**CODE BEST APP**

Submitted by

**ARAVIND V**

1P23MC006

In partial fulfillment of the requirements for the award of the Degree of

**MASTER OF COMPUTER APPLICATION**

From Bharathiar University, Coimbatore.

Under the internal supervision of

**Ms. K. Narmatha MCA.,**

**Assistant Professor**

****

**SCHOOL OF COMPUTER STUDIES**

**MASTER OF COMPUTER APPLICATIONS**

**RVS COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)**

**Sulur, Coimbatore – 641402.**

**Nov - 2024**

**RVS COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)**

**SCHOOL OF COMPUTER STUDIES -MCA**

**Master Of Computer Applications**

****

**Nov - 2024**

**Register Number: 1P23MC006**

Certified bona fide Project Work done by ARAVIND V

**Guide** **Director**

Submitted for the Project Evaluation and *Viva voce* held on **\_\_\_\_\_\_\_\_\_**

Internal Examiner External Examiner

##### **Declaration**

**DECLARATION**

Iam **ARAVIND V**, hereby declare that the project entitled **CODE BEST APP,** submitted to the School of Computer Studies, RVS College of Arts and Science, in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a record of original project work done by me during the period June 2024 to November 2024 under the internal supervision of **Ms. K. Narmatha MCA., Assistant Professor, RVS College of Arts and Science (Autonomous)** From Bharathiar University, Coimbatore.

Signature of the Candidate

I

**Acknowlegement**

**ACKNOWLEDGEMENTS**

I express my sincere thanks to our Managing trustee **Dr. K. Senthil Ganesh MBA (USA)., MS (UK)., Ph.D.,** for providing us with the adequate faculty and laboratory resources for completing my project successfully.

I take this as a fine opportunity to express my sincere thanks to **Dr. T. Sivakumar M.Sc., M. Phil., Ph.D., Principal,** RVS College of Arts and Science (Autonomous) for giving me the opportunity to undertake this project.

I express my sincere thanks to **Dr. P. Navaneetham M.Sc., M.Phil., Ph.D., Director (Administration), Department of Computer Science** for the help and advice throughout the project.

I express my sincere thanks to **Dr. S. Yamini M.Sc., (CC)., M. Phil., Ph.D., Director (Academic), Department of Computer Science** for her valuable guidance and prompt correspondence throughout the curriculum to complete the project.

I express my gratitude to **Ms. K. Narmatha MCA., Assistant Professor (MCA)** for his valuable guidance, support, encouragement and motivation rendered by her throughout this project.

Finally, I express my sincere thanks to all other staff members and my dear friends, and all dear and near for helping me to complete this project.

**ARAVIND V**

II

**CODE BEST APP**

**Abstract**

**ABSTRACT**

The Code Best App is a cutting-edge mobile application designed to enhance the learning experience by providing tailored modules for both coding students and instructors. This app functions as a seamless platform where users can view course videos, participate in coding quizzes, and track their progress toward certification. Instructors can efficiently view student user pages, streamlining their tasks. Additionally, the app offers real-time performance tracking for instructors to monitor student progress.

Students can access video lessons, check their quiz results immediately, and engage in interactive assessments to reinforce their learning. The app also provides personalized course recommendations based on individual performance, helping learners focus on areas that need improvement. Developed with a React Native frontend, a Node.js backend, and a MongoDB database, the app is built for reliability and scalability, ensuring smooth performance for all users.

By combining modern technology with essential educational features, the Code Best App promotes transparency, empowers users to take control of their coding education, and provides pathways for certification. The app supports collaboration between peers through coding discussions and project sharing, creating a strong sense of community. With comprehensive tracking and reporting features, both students and instructors can easily review progress, set learning goals, and achieve milestones. This holistic approach fosters a more connected, interactive, and efficient environment for coding education, making it easier for learners to enhance their programming skills and for instructors to manage courses effectively.

III

**Contents**

**CONTENTS**

Declaration I

Acknowledgements II

Abstract III

1. Introduction

1.1 An Overview of the Project 1

1.2 Mission of the Project 1

1.3 Background Study 1

1.3.1 A Study of the Existing System 1

2. System Analysis

2.1 A Study on the Proposed System 2

2.2 User Requirement Specification 2

2.2.1 Major Modules 2

2.2.2 Sub Modules 3

2.3 Software & Hardware Specification 4

2.3.1 Software Specification 4

MongoDB

Node.JS

React Native

Express.JS

2.3.2 Hardware Specification 5

3. System Design and Implementation

3.1 Fundamentals of Design Concepts 6

3.1.1 Abstraction 6

3.1.2 Modularity 7  
 3.1.3 Refinements 7

3.2 Design Models 7 3.2.1 Structure Chart 8

IV

3.2.2 System Flow Diagram 9

3.2.3 Data Flow Diagram 10

3.2.4 Software Engineering Model 12

3.3 [Design process](#_bookmark23)

3.3.1 [Software Architecture](#_bookmark24) 14

3.3.2 [Control Hierarchy](#_bookmark25) 14

3.3.3 [Structural Partitioning](#_bookmark26) 14

3.3.4 [Data structure](#_bookmark27) 15

3.3.5 [Software procedure](#_bookmark28) 15

3.4 Database Design 16

3.5 Input Design 16

3.6 Output Design 16

4. Testing and Implementation 4.1 Testing 17

4.1.1 Testing Methodologies 17 4.2 Quality Assurance 18

4.2.1 Generic Risks 18

4.2.2 Security Aspects and Policies 19

4.3 System Implementation 19

4.4 System Maintenance 19

5. Conclusion

5.1 Directions for Future Enhancements 21

6. Bibliography

6.1 Web References 22

6.1 Book References 23

Annexure-A – Input Design 24

Annexure-B – Output Design 26

Annexure-C – Source Code 31

V

**Introduction**

**1.INTRODUCTION**

This document presents the development and structure of a Programming Subject Related platform called a CODE BEST APP, which enables students easily understand for using this application. The system utilizes the MERN stack, which is a commonly used web development stack composed of four technologies: MongoDB, Express, React Native, and Node.js.

**1.1 An overview of the project**

The aim of this project idea that incorporates Development, Programming, and Database courses. This project would be suitable for a learning platform, and it includes multiple features to showcase different courses, including backend and frontend components. the platform can offer personalized course recommendations based on user activity and progress. It will also support real-time notifications to keep users updated on new courses, quiz deadlines, and progress milestones.

**1.2 Mission of the Project**

The mission of this project is to create an interactive and user-friendly learning platform that provides access to a wide range of Development, Programming, and Database courses. The platform aims to empower users by offering personalized learning experiences, enabling them to enhance their skills through engaging content, quizzes, and real-time feedback while providing admins with powerful tools to manage and track user progress efficiently.

**1.3 Background Study**

**1.3.1 A Study of the Existing System**

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

* Administrator
* Student

1

**System Analysis**

**2. SYSTEM ANALYSIS**

A proposal consisting of potential solutions is provided, and upon the user's request, the proposal is evaluated, and necessary adjustments are made. This process continues until the user is content with the proposal.

Upon the user's request, the proposal undergoes a thorough evaluation, and necessary modifications are implemented until the user expresses satisfaction. The entire system is examined in its entirety, and the inputs are scrutinized to identify any areas of concern.

**2.1 A Study on the Proposed System**

The proposed Learning Management System (LMS) addresses the need for flexible, accessible, and organized online learning platforms. It offers courses across three domains: Development, Programming, and Database management, catering to a diverse audience from beginners to advanced learners.

**2.2 User Requirement Specification**

The User Requirement Specification (URS) details the needs and expectations of various users within the proposed Learning Management System (LMS). The system will serve two main user groups: Learners (Students) and Administrators (Admin), each with unique functional requirements.

**2.2.1 Major Modules**

* **Authentication Module**

This module handles user authentication, including user registration, login, and secure access control.

* **Main Page Module**

The Main Page Module is a fundamental component in software development that often represents the core functionality of an application. It serves as the entry point for users, providing the primary interface and features that define the app's purpose and value.

2

* **Likes Module**

The Likes Module enables users to express appreciation for courses and learning materials in the Learning Management System (LMS), enhancing engagement and providing feedback on popular content.

* **Profile Module**

The Profile Module is a vital feature of the Learning Management System (LMS), allowing users to manage their personal information and track their learning progress, providing a personalized and organized experience on the platform.

* **Security and Privacy Module**

This module is responsible for implementing security measures to protect user data and privacy.

* **User Experience and Usability Module**

This module ensures that the user interface is intuitive and user-friendly. It encompasses aspects of UI/UX design, responsive design, and user testing to optimize the overall user experience.

**2.2.2 Sub Modules**

* **Signup Sub-Module**

Allows users to create an account by providing necessary information such as username, password, email, and personal details.

* **Login Sub-Module**

Provides a secure login mechanism for users to access their accounts.

* **Course Listings Sub-Module**

Display a catalogue of available courses, including images, descriptions, and

videos, quiz, certificate, key information.

3

* **Search and Filter Sub-Module**

Implement search and filter options to help users find specific course based on criteria like course name, image.

* **Course Details Sub-Module**

Allow users to view detailed information about each Course, including descriptions, requirements, instructors, and course content.

* **Likes Page Sub-Module**

Create a page where users can view and manage a list of course they have liked or marked as favourites.

* **Profile Page Sub-Module**

Provide users with a profile page where they can view and update their personal information.

