HOME CARE SERVICE FOR SENIORS

# 21IT902 – ADVANCED APPLICATION DEVELOPMENT

***Submitted by***

|  |  |
| --- | --- |
| **MOHAMMED SHARAFATH P** | **(727722EUCD023)** |
| **SUBAHARINI N** | **(727721EUIT161)** |
| **SWETHA S** | **(727721EUIT168)** |

***in partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

INFORMATION TECHNOLOGY

# SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

**(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)**

MARCH 2024



**BONAFIDE CERTIFICATE**

Certified that this project report titled **“HOME CARE SERVICE FOR SENIORS”** is the bonafide work of **SAHANADEVI P (727721EUIT136), SUBAHARINI N (727721EUIT161),**

**SWETHA S (727721EUIT168)** who carried out the project work under my supervision.

# SIGNATURE SIGNATURE

Dr. N. SUSILA Ms. R. JANANI

# HEAD OF THE DEPARTMENT SUPERVISOR

Professor Assistant Professor

Information Technology, Information Technology,

Sri Krishna College of Engineering and Sri Krishna College of Engineering and Technology, Technology,

Kuniyamuthur, Coimbatore–641008. Kuniyamuthur, Coimbatore–641008**.**

Submitted for the Project viva-voce examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

At this juncture, we take the opportunity to convey our sincere thanks and gratitude to the management of the college for providing all the facilities to us.

We wish to convey our gratitude to our college principal**, Dr. S. Sophia** for forwarding usto do our project and offering adequate duration to complete our project.

We would like to express our grateful thanks to **Dr. N. Susila** Head of the department, Department of Information Technology for her encouragement and valuable guidance to this project.

We extend my gratitude to our beloved guide **Ms. R. Janani, ME.**, **MBA., (PhD).,** Assistant Professor, Department of Information Technology for her constant support and immense help at all stagesof the project.

# ABSTRACT

The aging global population has created a growing demand for healthcare solutions tailored to the unique needs of senior citizens. Many seniors prefer to receive care in their homes, highlighting the need for efficient coordination and delivery of home care services. However, the existing healthcare infrastructure lacks a dedicated platform to connect seniors, caregivers, and healthcare providers seamlessly. This project aims to address these challenges by developing a user-friendly and secure web-based application. The application will provide a centralized system to efficiently match seniors with suitable healthcare providers, streamline appointment scheduling, facilitate real-time communication, and coordinate services effectively. The core feature of this system is to overcome the miscommunication between seniors and care providers by providing suitable services and booking schedules based on the user's necessity. By providing a dedicated platform for coordinating and delivering home care services for seniors, this project aims to enhance the quality of care and ultimately improve the quality of life for seniors receiving care at home.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE**  **No.** |
|  | **ACKNOWLEDGEMENT ABSTRACT**  **LIST OF TABLES LIST OF FIGURES**  **LIST OF ABBREVIATIONS** | **iii iv vii viii ix** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 COMPONENTS OF SYSTEM | 3 |
|  | 1.3 ADVANCED TECHNOLOGIES | 4 |
|  | 1.4 GLOBAL PERSPECTIVES | 6 |
| **2** | **SYSTEM ANALYSIS** | **7** |
|  | 2.1 EXISTING SYSTEM | 7 |
|  | 2.1.1 DRAWBACKS | 8 |
|  | 2.2 PROBLEM DEFINITION | 8 |
|  | 2.3 PROPOSED SYSTEM | 9 |
|  | 2.3.1 ADVANTAGES | 9 |
| **3** | **SYSTEM REQUIREMENTS** | **10** |
|  | 3.1 HARDWARE REQUIREMENTS | 10 |
|  | 3.2 SOFTWARE REQUIREMENTS | 10 |
|  | 3.3 SOFTWARE DESCRIPTION | 10 |
|  | 3.3.1 FRONTEND | 10 |
|  | 3.3.2 BACKEND | 13 |

