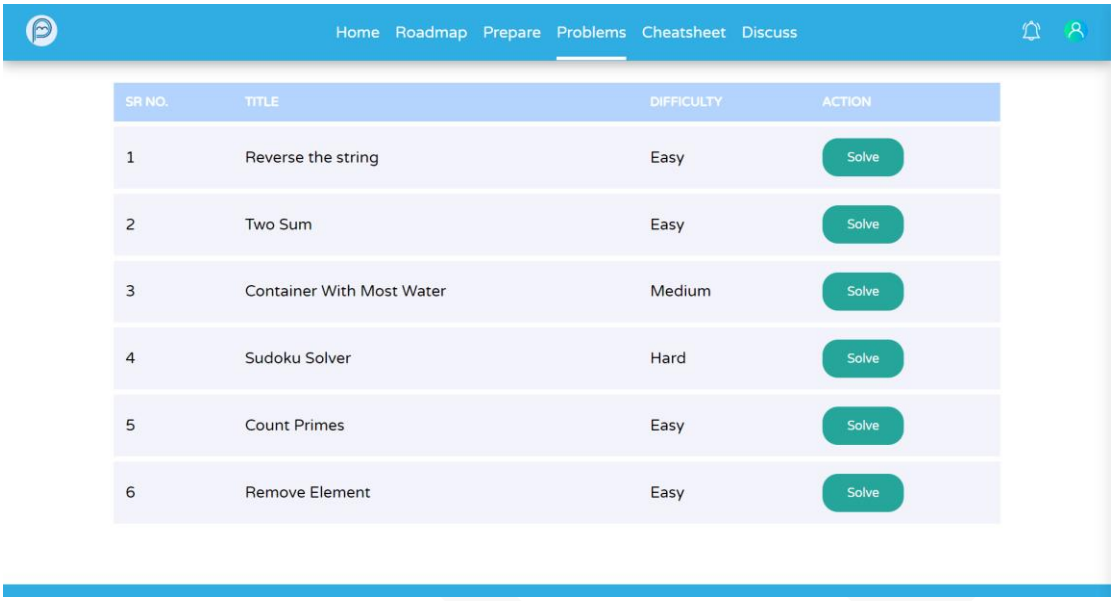

Fig. 5.9 Problems Page of Placement Preparation Module for DSA.

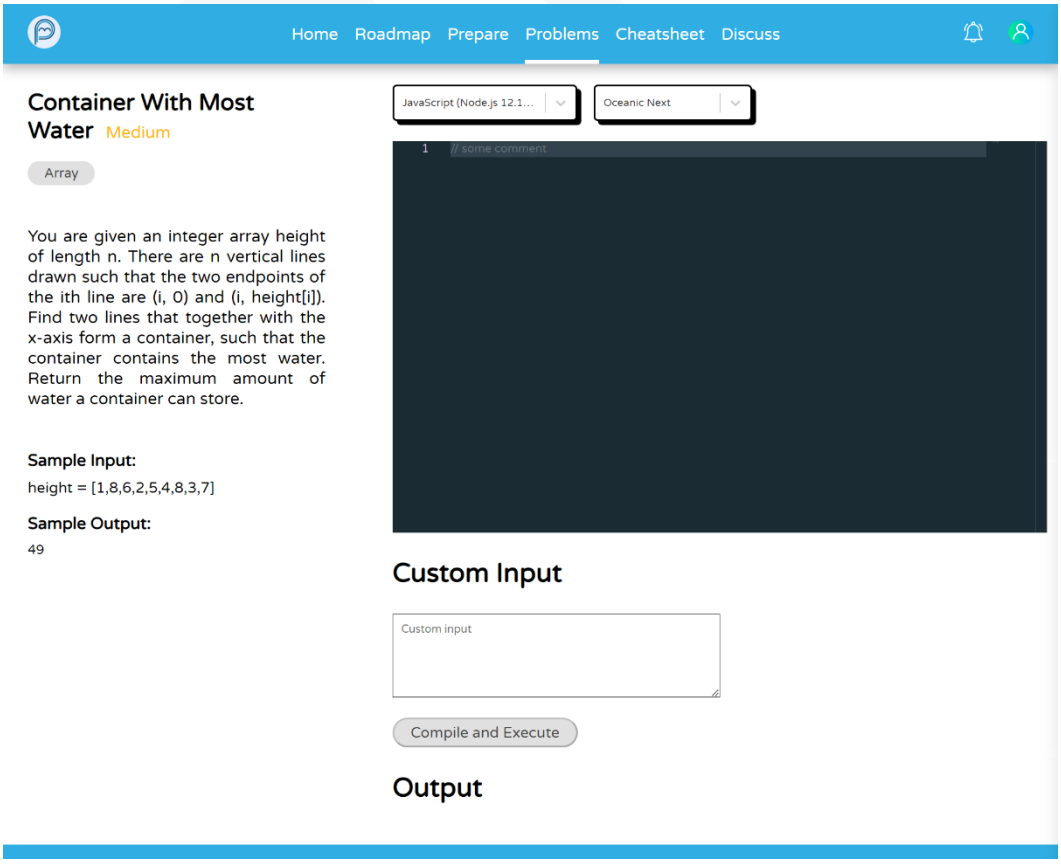

Fig. 5.10 Code Editor of Placement Preparation Module for DSA.