**2.3 Software Specifications and Hardware Specifications**

Software specification, often referred to as a software requirements specification (SRS), is a comprehensive document that outlines the functional and non-functional requirements of a software system or application. Hardware specification, on the other hand, refers to the detailed description of the physical components and characteristics of computer hardware, devices, or systems, providing technical information about the hardware's capabilities and limitations, enabling compatibility, interoperability, and effective utilization.

**2.3.1 Software Specification**

The objective of this document is to provide a comprehensive and precise description of the requirements needed for the development of a web application utilizing the MERN (MongoDB, Express, React Native, Node.js) stack for course adoption. The MERN stack is a well-suited technology stack for creating high-performance web applications. The primary goal of this project is to design a system that facilitates the browsing, searching, selection, and adoption of course by potential adopters.

4

**MongoDB:**

To ensure compatibility with the driver used to connect to the database and Node.js, the version of MongoDB utilized must be specified. MongoDB is a NoSQL database that stores and manages customer data, product information, and order details in a document-oriented manner.

**React Native:**

In a MERN stack project to React Native involves transitioning from a web-based application to a mobile app. React Native allows you to build mobile apps for iOS and Android using JavaScript and React Native.

**Node.js:**

Node.js is a JavaScript runtime environment utilized to build and execute the server-side of the online shopping application. The selection of Node.js version should be consistent with the project's dependencies and libraries.

**Express.js:**

Express.js is a widely-used framework for Node.js that provides a streamlined way to build APIs and manage HTTP requests in web applications. It provides a set of features for building web and mobile applications, including robust routing, middleware support, and easy integration with various template engines.

**2.3.2 Hardware Specification**

The minimum hardware requirements for a MERN stack project will vary based on the scope and intricacy of the application.

* Processor 12th Gen Intel(R) Core (TM) i3-1215U
* RAM 8GB
* Dual-core CPU with a minimum of 1.2 GHz
* High-speed internet connection

5

**Design Modules**

**3. SYSTEM DESIGN AND IMPLEMENTATION**

This section outlines the design and implementation of the Learning Management System (LMS), detailing the architecture, components, technologies used, and the overall workflow of the system. The focus is on how the system meets the user requirements specified in the previous sections while ensuring scalability, security, and usability.

**3.1 Fundamentals of Design Concepts**

When designing the application, it is important to prioritize security by implementing measures such as authentication, authorization, and encryption of sensitive user data. Clear and concise navigation should be used to enhance usability and enable easy access to the desired features.

Additionally, the application should be designed with accessibility in mind, incorporating features like screen readers and keyboard navigation, to ensure that it is usable by individuals with disabilities and meets accessibility guidelines.

Understanding user needs and behaviour through the use of personas and user journeys can help to tailor the system accordingly. Finally, maintaining a consistent visual design and branding throughout the application can create a unified and seamless experience for the user.

Maintain a consistent branding and visual design throughout the application to create a unified experience. Use colours, fonts, and imagery that align with the brand identity.

Design the application with scalability in mind to accommodate future growth and changes. Consider the potential need for additional features and functionality as the business expands.

**3.1.1 Abstraction**

Abstraction in software design involves the process of simplifying complex system components by focusing on their essential characteristics while hiding unnecessary details. It allows developers to create a high-level representation of the system, making it easier to conceptualize and work with.

6

**3.1.2 Refinement**

Refinement is an iterative and essential phase in software development where each module of the course adoption app is continuously improved to meet evolving requirements and enhance overall efficiency.

**3.1.3 Modularity**

The principle of modularity in software design involves breaking down a complex application into smaller, independent modules or components. Each module focuses on a specific set of functionalities or features, making it easier to develop, test, and maintain. By dividing the Code best app into these distinct and self-contained modules, developers can work on individual parts of the application without impacting other areas.

**3.2 Design Modules**

A design module in the context of software development and system design typically refers to a self-contained unit or component that focuses on a specific aspect of the system's design. It encapsulates a set of related functionalities or features that work together to achieve a particular goal within the overall system.

**Course Listing Module:**

The Course Listing Module enables users to browse and enroll in available courses across categories like Development, Programming, and Database Management. It provides a structured interface for easy navigation and discovery of course offerings.

**User Profile Module:**

The User Profile Module allows users to manage their personal information and track their learning progress within the Learning Management System (LMS). This module enhances user experience by providing personalized insights and easy access to relevant data.

**Search and Filter Module:**

The **Search and Filter Module** enhances the Learning Management System (LMS) by enabling users to efficiently find and access courses based on specific criteria. This module improves the overall user experience by providing quick navigation and tailored course recommendations.

7

**Likes and Favourites Module:**

The Likes and Favorites Module allows users to express their appreciation for courses and mark their preferred content within the Learning Management System (LMS). This module enhances user engagement and helps users easily access their favorite courses.

**Authentication and Authorization Module**

The Authentication Module is tasked with ensuring secure user interactions within the course adoption application. It manages user authentication, facilitating a secure and controlled login process. By implementing robust access control mechanisms, this module ensures that only authorized users can access the application's features and data, safeguarding user information and system integrity.

**3.2.1 System Structure Chart**

This architecture has four layers. The first layer is responsible for managing data storage and retrieval. It includes components like data access objects and MongoDB databases. The second layer handles business logic and data management. It includes components such as Node.js modules, server-side scripts, and MongoDB models. The third layer exposes the application's functionality through a RESTful API. It includes components like RESTful endpoints. The fourth layer manages communication between the front-end and back-end, handling requests and responses. It includes components like Express middleware.

8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Client  Layer | HTTP  Req  HTTP  Res | Backend  Server | Data  Req  Data  Res | Database  Server |
| |  | | --- | | React Native |  |  | | --- | | Mobile | | |  | | --- | | Express Application |   Node Server | Database Layer  MongoDB  Database |

**Front-end:**

React Native allows developers to build native mobile app UI components that work seamlessly on both iOS and Android platforms, offering a consistent and responsive user experience.

**Back-end:**

This layer handles server-side logic and interacts with the database. It's built using Node.js, a JavaScript runtime for building scalable network applications.

**Database:**

This layer stores and retrieves data for the application. It's typically built using MongoDB, a popular NoSQL database that stores data in JSON-like documents.

**3.2.2 System Flow Diagram**

Creating a system flow diagram for a project is an essential step in visualizing and understanding the project's processes and components. This diagram serves as a roadmap, outlining the sequential order of tasks, decision points, data flow, and dependencies within the project. It begins with the project's initiation trigger and progresses through key components, illustrating how they interact and influence each other.

9

Decision points are highlighted to indicate where choices are made, and loops or iterations are included to account for repetitive processes. Exception handling and feedback mechanisms are integrated to ensure a robust representation of the project's functionality. Overall, the system flow diagram is a valuable tool for project planning, communication, and analysis, enabling stakeholders to grasp the project's flow and dependencies at a glance.

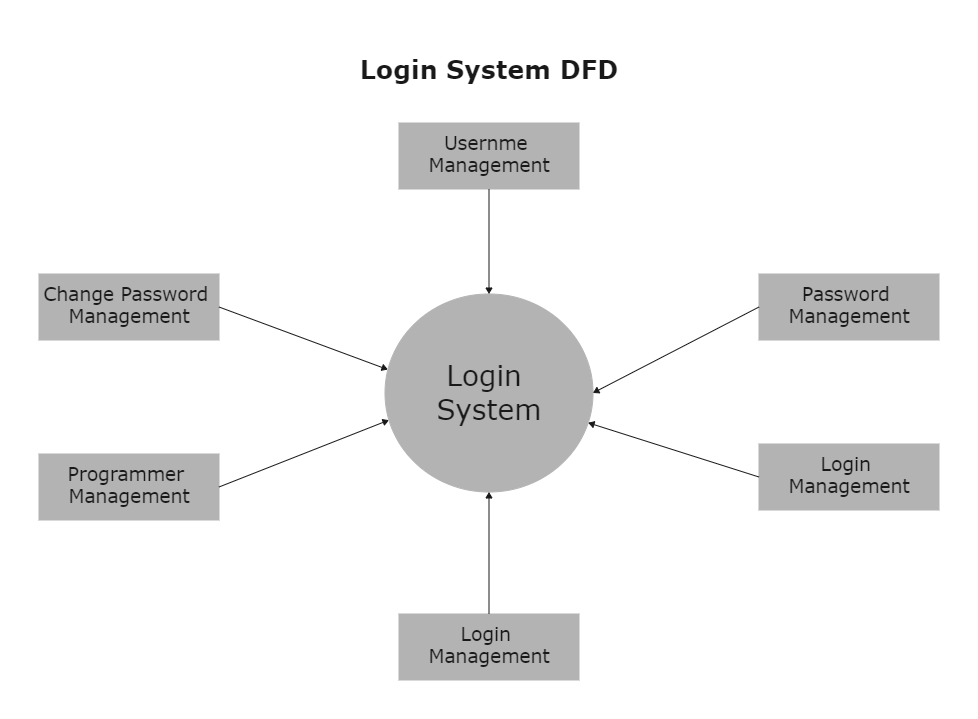
**3.2.3 Data Flow Diagram**

A data-flow diagram (DFD) is a visual representation that illustrates how data moves through a process or system, typically within an information system. It shows how data enters and exits various entities and processes. DFDs emphasize data flow and do not depict control flow, meaning they don't contain decision rules or loops. Instead, they focus on mapping the paths that data takes within a system, making them valuable tools for understanding data relationships and processes in a simplified and structured manner.