1. SYSTEM DESIGN 15
   1. [MODULE DESCRIPTION 15](#_TOC_250016)
      1. [PROVIDERS MANAGEMENT 15](#_TOC_250015)
      2. [SERVICE MANAGEMENT 16](#_TOC_250014)
      3. [BOOKING MANAGEMENT 17](#_TOC_250013)
      4. [USER PROFILE MANAGEMENT 18](#_TOC_250012)
   2. [USE CASE DIAGRAM 19](#_TOC_250011)
   3. [SEQUENCE DIAGRAM 19](#_TOC_250010)
   4. [DATA FLOW DIAGRAM 20](#_TOC_250009)
2. TESTING 21
   1. [UNIT TESTING 21](#_TOC_250008)
   2. [INTEGRATION TESTING 21](#_TOC_250007)
   3. [SECURITY AND AUTHENTICATION](#_TOC_250006)
   4. [TEST CASES 23](#_TOC_250005)
      1. [TEST CASE I 23](#_TOC_250004)
      2. [TEST CASE II 24](#_TOC_250003)
3. CONCLUSION AND FUTURE WORK 25
   1. [CONCLUSION 25](#_TOC_250002)
   2. [FUTURE WORK 25](#_TOC_250001)
4. APPENDICES 26

APPENDIX I

SOURCE CODE 26

APPENDIX II

SCREENSHOTS 32

[REFERENCES 39](#_TOC_250000)

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE** |
|  |  | **No.** |
| 4.1 | Providers Management | 15 |
| 4.2 | Service Management | 16 |
| 4.3 | Bookings Management | 17 |
| 4.4 | User Profile Management | 18 |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE No.** | **TITLE** | **PAGE** |
|  |  | **No.** |
| 3.1 | VS Code Logo | 10 |
| 4.1 | Use Case Diagram | 17 |
| 4.2 | Sequence Diagram | 18 |
| 4.3 | Data Flow Diagram | 18 |
| 5.1 | Storing the token in local storage | 22 |
| 5.2 | Authenticating the user using |  |
|  | bearer token | 22 |
| 5.3 | Test case I | 23 |
| 5.4 | Test case II | 24 |
| A.2.1 | Main page | 32 |
| A.2.2 | Admin login | 32 |
| A.2.3 | Admin landing page | 33 |
| A.2.4 | Admin Dashboard | 33 |
| A.2.5 | Providers Management | 34 |
| A.2.6 | Service Management | 34 |
| A.2.7 | User Login Page | 35 |
| A.2.8 | User Registration Page | 35 |
| A.2.9 | User Landing Page | 36 |
| A.2.10 | User Dashboard | 36 |
| A.2.11 | Booking Management | 37 |
| A.2.12 | Managing the status of booking | 37 |
| A.2.13 | User Profile Management | 38 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **ABBREVIATIONS** | **EXPANSION** |
| 1 | ADL | Activities of Daily Living |
| 2 | RAM | Random Access Memory |
| 3 | GB | Giga Bytes |
| 4 | VS | Visual Studio |
| 5 | OS | Operating System |
| 6 | HTTP | Hyper Text Transfer Protocol |
| 7 | JPA | Java Persistence API |
| 8 | API | Application Programming |
|  |  | Interface |
| 9 | JDBC | Java Database Connectivity |
| 10 | SQL | Sequential Query Language |
| 11 | UI | User Interface |
| 12 | DOM | Document Object Model |
| 13 | JSX | Java Script XML |
| 14 | JWT | JSON Web Token |
| 15 | UML | Unified Modelling Language |
| 16 | DFD | Data Flow Diagram |
| 17 | FAQ | Frequently Asked Questions |

# OVERVIEW

**CHAPTER 1 INTRODUCTION**

As self-directed learning gains popularity, it presents a transformative

opportunity for individuals to pursue knowledge and skills independently. However, this approach introduces several challenges that traditional educational resources often fail to address effectively. Self-learners frequently encounter issues such as a lack of interactive engagement, insufficient feedback, and difficulties in maintaining motivation. These challenges can lead to gaps in understanding and hinder progress, highlighting the need for a more integrated and supportive solution.