5.5 Cheatsheet Page

The interface contains the summary of important DSA contents such as time and space complexity of various data structures and algorithms and short revision material.

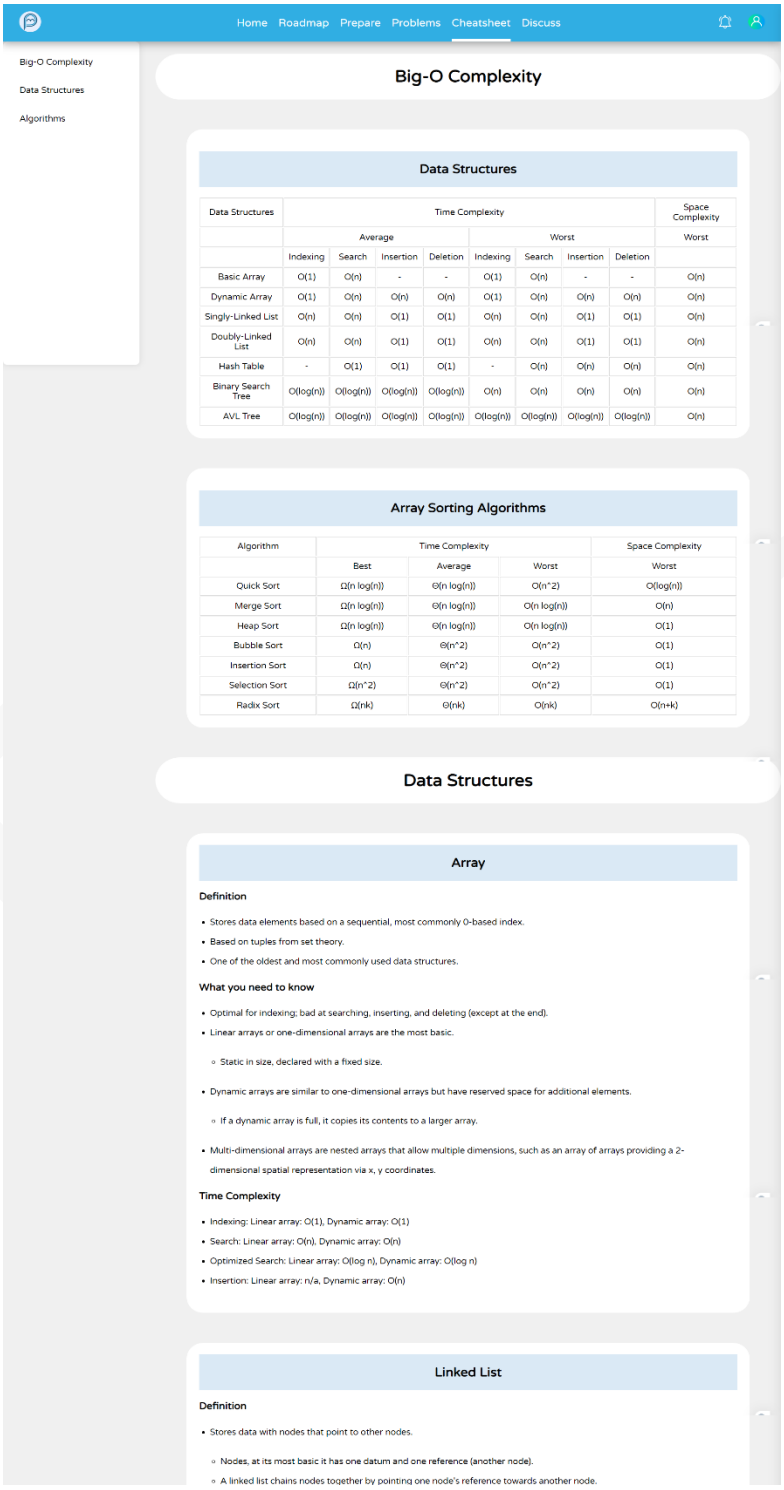

Fig. 5.11 Cheatsheet Page of Placement Preparation Module for DSA.

5.6 Discuss Page

This interface has question and answer section and user will add question through filling a form with question and submit it and also user can add answers to the questions.