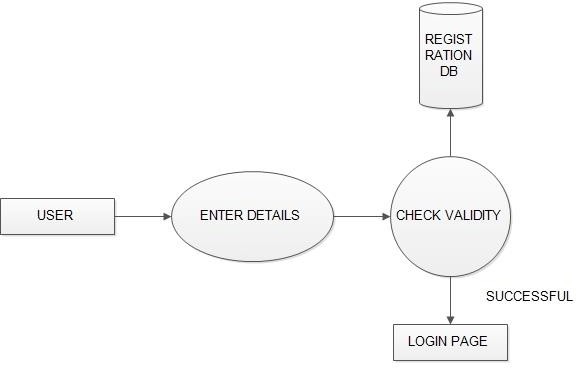
* **USER PAGE**
* **ADMIN PAGE**

10

* **LOGIN DFD**

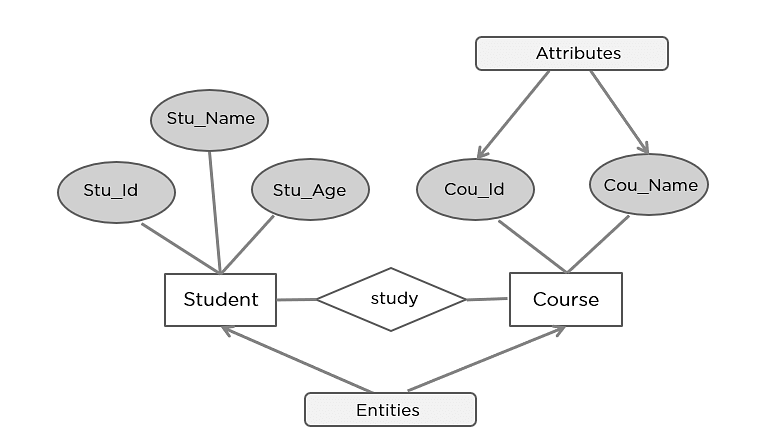
****

* **REGISTRATION DFD**



11

* **ER DIAGRAM**



**3.2.4 Software Engineering Model**

Software engineering models are essential conceptual frameworks and methodologies that serve as roadmaps for software development. They provide structure and guidance throughout the entire software development lifecycle, from initial planning and design to actual coding, testing, and ongoing maintenance. These models help ensure that software projects are organized, efficient, and produce high-quality results.

**Agile Model:**

Agile methodologies like Scrum or Kanban are popular for mobile app development. They emphasize flexibility, collaboration, and delivering incremental updates to meet evolving user needs. Agile allows for continuous improvement and frequent releases.

**Waterfall Model:**

While not as common for mobile apps, the waterfall model can still be used for simple and well-defined projects where requirements are stable from the beginning. It follows a sequential approach, with each phase being completed before moving to the next.

12

**Iterative and Incremental Models:**

DevOps principles are instrumental in mobile app development as they streamline the entire lifecycle of an app, from its initial development phase through deployment and ongoing maintenance. By fostering collaboration between development and operations teams, DevOps promotes faster development cycles, shorter release times, and improved overall quality. This approach also encourages automation of repetitive tasks, ensuring consistency and reducing human errors.

**DevOps Model:**

DevOps principles play a crucial role in mobile app development by streamlining the entire process, from initial development to deployment and ongoing maintenance.

**Mobile-First Design Approach:**

The mobile-first design approach, although not a conventional software engineering model, is a critical mindset in modern application development. It places mobile devices as the foremost platform for design and development, prioritizing user experiences on smartphones and tablets.

**Prototyping Model:**

Creating prototypes or mock-ups of a mobile app's user interface is a valuable practice in the app development process. These visual representations serve as tangible examples that help clarify design and functionality requirements.

**Hybrid Model:**

Mobile app development is a dynamic field that frequently combines a variety of methodologies, tools, and techniques to cater to the unique requirements of each project. Development teams often blend elements from different software development models, such as Waterfall, Agile, or DevOps, to create a customized approach.

13

### **3.3 Design process**

### **3.3.1 Software Architecture**

* Software architecture is the set of structures needed to reason about a software system and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations.
* The architecture of a software system is a metaphor, analogous to the architecture of a building. It functions as the blueprints for the system and the development project, which project management can later use to extrapolate the tasks necessary to be executed by the teams and people involved.

### **Control Hierarchy**

* Control hierarchy, also called program structure, represents the organization of program components (modules) and implies a hierarchy of control. It does not represent procedural aspects of software such as sequence of processes, occurrence or order of decisions, or repetition of operations; nor is it necessarily applicable to all architectural styles.
* The control hierarchy also represents two subtly different characteristics of the software architecture: visibility and connectivity. Visibility indicates the set of program components that may be invoked or used as data by a given component, even when this is accomplished indirectly.

### **3.3.3 Structural Partitioning**

The program structure should be partitioned both vertically and horizontally. As describe in horizontal partitioning describe separate branches of the modular hierarchy of reach major program function. The Control modules, represented in a darker shade are used to coordinate communication among and execution of program functions. The easiest approach to horizontal partitioning describes 3 partitions - input, data transformation often called processing and output.

The nature of modification in program architectures justifies the requirement for vertical partitioning. The change in a control module high in the architecture will have a higher possibility of propagating side effects to modules are subordinate to it. The change to a worker module given its low level in the structure is less likely to cause the propagation effects. In common changes to computer programs revolve around changes to input.

14

### **3.3.4 Data structure**

Data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.

Depending on your requirement and project, it is important to choose the right data structure for your project. For example, if you want to store data sequentially in the memory, then you can go for the Array data structure.

Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by a pointer—a bit string, representing a memory address, that can be itself stored in memory and manipulated by the program. Thus, the array and record data structures are based on computing the addresses of data items with arithmetic operations, while the linked data structures are based on storing addresses of data items within the structure itself.

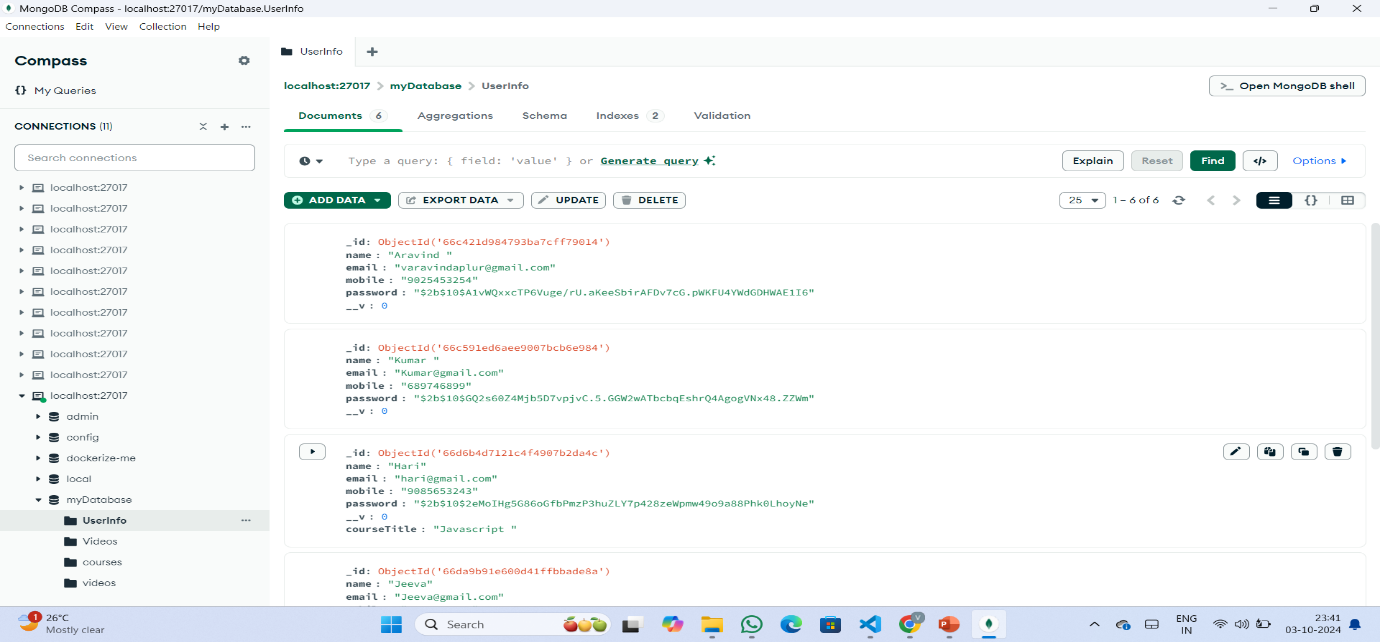
### **3.3.5 Software procedure**

The development process begins with crafting a high-level design that defines the application's architecture and selecting the appropriate technology stack. Secure user authentication and authorization mechanisms are implemented, using methods like email/phone verification and password hashing.

Coding Learning offers structured video lessons, interactive quizzes, and real-time collaboration to enhance coding skills. Learners earn certificates upon course completion, showcasing their achievements.

15

**3.4 Database Design**

The Learning Management System (LMS) database is designed using MongoDB for flexibility and scalability, supporting user authentication, course enrollment, and quiz management. Its document-based structure enables efficient queries and easy integration of features like progress tracking and performance analytics. 

**3.5 Input Design**

The input design for the Learning Management System (LMS) focuses on creating user-friendly, efficient forms and interfaces that capture necessary data accurately. It ensures smooth interactions for both learners and administrators by providing clear and intuitive input methods. These include login and registration forms, allowing users to easily input personal and credential details. Course creation and enrollment forms enable admins to add course information, while students can enroll with minimal effort. Additionally, quiz submission forms, search filters, and profile management inputs are designed to streamline user interactions across the platform.