To address these needs, the MY TUTOR project has been developed to provide a comprehensive and user-centric app designed specifically for self-learners. This innovative application aims to revolutionize the self-learning experience by offering a centralized platform that combines several advanced educational tools into one cohesive system. The primary goal of MY TUTOR is to bridge the gap between traditional educational methods and the evolving needs of modern learners, providing a seamless and engaging learning experience.

At the heart of MY TUTOR are its core features: summarization, interactive quizzes, and customizable flashcards. The summarization tool is designed to distill complex topics into clear and concise summaries, allowing users to quickly grasp essential concepts and review critical information without feeling overwhelmed. This feature is especially valuable for learners who need to process large amounts of material efficiently, providing a streamlined approach to understanding key points.

The interactive quizzes feature of MY TUTOR offers a dynamic way to assess and reinforce learning. These quizzes are tailored to the user’s knowledge level and adapt to their progress, providing immediate feedback to help identify strengths and areas for improvement. This real-time feedback mechanism ensures that learners can continuously track their understanding and adjust their study strategies accordingly, maintaining engagement and motivation throughout their learning journey.

In addition to summarization and quizzes, MY TUTOR includes customizable flashcards, which serve as a flexible tool for memorization and review. Users can create and personalize flashcards based on their study needs, allowing for targeted revision and ongoing reinforcement of key concepts. This feature supports a personalized learning approach, enabling users to focus on areas where they need the most practice.

By integrating these features into a single platform, MY TUTOR addresses several critical challenges faced by self-learners. It enhances learning efficiency by streamlining study sessions and providing interactive content that keeps users engaged. The app’s personalized feedback and adaptive quizzes ensure that learners receive tailored support and can track their progress effectively. Moreover, the customizable flashcards offer a convenient way to reinforce knowledge and prepare for assessments.

The MY TUTOR project aims to set a new standard in self-directed learning by offering a solution that is both effective and user-friendly. By addressing the limitations of traditional educational tools and providing a comprehensive set of features, MY TUTOR seeks to empower individuals to achieve their educational goals more efficiently and enjoyably. This project not only improves the self-learning experience but also fosters greater independence and confidence among learners, ultimately contributing to their overall success and satisfaction.

In summary, MY TUTOR is poised to transform the landscape of self-directed learning by offering a robust and integrated platform that meets the diverse needs of modern learners. Through its innovative features and user-centric design, MY TUTOR aims to enhance educational outcomes, support effective study practices, and promote a more engaging and personalized learning experience for users around the world.

# COMPONENTS OF SYSTEM

MY TUTOR consists of several key components, each contributing to the overall functionality and effectiveness of the application. These components are carefully designed to meet the diverse needs of self-learners and ensure a seamless user experience.

Home: The Home component serves as the central hub of the application. It provides users with an overview of their study progress, quick access to different features, and a summary of recent activities. The Home page is designed to be user-friendly, allowing users to easily navigate between different sections of the app.

Sign In: The Sign In component is crucial for user authentication and personalization. It allows users to log in to their accounts or create new ones, ensuring that their data and progress are securely stored. The Sign In feature includes options for both traditional email/password authentication and social media logins, offering flexibility and convenience.

Summary: This component provides users with concise and clear summaries of their study materials. By distilling complex topics into manageable chunks, the Summary feature helps learners quickly understand and retain essential information. This component is particularly useful for reviewing large volumes of material in a short period.

Flashcard: The Flashcard component is designed to enhance memorization and revision. Users can create custom flashcards with key terms and concepts, which can be reviewed at any time. The interactive nature of flashcards makes them an effective tool for reinforcing learning and testing knowledge on the go.

Quiz: The Quiz component offers interactive assessments to evaluate users’ understanding of the study material. Quizzes are designed to adapt to the user’s level of knowledge, providing a personalized learning experience. Immediate feedback helps users identify areas for improvement and track their progress. Recent Files: The Recent Files component allows users to manage and access recently viewed or created documents and study materials. This feature helps users quickly locate important files and continue their studies without having to navigate through multiple folders.