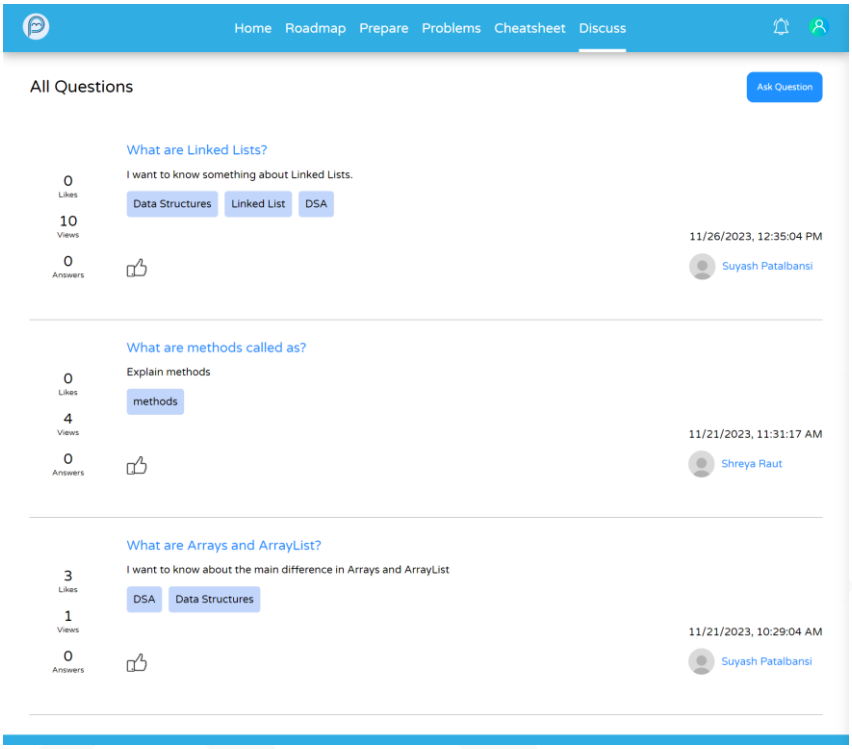

Fig. 5.12 Discussion Forum of Placement Preparation Module for DSA.

5.7 Login Page

The login interface enables you to integrate user login with the content of our website. Login as a user or admin.

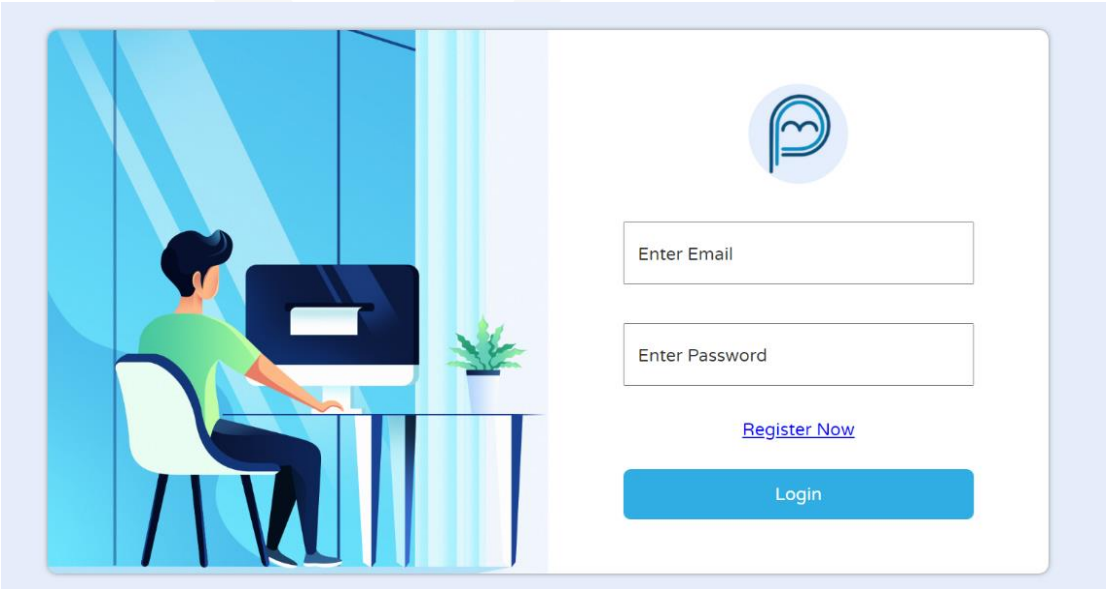

Fig. 5.13 Login Page of Placement Preparation Module for DSA.

5.8 User Dashboard

The homepage interface enables user to navigate through the concept and purpose of learning DSA and learn about features of our website. There is also contact us sections with option of newsletters.