**3.6 Output Design**

The output design for the Learning Management System (LMS) ensures that data is presented to users in a clear, organized, and accessible format. It focuses on delivering relevant information effectively to both learners and administrators. Course listings display key details like titles, categories, and instructors in a user-friendly manner. Learners receive progress reports that show course completion, quiz scores, and overall performance. Quiz results are provided immediately, offering feedback to help users improve.

16

14

**Testing and Implementation**

**4. TESTING AND IMPLEMENTATION**

**4.1 Testing**

Software testing is a critical process aimed at identifying errors and defects in software. It serves as a final review of all the previous phases of software development, including specification, design, and code generation. Ultimately, software testing is a crucial part of the software development life cycle, serving as a key means of guaranteeing that the software functions as intended and meets the expectations of users.

**4.1.1 Testing Methodologies**

**Functional Testing:**

This type of testing ensures that the application functions as expected based on the requirements specification. Functional testing for a CODE BEST APP would include testing features such as user registration, course search, course video, quiz, and certificates.

**Usability Testing:**

This type of testing ensures that the application is easy to use and understand for the intended users. Usability testing for a CODE BEST APP would involve testing the user interface design, navigation, and overall user experience.

**Performance Testing:**

This type of testing ensures that the application performs well under different conditions, such as high traffic or slow network connections. Performance testing for a CODE BEST APP would include testing page load times, response times for API calls, and overall system performance under different loads.

**Security Testing:**

This type of testing ensures that the application is secure and protects user data from unauthorized access or other security threats. Security testing for a CODE BEST APP would include testing authentication and authorization, input validation, and handling of sensitive data such as quiz and certificate information.

17

**4.2 Quality Assurance**

In software development, Quality Assurance (QA) refers to the process of establishing and adhering to quality metrics and standards to ensure that a software application is developed and tested to meet the specified requirements.

This involves developing and adhering to coding standards to ensure consistency and maintainability, conducting code reviews to identify and address any issues or bugs, performing continuous integration and testing to ensure that new code does not introduce any defects, and conducting user acceptance testing to ensure that the system meets the needs of its users. Ultimately, the goal of QA is to verify that a software application functions as intended and meets the expected quality standards.

**4.2.1 Generic Risks**

A project can encounter a range of generic risks that have the potential to impede its success. Security risks, for example, may involve data breaches or hacking that could compromise the confidentiality and integrity of sensitive data.

Technical risks such as system failure, bugs, and scalability issues can impact the performance and availability of the system. Legal risks such as non-compliance with regulations and laws could lead to legal action and reputational damage.

These generic risks are applicable to all projects, regardless of their industry or specific technology used.

CODE BEST APP may face a variety of generic risks that could potentially impact their success. Security risks such as data breaches and hacking attacks can compromise the confidentiality and integrity of sensitive data.

Technical risks like system failures, bugs, and scalability issues can affect the system's performance and availability.

Legal risks such as non-compliance with regulations and laws can lead to legal action and damage to the organization's reputation. Operational risks such as insufficient infrastructure, inadequate project management, and lack of resources can also impact the success of the project.

18

**4.2.2 Security Aspects and Policies**

In CODE BEST APP projects, security is a critical aspect that requires several policies and protocols to be implemented. Regular security audits should be conducted to assess the application's security posture and identify any areas for improvement.

To ensure that the project is secure, software updates and patches should be applied regularly to protect against known vulnerabilities. It is important to encrypt sensitive data such as credit card information and personal details using encryption technologies like SSL/TLS and HTTPS to protect data at rest and data in transit.

Secure features like quiz functionality and certificate generation will ensure that learners can complete assessments and receive recognition for their achievements, while robust access control measures will protect sensitive user data within the Learning Management System (LMS).

Strong user authentication systems, regular security audits, and backup and disaster recovery plans are essential to prevent unauthorized access and protect against potential data loss. Compliance with regulatory requirements such as GDPR, CCPA, and HIPAA is also necessary to maintain data security and protect customers' privacy.

**4.3 System Implementation**

System Implementation is a critical phase of a project where the theoretical design is transformed into a functional system. However, if not carefully controlled and planned, it can cause chaos and undermine the project's success.

Therefore, a well-planned and structured approach is necessary for the successful implementation of a CODE BEST APP project.

This involves creating a detailed plan that outlines the project's scope, objectives, timelines, and milestones, and selecting the appropriate CODE BEST APP platform to meet specific business needs. Thorough testing is crucial to ensure the system works correctly, and all features and functionalities are functioning as intended.

After testing is complete, the website can be launched and deployed, and post-launch support should be provided to ensure the website runs smoothly, and any issues or bugs are promptly addressed.

19

**4.4 System Maintenance**

Maintaining a MERN (MongoDB, Express, React Native, Node.js) stack involves several key activities. These include:

**1. Regularly updating dependencies**

MERN applications rely on various third-party libraries and frameworks, and it is essential to keep them up to date to ensure optimal performance and security.

**2. Monitoring server performance**

Monitoring server performance is crucial to ensure that the application is running smoothly and to identify any performance issues or bottlenecks that may arise.

**3. Backing up data**

Regularly backing up application data is essential to ensure that data is not lost in the event of a system failure or other catastrophic event.

**4. Testing**

Regularly testing the application's various components and features is critical to identify any bugs or errors that may occur and to ensure that the application functions as intended.

20

**Conclusion**

**5. CONCLUSION**

In summary, developing a Code best application using the React Native, Node.js, and MongoDB stack establishes an agile and effective platform for linking prospective course with their companions. This technology combination offers versatility across platforms, real-time functionality, scalability, and adaptability when managing course-related data. This not only guarantees a seamless and user-friendly experience but also champions the cause of program welfare and responsible courses. By harnessing the strengths of React Native, Node.js, and MongoDB.

Furthermore, the conversion of a Code best mobile application into a MERN stack-based solution lays a robust foundation for success. By adopting React Native for the frontend and implementing Node.js and Express for the backend, you can establish a highly efficient and user-centric platform for connecting course enthusiasts with their course companions.

The utilization of MongoDB for data storage ensures adaptability and scalability, which are pivotal for efficiently managing course-related information. To successfully execute this project, it is imperative to adhere to best practices, encompassing the definition of clear project objectives, the creation of a meticulously planned project structure, the design and development of essential features, comprehensive testing, and the implementation of effective marketing strategies.

**5.1 Directions for Future Enhancements**

Future enhancements for the Learning Management System (LMS) can focus on improving user experience and expanding functionality. Key areas include implementing personalized learning paths through adaptive algorithms that tailor course recommendations based on user performance. Introducing gamification elements, such as badges and leaderboards, can motivate learners and increase engagement. Developing a dedicated mobile application will provide users with seamless access to courses and quizzes on-the-go. Additionally, integrating an AI-powered chatbot can assist users with FAQs and technical support, further enhancing the overall user experience.

21

**Bibliography**

**6. BIBLIOGRAPHY**

**6.1 Web References:**

**React Native**

<https://reactnavigation.org/docs/getting-started/>

<https://www.tutorialspoint.com/react_native/index.htm>

<https://reactnative.dev/><https://reactnavigation.org/><https://reactnative.dev/docs/profiling><https://reactnative.dev/docs/signed-apk-android><https://reactnative.dev/docs/testing-overview>

**Redux**

<https://redux.js.org/>

**NodeJS**

<https://nodejs.dev/en/learn/>

<https://www.tutorialspoint.com/nodejs/index.htm>

**MongoDB**

<https://www.tutorialspoint.com/mongodb/index.htm>

**Express**

<https://www.tutorialspoint.com/expressjs/index.htm>

**NPM Website**

<https://www.npmjs.com/>

**NVM**

<https://github.com/nvm-sh/nvm>

22

**6.2 Book References:**

1."Learning React: A Hands-On Guide to Building Web Applications Using React and Redux", Author: Kirupa Chinnathambi, Publisher: Addison-Wesley Professional, Year: 2020, Edition: 2nd Edition

2."Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", Author: Vasan Subramanian, Publisher: A press, Year: 2019, Edition: 2nd Edition

3."Full-Stack React Projects: Learn MERN Stack Development by Building Modern Web Apps Using MongoDB, Express, React, and Node.js", Author: Shama Hoque, Publisher: Packet Publishing, Year: 2020, Edition: 2nd Edition

4."The Road to React: Your Journey to Master React.js in JavaScript”, Author: Robin Wieruch,

Publisher: Independently Published, Year: 2020, Edition: 4th Edition

5."MongoDB: The Definitive Guide: Powerful and Scalable Data Storage", Author: Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, Publisher: O'Reilly Media, Year: 2019, Edition: 3rd Edition