Each component of MY TUTOR is integrated to work seamlessly with the others, creating a cohesive learning experience that addresses the various needs of self-learners.

# ADVANCED TECHNOLOGIES

MY TUTOR integrates several cutting-edge technologies to deliver its features and ensure an optimal user experience. Among these technologies, the Gemini API plays a pivotal role in processing uploaded files, enhancing the application's capabilities.

Gemini API Integration:

A central component of MY TUTOR's functionality is its use of the Gemini API for processing uploaded files. The Gemini API is a powerful tool designed to handle various types of data inputs, including text documents, spreadsheets, and other file formats commonly used in educational contexts. Here’s how the Gemini API contributes to MY TUTOR:

File Parsing and Analysis: The Gemini API enables MY TUTOR to efficiently parse and analyze the contents of uploaded files. This includes extracting relevant information, organizing data, and converting it into a format suitable for summarization and other educational purposes. By leveraging advanced algorithms, the API ensures that the data is processed accurately and promptly.

Text Summarization: One of the key features of MY TUTOR is its ability to provide concise summaries of complex texts. The Gemini API supports this functionality by analyzing the content of uploaded documents and generating summaries that highlight key points and essential information. This helps users quickly grasp important concepts without having to sift through lengthy documents.

Data Extraction: For files containing structured data, such as spreadsheets, the Gemini API facilitates the extraction and organization of data. This enables MY TUTOR to present information in a user-friendly format, such as interactive tables or charts, enhancing the learning experience by making data more accessible and understandable.

Compatibility and Flexibility: The Gemini API is designed to handle a wide range of file formats and types. This compatibility ensures that users can upload and process various types of educational materials, including text files, PDFs, and more. The flexibility of the API allows MY TUTOR to accommodate diverse user needs and preferences.

Efficiency and Scalability: With the Gemini API, MY TUTOR can process large volumes of data efficiently. The API’s scalable architecture ensures that the application can handle an increasing number of users and file uploads without compromising performance. This scalability is crucial for maintaining a smooth and responsive user experience as the app grows

Responsive Design:

MY TUTOR is designed with responsive design principles to ensure that the application is accessible and functional across various devices and screen sizes. Whether users are accessing the app on a desktop, tablet, or smartphone, the responsive design ensures a seamless and intuitive user experience.

By integrating these advanced technologies, including the Gemini API, MY TUTOR offers a sophisticated and efficient learning platform that meets the diverse needs of self-learners and enhances their educational experience.

# GLOBAL PERSPECTIVES

MY TUTOR is positioned to address educational challenges on a global scale by aligning with international trends in e-learning, self-directed learning, and technological advancements. Here’s how MY TUTOR fits into the global educational landscape:

1. Bridging the Education Gap

In many regions worldwide, access to quality education remains a significant challenge. MY TUTOR helps bridge this gap by providing a platform that offers educational resources and tools to individuals regardless of their geographical location. By enabling self-learning through summarization, quizzes, and flashcards, MY TUTOR empowers users who may not have access to traditional educational resources or formal instruction.

2. Supporting Lifelong Learning

The concept of lifelong learning is increasingly emphasized across the globe. As the job market evolves and new skills are required, individuals are seeking flexible and efficient ways to continue their education. MY TUTOR caters to this need by offering a platform that supports ongoing learning. Users can revisit and update their knowledge, explore new subjects, and engage with interactive content, aligning with the global push for continuous personal and professional development. 3. Enhancing Self-Learning Experiences

Self-learning is gaining popularity as a method of education due to its flexibility and accessibility. MY TUTOR enhances self-learning experiences by providing tools that make studying more engaging and effective. Through features like interactive quizzes and customizable flashcards, MY TUTOR supports diverse learning styles and helps users stay motivated and on track with their educational goals.

4. Leveraging Technological Advancements

The integration of advanced technologies such as the Gemini API and Natural Language Processing (NLP) reflects a global trend towards leveraging technology to improve educational outcomes. By incorporating these technologies, MY TUTOR stays at the forefront of educational innovation, offering users sophisticated tools for processing information and personalizing their learning experience. This approach aligns with the global shift towards technology-driven education solutions.