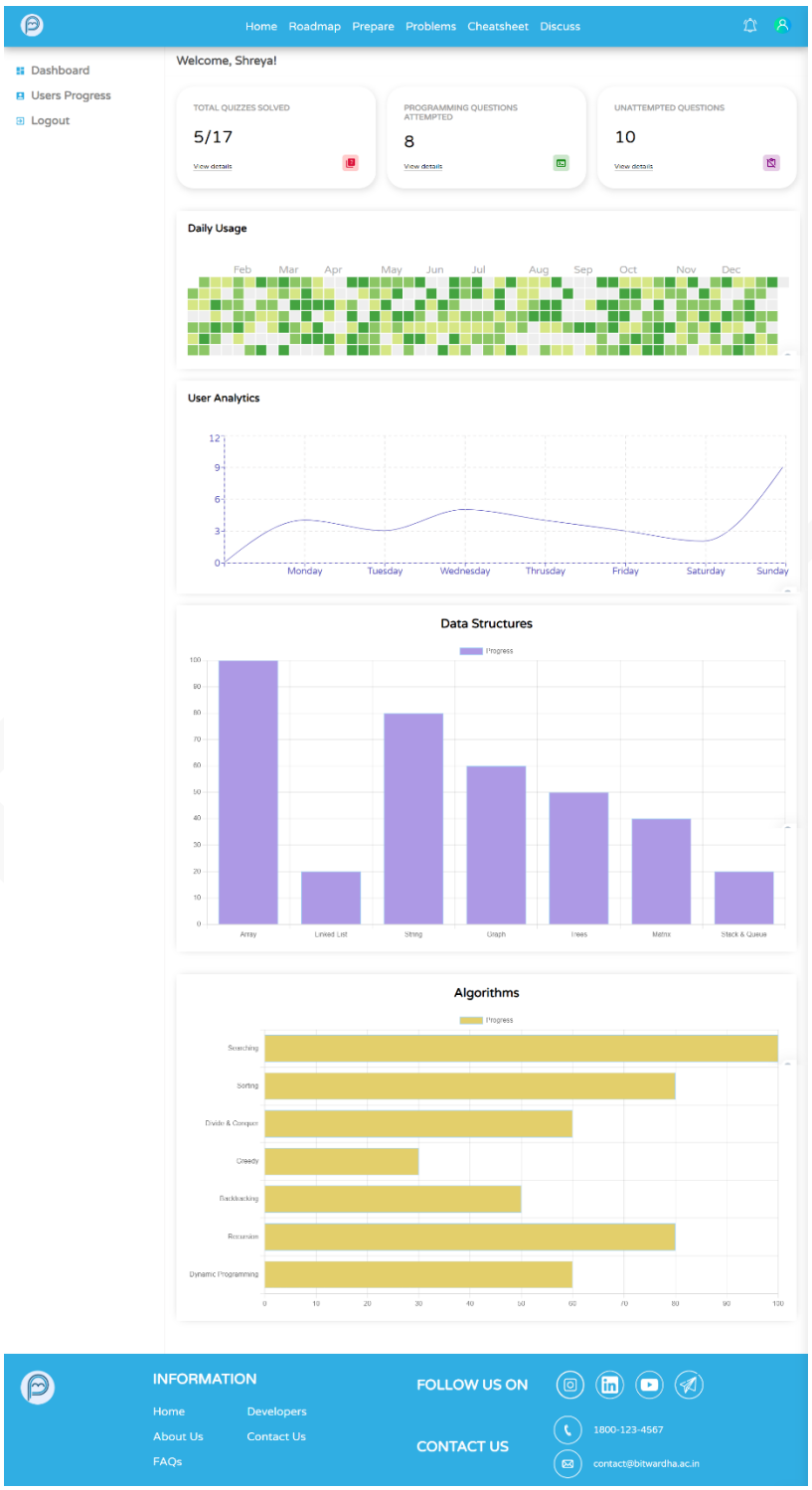

Fig. 5.14 User Dashboard of Placement Preparation Module for DSA.

5.9 Admin Dashboard

In this interface Admin will add new content or edit the previous content. The admin dashboard has user statistics and page statistics.