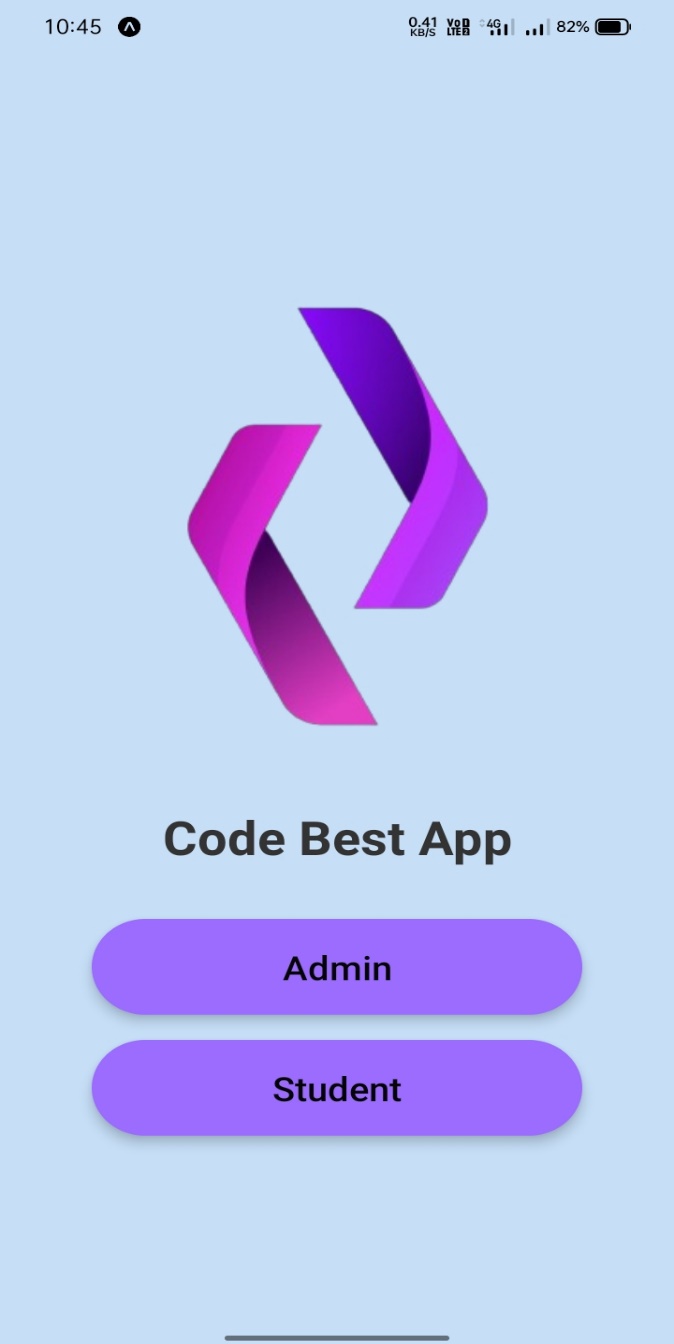
These books provide in-depth knowledge on MERN stack development, from React front-end programming to server-side scripting with Node.js and managing databases using MongoDB.

23

**Annexures**

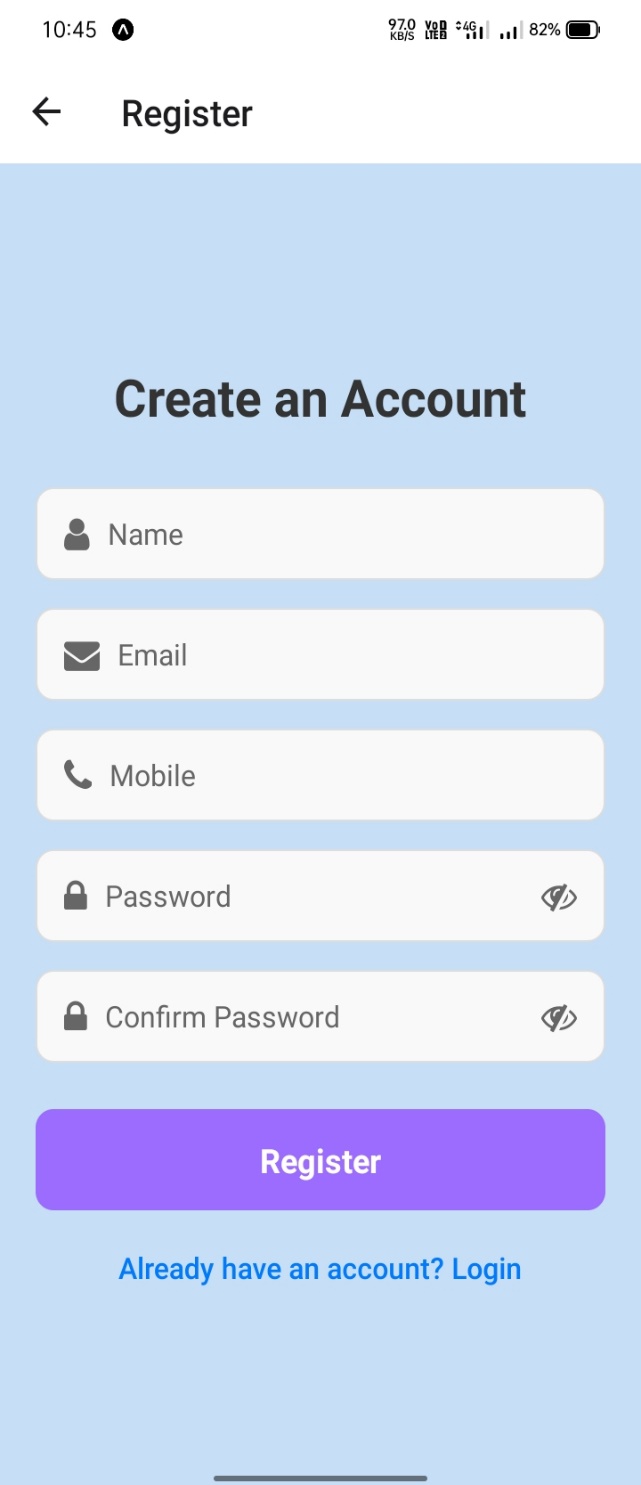
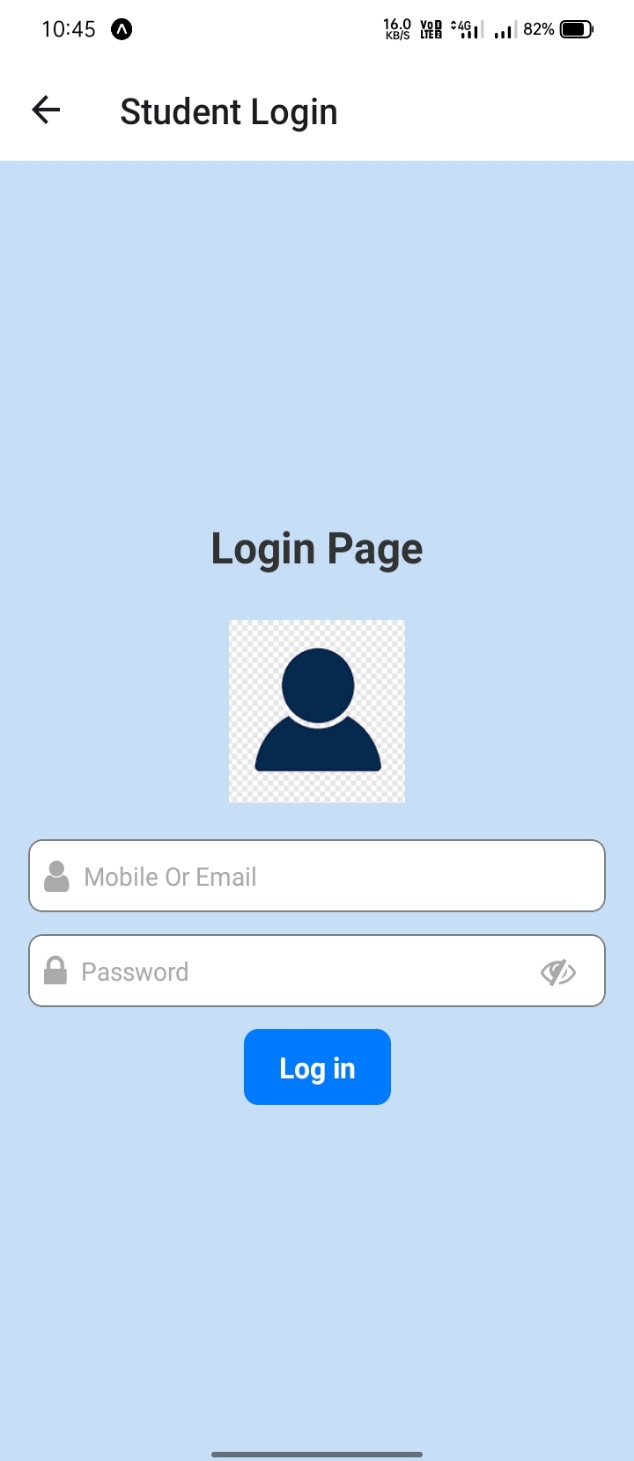
**Annexure-A – Input Design**