5. Addressing Diverse Educational Needs

Educational needs vary widely across different cultures and regions. MY TUTOR is designed to be adaptable and responsive to diverse user requirements. The ability to process various types of educational materials and provide personalized feedback ensures that the platform can cater to users from different educational backgrounds and levels. This adaptability helps MY TUTOR serve a global audience, offering tailored solutions that meet the specific needs of individual learners. 6. Promoting Inclusivity and Accessibility

Inclusivity and accessibility are critical considerations in the global educational landscape. MY TUTOR is committed to providing an inclusive learning environment by ensuring that its features are accessible to all users, including those with disabilities. The platform’s responsive design and user-friendly interface contribute to its accessibility, allowing users with diverse needs to benefit from its educational tools.

7. Encouraging Collaboration and Community

While MY TUTOR primarily focuses on self-learning, it also fosters a sense of community and collaboration through its features. Users can share their progress, compare quiz results, and engage with others through the platform. This aspect of MY TUTOR supports the global trend towards collaborative learning and peer interaction, even in a self-directed learning environment.

8. Supporting Global Competitiveness

In a globalized world, the ability to acquire and apply knowledge effectively is crucial for maintaining competitiveness. MY TUTOR helps users develop their skills and knowledge in a way that is relevant to global standards and practices. By providing high-quality educational tools and resources, MY TUTOR contributes to the global effort of preparing individuals for the challenges of the modern workforce.

In summary, MY TUTOR aligns with global educational trends and needs by offering a flexible, technology-driven learning platform that supports lifelong learning, addresses diverse educational needs, and promotes inclusivity. By bridging educational gaps and enhancing self-learning experiences, MY TUTOR contributes to the advancement of education on a global scale.

# EXISTING SYSTEM

**CHAPTER 2 SYSTEM ANALYSIS**

Overview

Before the advent of sophisticated educational technology platforms like MY TUTOR, self-learners primarily relied on traditional methods of study which included textbooks, handwritten notes, and offline study aids. These methods, while foundational, have several limitations in addressing the dynamic needs of modern learners.

Traditional Learning Methods

Traditional learning systems involve the use of physical resources such as textbooks and printed notes. These methods are often static, providing a fixed amount of information that may not be updated or tailored to individual learning needs. Students typically engage in self-study by reading and reviewing these materials, but this approach lacks interactive elements and immediate feedback mechanisms.

Digital Learning Tools

With the rise of digital technology, various educational tools have been developed, including online courses, educational apps, and e-books. While these tools offer more flexibility compared to traditional methods, they often suffer from issues such as lack of integration, limited interactivity, and the absence of personalized learning paths. Many digital platforms provide generic content without considering individual learning preferences or progress.

Limitations of Existing Systems

Lack of Personalization: Existing systems often fail to adapt to individual learning styles and preferences. Users receive the same content regardless of their prior knowledge or learning pace.

Limited Interactivity: Many digital learning tools lack interactive features that engage users and enhance their learning experience. Traditional methods and even some digital tools do not offer real-time feedback or adaptive learning experiences.

Static Content: Traditional educational resources are static and may become outdated over time. Users cannot easily update or customize the content according to their evolving needs.

Fragmented Experience: Users often have to switch between multiple platforms or tools to get a comprehensive learning experience. This fragmented approach can lead to inefficiencies and a lack of coherence in the learning process.

# DRAWBACKS

Identified Issues in Existing Systems

1. Limited Feedback Mechanisms

Existing systems generally provide minimal feedback to learners, which is crucial for understanding and improving performance. Without immediate feedback, learners may not be aware of their mistakes or areas that need improvement, hindering their learning progress.

2. Low Engagement and Motivation

Traditional methods and many digital tools lack engaging features that can maintain a learner's interest. The absence of interactive elements such as quizzes, games, or dynamic content often results in lower motivation and less effective learning outcomes.