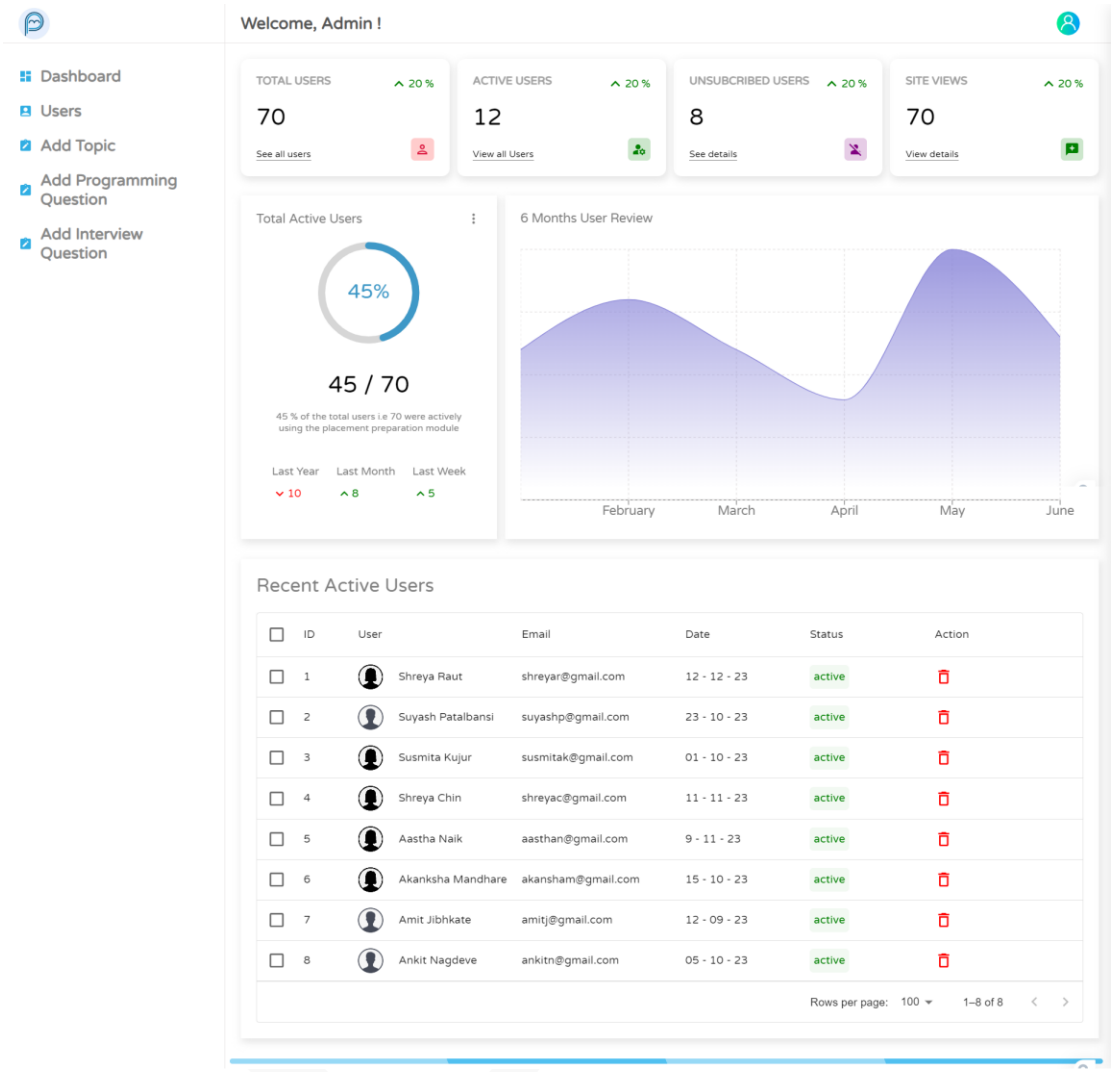

Fig. 5.15 Admin Dashboard of Placement Preparation Module for DSA.