**Layout Screen: Main Screen:**



24

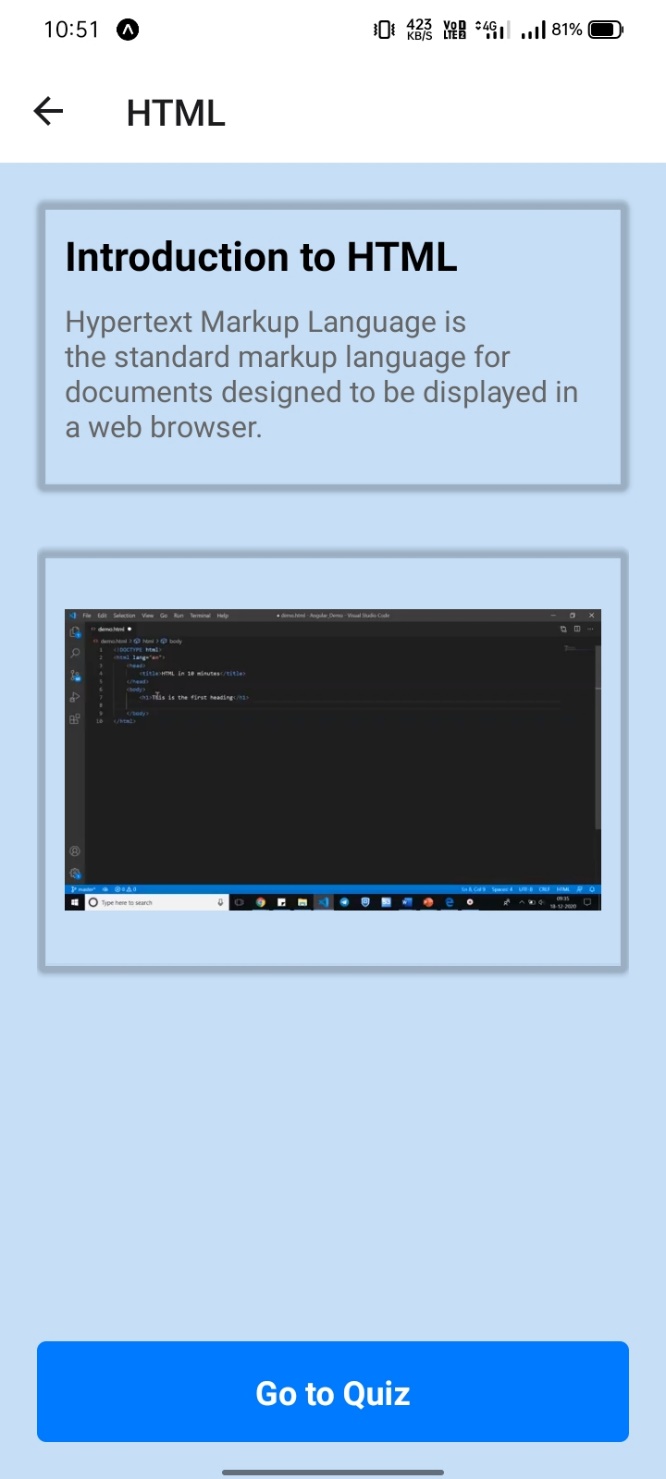
Register Screen Login Screen

25

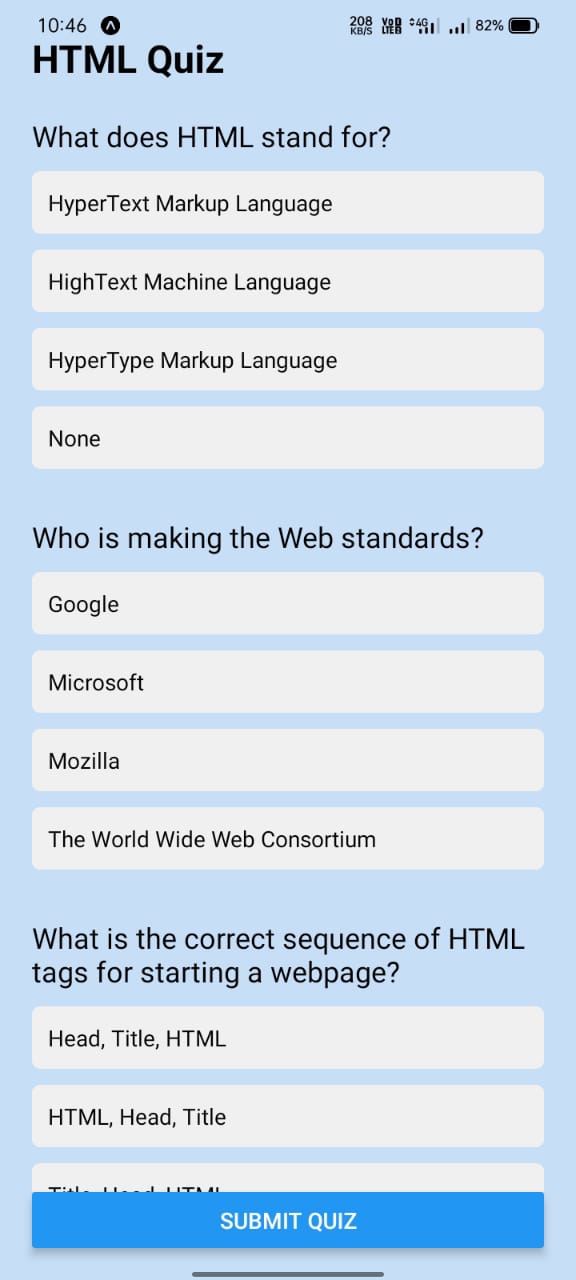
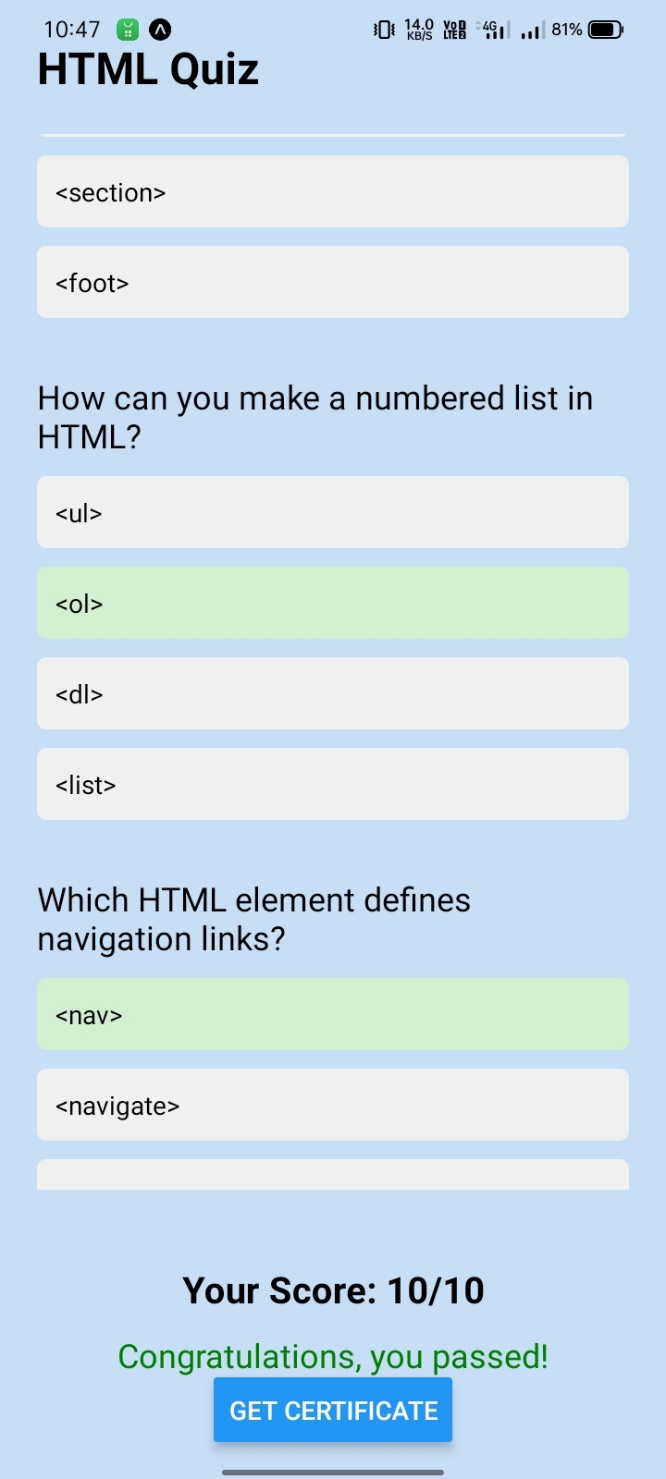
**Annexure-B – Output Design**

**Home Screen Course Screen**

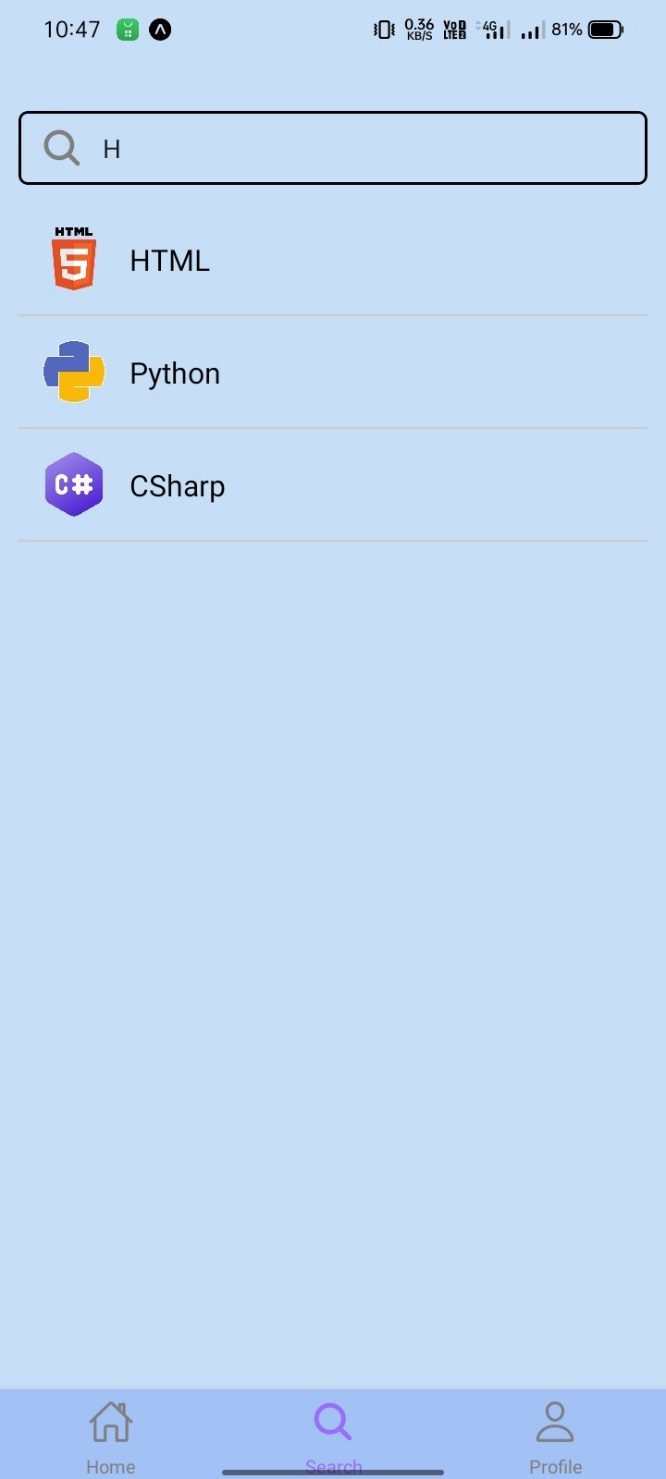
26

**Quiz Screen After Quiz Screen**

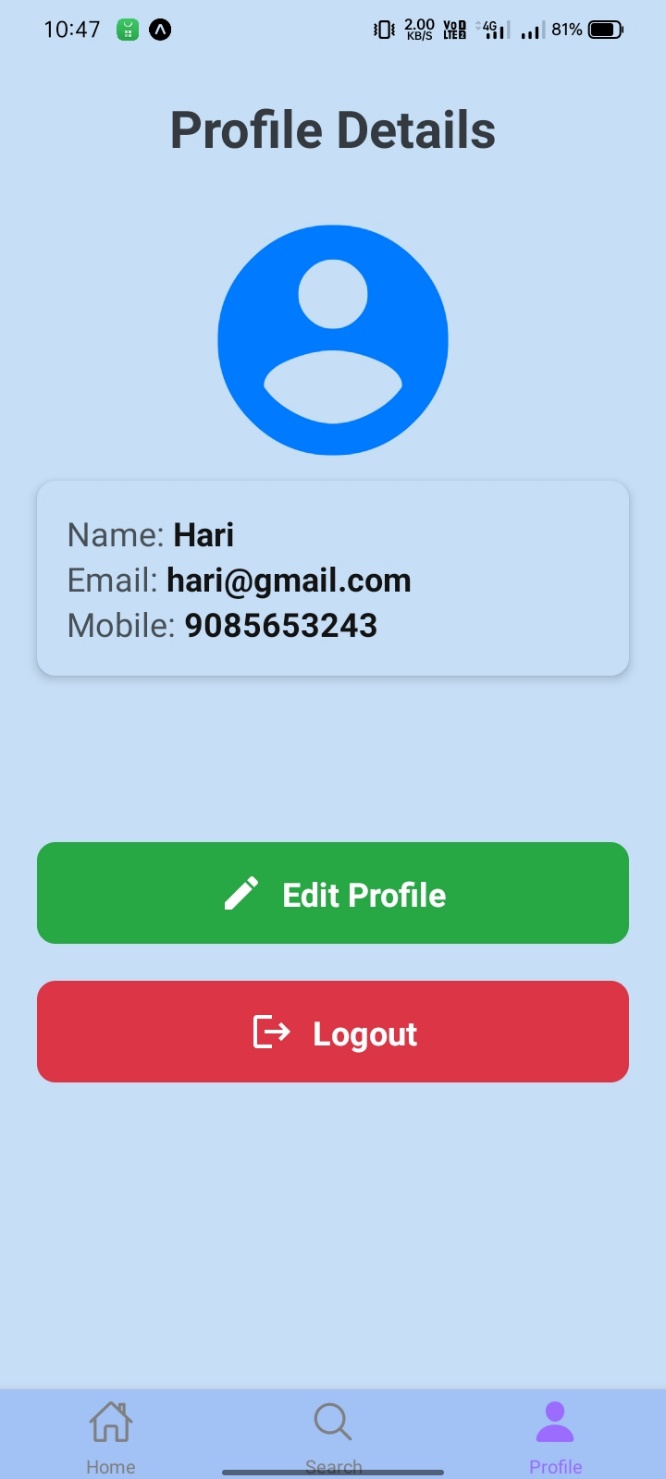
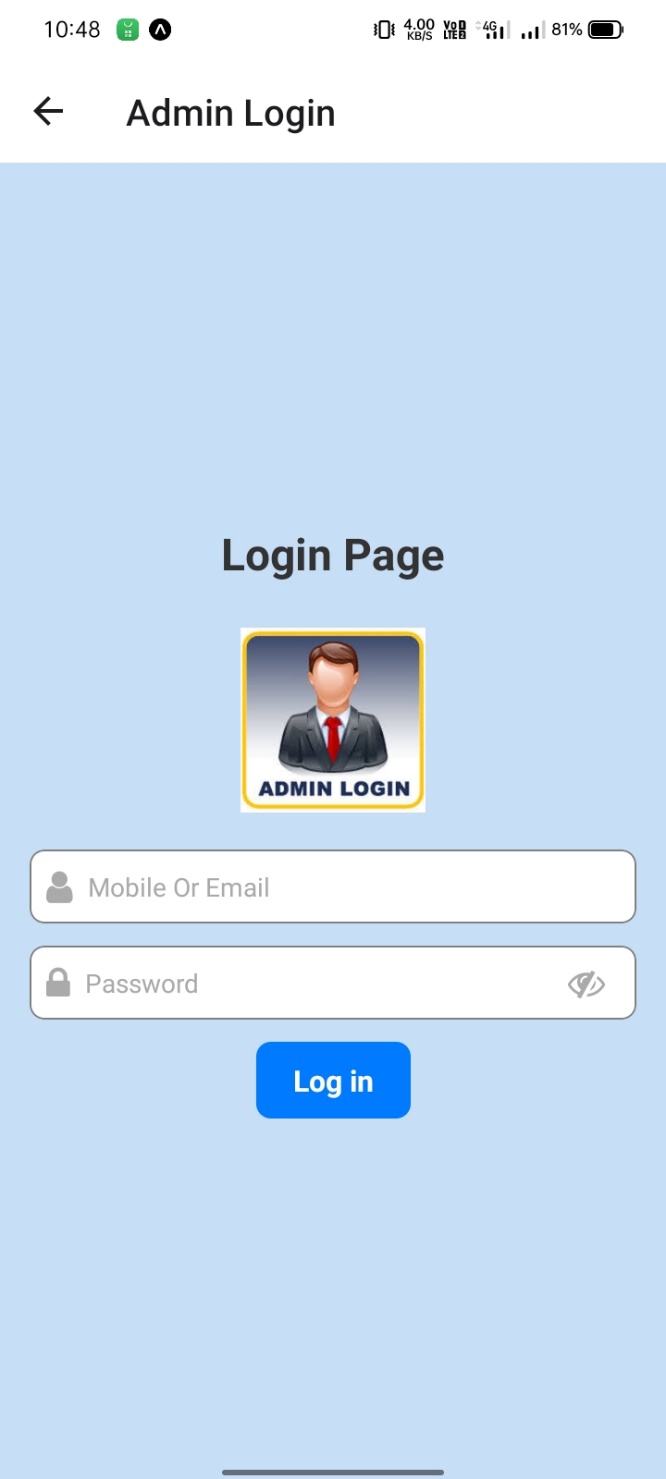
27