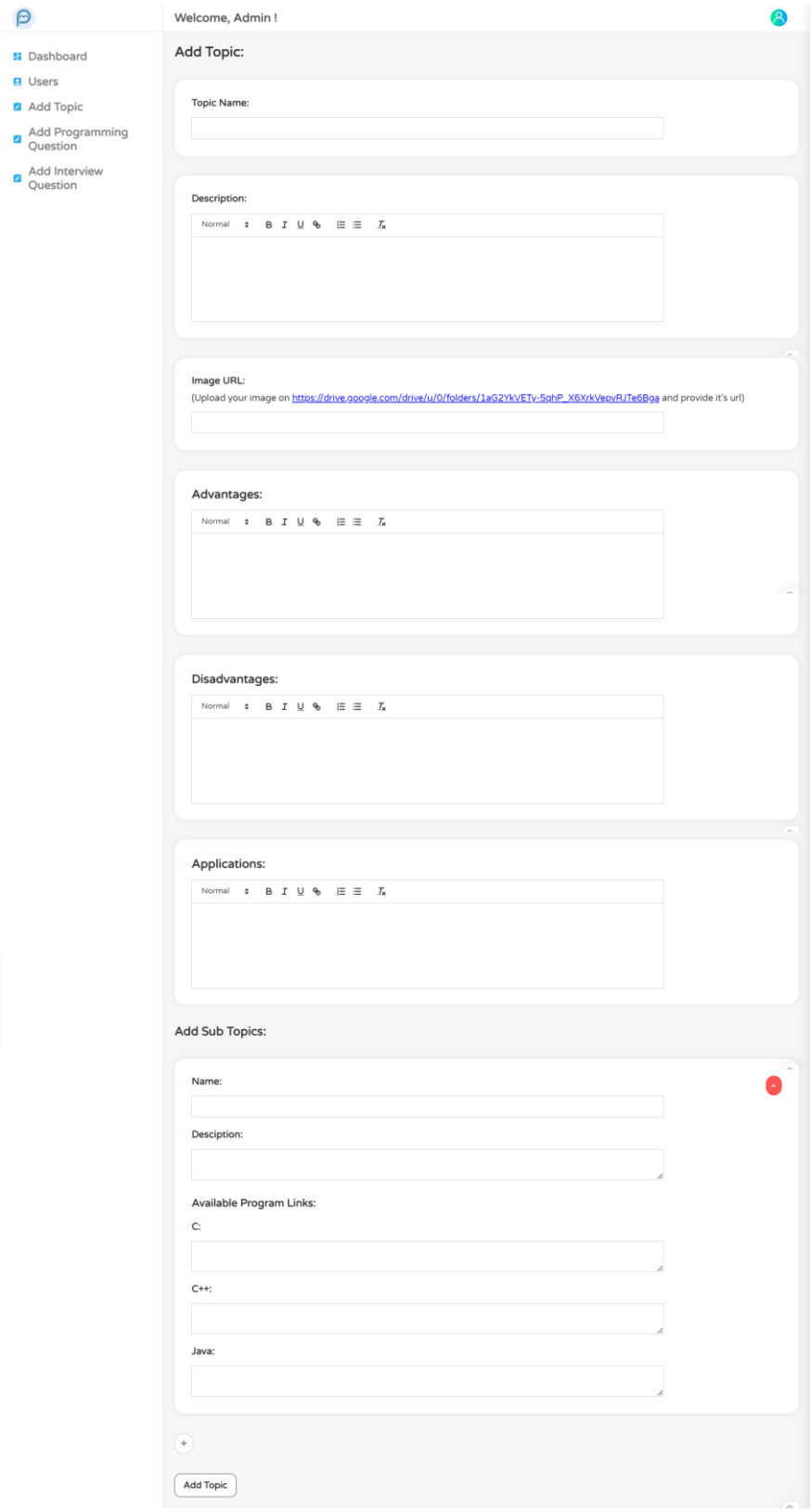

Fig. 5.16 Add Content in Admin Dashboard.

5.10 Backend Database

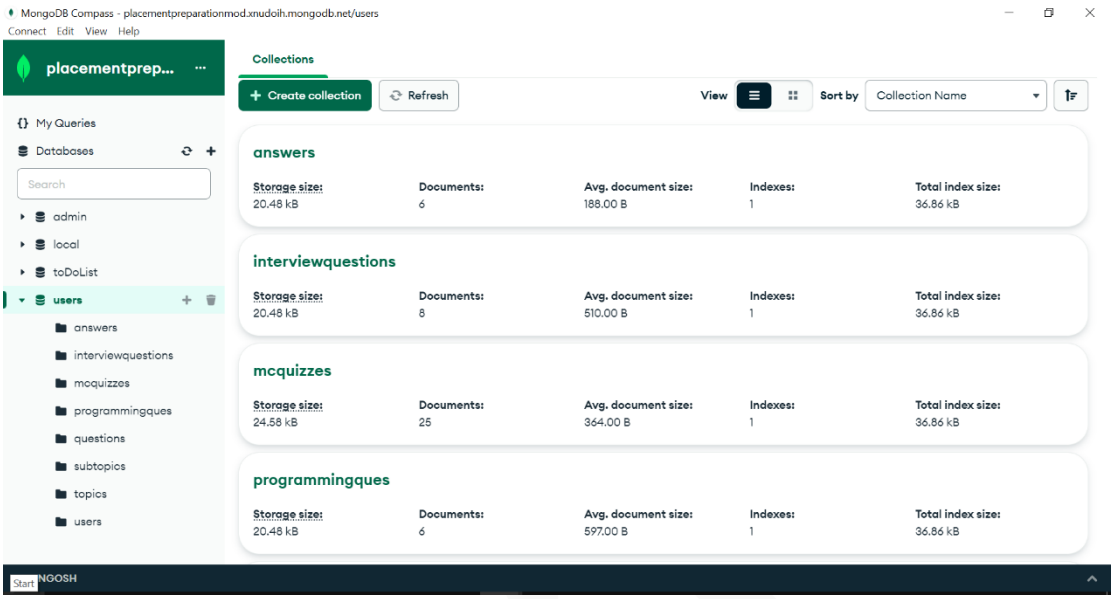

Fig. 5.17 MongoDB Database of Placement Preparation Module for DSA.