**Certificate Screen Search Screen**

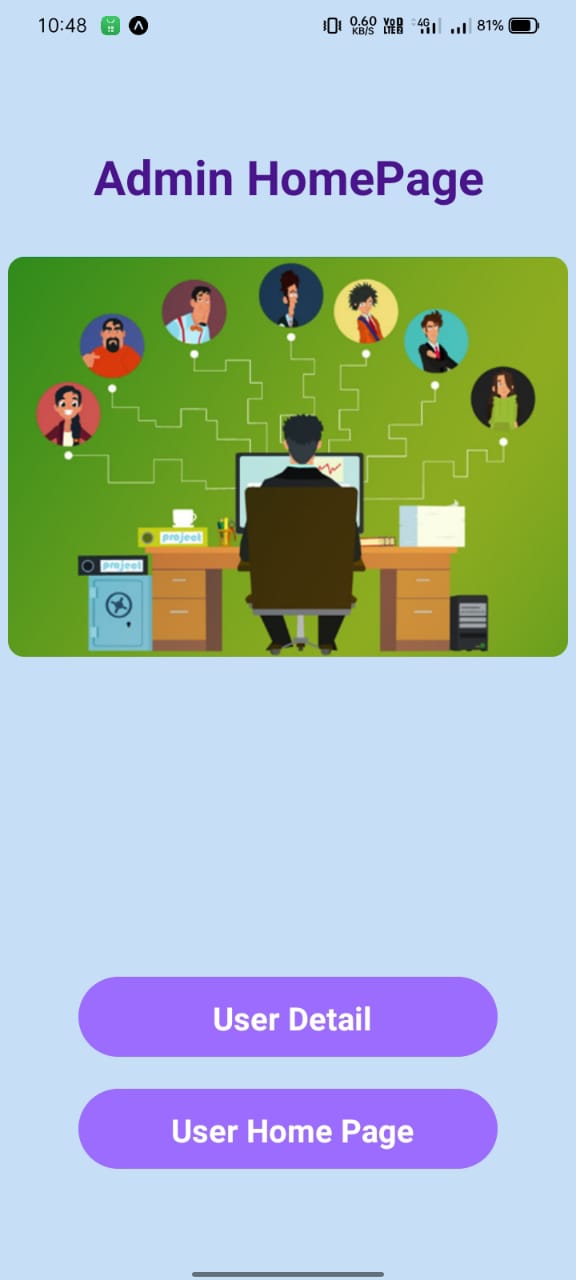
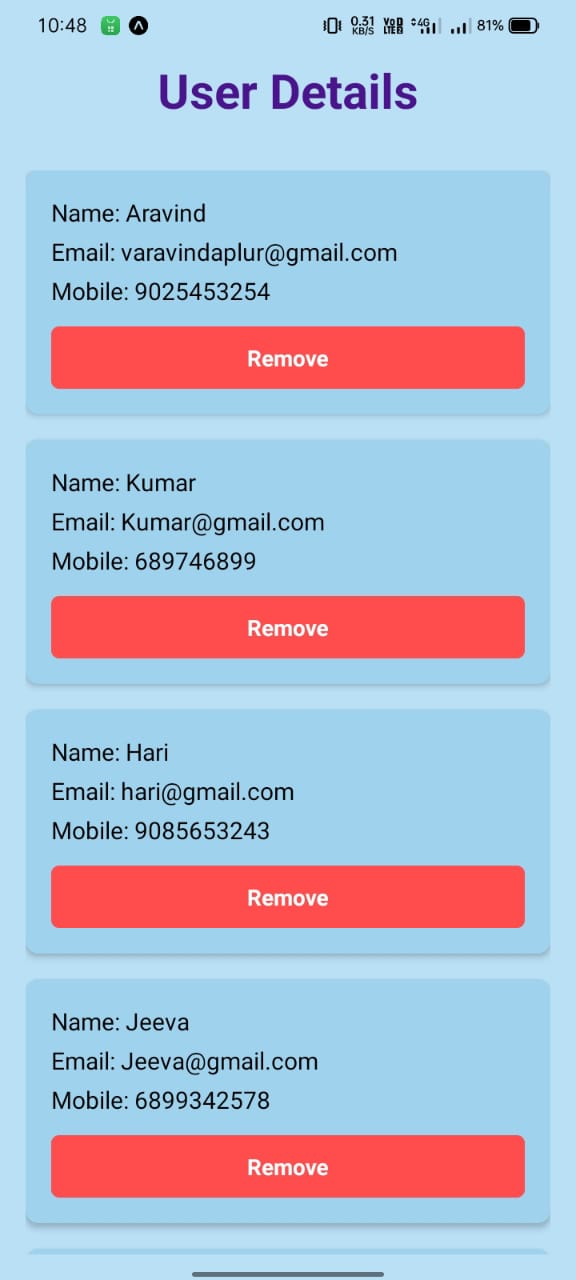
28

**Profile Screen Admin Login Screen**

29

**Admin Home Screen User Details Screen**

**Annexure-C – Source Code**

30

**App.js**

import { StyleSheet, Text, View } from 'react-native'

31

import React from 'react'

import Screen from './screen'

export default function App() {

return (

<Screen/>

)

}

const styles = StyleSheet.create({})

**LoginPage.js**

import React, { useState } from 'react';

import axios from 'axios';

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert, Image } from 'react-native';

import { useNavigation } from '@react-navigation/native';

import Icon from 'react-native-vector-icons/FontAwesome';

import AsyncStorage from '@react-native-async-storage/async-storage';

const HARD\_CODED\_ADMIN = {

email: 'admin@example.com',

password: 'adminpassword'

};

export default function LoginPage({ route }) {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [showPassword, setShowPassword] = useState(false); // State for toggling password visibility

const navigation = useNavigation();

const { type } = route.params || {}; // get 'type' from route params

const handleSubmit = async () => {

if (!email || !password) {

Alert.alert('Please fill in both email and password.');

return;

}

if (type === 'Admin') {

// Hardcoded admin login validation

if (email === HARD\_CODED\_ADMIN.email && password === HARD\_CODED\_ADMIN.password) {

Alert.alert('Logged In Successfully');

navigation.navigate('AdminHomePage'); // Navigate to 'AdminHomePage' for admin login

return;

} else {

Alert.alert('Login Failed', 'Please check your email and password and try again.');

return;

}

}

// For user login

const userData = { email: email, password: password };

try {

const response = await axios.post("http://192.168.68.231:8002/login-user", userData);

console.log(response.data);

32

navigation.navigate('UserHomePage'); // Navigate to 'UserHomePage' for user login

} else {

Alert.alert('Login Failed', 'Please check your email and password and try again.');

}

} catch (error) {

console.error(error);

Alert.alert('An error occurred during login', 'Please try again later.');

}

};

return (

<View style={styles.container}>

<Text style={styles.title}>Login Page</Text>

{/\* Admin Image \*/}

{type === 'Admin' && (

<Image

source={require('../assets/adminlogin.jpeg')} // Replace with your admin image path

style={styles.image}

/>

)}

{/\* User Image \*/}

{type === 'User' && (

<Image

source={require('../assets/userlogin.png')} // Replace with your user image path

style={styles.image}

/>

)}

33

**RegisterPage.js**

import React, { useState } from 'react';

import axios from 'axios';

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert } from 'react-native';

import { useNavigation } from '@react-navigation/native';

import Icon from 'react-native-vector-icons/FontAwesome';

export default function RegisterPage() {

const navigation = useNavigation();

const [name, setName] = useState("");

const [email, setEmail] = useState("");

const [mobile, setMobile] = useState("");

const [password, setPassword] = useState("");

const [confirmPassword, setConfirmPassword] = useState(""); // State for confirm password

const [showPassword, setShowPassword] = useState(false);

const [showConfirmPassword, setShowConfirmPassword] = useState(false); // State for toggling confirm password visibility

function validateEmail(email) {

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return emailRegex.test(email);

}

function validateMobile(mobile) {

const mobileRegex = /^[0-9]{10}$/;  // Example: 10 digits mobile number

return mobileRegex.test(mobile);

}

34

Alert.alert('Validation Error', 'Name is required.');

return;

}

if (!email || !validateEmail(email)) {

Alert.alert('Validation Error', 'Please enter a valid email address.');

return;

}

if (!mobile || !validateMobile(mobile)) {

Alert.alert('Validation Error', 'Please enter a valid 10-digit mobile number.');

return;

}

if (!password || password.length < 6) {

Alert.alert('Validation Error', 'Password should be at least 6 characters long.');

return;

}

if (password !== confirmPassword) { // Check if password and confirm password match

Alert.alert('Validation Error', 'Passwords do not match.');

return;

}

const userData = { name, email, mobile, password };

axios.post("http://192.168.68.231:8002/register", userData)

// axios.post("http://192.168.209.3:8002/register", userData)

.then((res) => {

35

})

.catch((error) => {

console.error('Error during registration:', error.response ? error.response.data : error.message);

Alert.alert('Registration Failed', 'An error occurred during registration. Please try again.');

});

}

function handleNameInput(text) {

const regex = /^[a-zA-Z\s]\*$/;  // Allows letters and spaces only

if (regex.test(text)) {

setName(text);

}

}

function handleMobileInput(text) {

const validMobile = text.replace(/[^0-9]/g, ''); // Remove non-numeric characters

if (validMobile.length <= 10) {  // Ensure max 10 digits

setMobile(validMobile);

}

}

return (

<View style={styles.container}>

<Text style={styles.title}>Create an Account</Text>

<View style={styles.inputContainer}>

<View style={styles.inputWithIcon}>

<Icon name="user" size={20} color="#666" style={styles.icon} />

<TextInput

placeholder='Name'

style={styles.textInput}

36

}

value={name}

/>

</View>

<View style={styles.inputWithIcon}>

<Icon name="envelope" size={20} color="#666" style={styles.icon} />

<TextInput

placeholder='Email'

style={styles.textInput}

placeholderTextColor="#666"

onChangeText={setEmail}

value={email}

keyboardType='email-address'

autoCapitalize='none'

/>

</View>

<View style={styles.inputWithIcon}>

<Icon name="phone" size={20} color="#666" style={styles.icon} />

<TextInput

placeholder='Mobile'

style={styles.textInput}

placeholderTextColor="#666"

onChangeText={handleMobileInput}

value={mobile}

keyboardType='phone-pad'

/>

</View>

<View style={styles.inputWithIcon}>

<Icon name="lock" size={20} color="#666" style={styles.icon} />

<TextInput

placeholder='Password'

37

value={password}

secureTextEntry={!showPassword}

/>

<TouchableOpacity onPress={() => setShowPassword(!showPassword)}>

<Icon name={showPassword ? "eye" : "eye-slash"} size={20} color="#666" />

</TouchableOpacity>

</View>

<View style={styles.inputWithIcon}>

<Icon name="lock" size={20} color="#666" style={styles.icon} />

<TextInput

placeholder='Confirm Password'

style={styles.textInput}

placeholderTextColor="#666"

onChangeText={setConfirmPassword}

value={confirmPassword}

secureTextEntry={!showConfirmPassword}

/>

<TouchableOpacity onPress={() => setShowConfirmPassword(!showConfirmPassword)}>

<Icon name={showConfirmPassword ? "eye" : "eye-slash"} size={20} color="#666" />

</TouchableOpacity>

</View>

<TouchableOpacity style={styles.button} onPress={handleSubmit}>

<Text style={styles.buttonText}>Register</Text>

</TouchableOpacity>

</View>

<View>

38