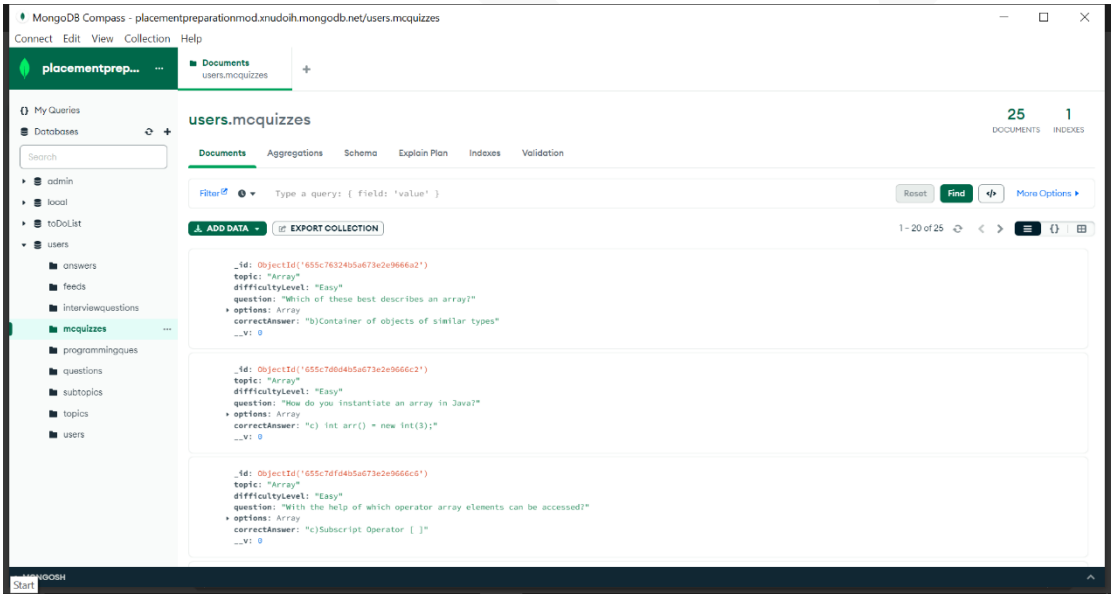

Fig. 5.18 MongoDB Database (quiz) of Placement Preparation Module for DSA.

CHAPTER 6
CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusion

In conclusion, the placement preparation DSA module is designed to provide a comprehensive and user-friendly platform for individuals aiming to excel in technical interviews and competitive coding assessments. The project objectives, outlined in the initial scope, were successfully translated into a functional and feature-rich application, aligning with the identified requirements. The architecture, based on the MERN stack, ensures a seamless flow of data and interactions between the frontend and backend components. The incorporation of a chatbot feature enhances user engagement, providing real-time assistance and fostering a more interactive learning experience.

The functional requirements address key aspects such as user authentication, learning resources, quizzes, discussions, progress tracking, and more. The chatbot, as a new addition, adds a layer of user support, allowing for natural language interactions and guidance. Non-functional requirements underscore the importance of performance, reliability, security, and usability. Ensuring scalability, maintainability, and compliance with industry standards further solidify the robustness of the platform.

The project's responsiveness to user needs, demonstrated by features like visual progress tracking, cheatsheets, and a user-driven Q&A system, aligns with the goal of creating a dynamic and adaptive learning environment. The platform's compatibility with various devices and its commitment to accessibility enhance inclusivity and user satisfaction. Moving forward, the project is poised for scalability and future enhancements. The establishment of a feedback mechanism encourages ongoing user engagement and ensures that the platform remains responsive to evolving requirements. The integration of a code editor, multi-language support, and external API compatibility positions the platform for versatility and broader integration possibilities.

In summary, the placement preparation DSA module stands as a comprehensive solution, leveraging technology to empower users in their journey towards mastering Data Structures and Algorithms, and ultimately excelling in their placement pursuits.

The collaboration of technical prowess, thoughtful design, and user-centric features solidify its potential impact in the realm of educational technology.

6.2 Future Scope

Although our placement preparation DSA module has successfully achieved its objectives, there are several potential avenues for future development and enhancement:

1. Advanced Learning Paths:

- Introduce advanced learning paths for specialized topics within Data Structures and Algorithms.
- Include modules for advanced algorithms, optimization techniques, and real-world application scenarios.

2. Collaborative Coding Environment:

- Implement a collaborative coding feature, allowing users to work together on coding challenges and projects.
- Support real-time code sharing and collaborative debugging.

3. Integration with Industry Platforms:

- Explore partnerships and integrations with industry platforms for real-world application scenarios and exposure to industry-standard coding practices.

4. Machine Learning and Personalization:

- Implement machine learning algorithms to analyze user performance data and provide personalized recommendations for improvement.
- Tailor learning paths based on individual strengths and weaknesses.

5. Gamification and Challenges:

- Introduce gamification elements such as badges, leaderboards, and virtual challenges to enhance user engagement and motivation.
- Organize coding competitions and challenges to simulate real-world scenarios.

8. Enhanced Chatbot Capabilities:

- Improve chatbot capabilities through continuous integration of natural language processing advancements.
- Enable the chatbot to provide more personalized and context-aware assistance.

9. Interactive Coding Challenges:

- Develop interactive coding challenges that simulate real-world scenarios and require problem-solving skills.
- Include scenarios from different domains like artificial intelligence, machine learning, and cybersecurity.

10. Advanced Analytics and Reporting:

- Enhance visual progress tracking with advanced analytics tools.
- Provide in-depth performance reports with insights into strengths, weaknesses, and areas for improvement.

11. Mobile Application Development:

- Develop a dedicated mobile application to facilitate on-the-go learning and practice.
- Ensure a seamless transition and synchronization between the web and mobile platforms.

13. Community Features:

- Expand community features by introducing forums, interest groups, and mentorship programs.
- Foster a sense of community and collaborative learning among users.

The future scope of the project lies in the pursuit of continuous innovation, adaptability to evolving educational trends, and a steadfast commitment to providing users with a cutting-edge and personalized learning experience in the realm of Data Structures and Algorithms.

REFERENCES

[1] M. P. Burgos, “Learning management system for data structures and algorithm,” *International Journal of Science, Technology, Engineering and Mathematics*, vol. 1, no. 1, pp. 1–26, 2021. doi:10.53378/352852

[2] Z. S. Seidametova, “Some methods for improving data structure teaching efficiency,” *Educational Dimension*, vol. 6, pp. 164–175, 2022. doi:10.31812/educdim.4509

[3] E. B. Costa, A. M. Toda, M. A. Mesquita, F. T. Matsunaga, and J. D. Brancher, “Interactive Data Structure Learning Platform,” *Computational Science and Its Applications – ICCSA 2014*, pp. 186–196, 2014. doi:10.1007/978-3-319-09153-2_14

[4] L. Yu, X. Zheng, and Y. Biao, “Research on data structure course teaching system based on open teaching model,” *Proceedings of the 2nd Symposium on Health and Education 2019 (SOHE 2019)*, 2019. doi:10.2991/sohe-19.2019.73

[5] I. Gaber, A. Kirsh, and D. Statter, “Studied questions in data structures and algorithms assessments,” *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 2023. doi:10.1145/3587102.3588843

[6] N. Adzharuddin, “Learning management system (LMS) among university students: Does it work?,” *International Journal of e-Education, e-Business, e-Management and e-Learning*, 2013. doi:10.7763/ijeeee.2013.v3.233

[7] M. Mcquaigue, D. Burlinson, K. Subramanian, E. Saule, and J. Payton, “Visualization, assessment and analytics in data structures learning modules,” *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018. doi:10.1145/3159450.3159460

[8] D. Dicheva and A. Hodge, “Active learning through game play in a Data Structures course,” *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018. doi:10.1145/3159450.3159605

[9] E. Budiman, H. Haeruddin, U. Hairah, and F. Alameka, “Mobile learning: Visualizing contents media of data structures course in mobile networks,” Journal of Telecommunication, Electronic and Computer Engineering (JTEC), https://jtec.utem.edu.my/jtec/article/view/3877 (accessed Oct. 25 2023).

[10] D. B. Silva, R. de Aguiar, D. S. Dvconlo, and C. N. Silla, “Recent studies about teaching algorithms (CS1) and data structures (CS2) for computer science students,” *2019 IEEE Frontiers in Education Conference (FIE)*, 2019. doi:10.1109/fie43999.2019.9028702

[11] A. Jagadeesha, P. Rayavaram, J. Marwad, S. Narain, and C. S. Lee, “Integrating data structures and algorithms in K-12 education using block-based programming,” *2023 IEEE Global Engineering Education Conference (EDUCON)*, 2023. doi:10.1109/educon54358.2023.10125165

[12] C. Segura, I. Pita, R. del Vado Vírseda, A. I. Saiz, and P. Soler, “Interactive learning of data structures and algorithmic schemes,” *Computational Science – ICCS 2008*, pp. 800–809, 2008. doi:10.1007/978-3-540-69384-0_85

[13] P. Hubwieser, M. Berges, M. Striewe, and M. Goedicke, “Towards competency based testing and feedback: Competency definition and measurement in the field of Algorithms & Data Structures,” *2017 IEEE Global Engineering Education Conference (EDUCON)*, 2017. doi:10.1109/educon.2017.7942896

[14] E. Fouh, S. Hamouda, M. F. Farghally, and C. A. Shaffer, “Automating learner feedback in an eTextbook for data structures and algorithms courses,” *Formative Assessment, Learning Data Analytics and Gamification*, pp. 135–165, 2016. doi:10.1016/b978-0-12-803637-2.00008-7

[15] V. Karavirta and C. A. Shaffer, “Creating engaging online learning material with the JSAV JavaScript algorithm visualization library,” *IEEE Transactions on Learning Technologies*, vol. 9, no. 2, pp. 171–183, 2016. doi:10.1109/tlt.2015.2490673

[16] E. Fouh *et al.*, “Design and architecture of an interactive etextbook – the opendsa system,” *Science of Computer Programming*, vol. 88, pp. 22–40, 2014. doi:10.1016/j.scico.2013.11.040