3. Inefficient Study Processes

Manual methods of studying, such as note-taking and reviewing textbooks, can be time-consuming and inefficient. Learners often spend excessive time organizing and summarizing information, which could be streamlined with the right tools.

4. Accessibility Issues

Many existing educational resources are not accessible to all learners, particularly those with disabilities or those who lack access to the latest technology. This can create barriers to effective learning and limit opportunities for a diverse range of users.

5. Lack of Integration

Educational tools are often not integrated with one another, leading to a disjointed learning experience. For example, users may need to use separate tools for summarization, quizzes, and flashcards, which can be cumbersome and inefficient.

# PROBLEM DEFINITION

Current Challenges in Self-Learning

1. Fragmented Learning Experience

The current educational landscape is characterized by a fragmented approach to self-learning, where students must piece together various resources and tools to create a comprehensive study plan. This fragmentation can lead to inefficiencies and a lack of coherence in the learning process.

2. Inadequate Feedback and Assessment

Without access to immediate feedback and assessment tools, learners struggle to gauge their understanding and progress. The absence of real-time evaluations can result in prolonged gaps in knowledge and hinder overall learning effectiveness.

3. Limited Engagement and Motivation

Traditional and many digital learning methods fail to keep learners engaged and motivated. The lack of interactive and dynamic content often leads to disengagement and decreased motivation to continue studying.

4. Personalization Deficit

Existing systems often provide a one-size-fits-all approach to education, which does not cater to the individual needs and learning styles of users. This lack of personalization can make it challenging for learners to effectively grasp and retain information. 5. Accessibility and Inclusivity Issues

Many educational tools and resources are not accessible to all learners, particularly those with disabilities or those who lack advanced technological resources. This creates barriers to learning and limits the effectiveness of educational initiatives.

# 2.3 Proposed System

Overview of MY TUTOR

MY TUTOR is designed to address the limitations of existing educational systems by offering a comprehensive, integrated platform that enhances the self-learning experience. The system provides tools and features that cater to diverse learning needs and preferences, ensuring a more effective and engaging study process.

1. Integrated Learning Platform

MY TUTOR combines summarization, quizzes, flashcards, and other educational tools into a single platform. This integration streamlines the learning process, making it easier for users to access and manage their study materials.

2. Personalized Learning Experience

The platform uses advanced technologies to tailor the learning experience to individual needs. Users receive personalized content and feedback based on their progress and performance, ensuring a more effective and relevant study experience.

3. Interactive Features

MY TUTOR incorporates interactive elements such as quizzes and customizable flashcards to enhance engagement and motivation. These features make learning more dynamic and enjoyable, helping users stay committed to their educational goals.

4. Real-Time Feedback and Assessment

Immediate feedback on quizzes and assessments allows users to quickly identify and address areas of improvement. This real-time feedback mechanism supports continuous learning and helps users track their progress effectively.

5. Accessibility and Inclusivity

MY TUTOR is designed to be accessible to a wide range of users, including those with disabilities. The platform’s responsive design and user-friendly interface ensure that all learners can benefit from its features, regardless of their technological resources.

2.3.1 Advantages

Benefits of Implementing MY TUTOR

1. Enhanced Learning Efficiency

By integrating various educational tools into a single platform, MY TUTOR improves learning efficiency. Users can quickly access and utilize different features, saving time and streamlining their study process.

2. Increased Engagement

The interactive features of MY TUTOR, including quizzes and flashcards, help maintain user engagement and motivation. These tools make learning more enjoyable and effective, leading to better educational outcomes.

3. Personalized Learning Paths

MY TUTOR’s use of advanced technologies to personalize the learning experience ensures that users receive content and feedback tailored to their individual needs. This personalized approach enhances the effectiveness of the learning process and supports better knowledge retention.

4. Real-Time Feedback

Immediate feedback on assessments allows users to quickly identify and address their weaknesses. This real-time feedback mechanism supports continuous improvement and helps users achieve their learning goals more effectively.

5. Accessible and Inclusive Design

The platform’s focus on accessibility ensures that all users, including those with disabilities, can benefit from its features. MY TUTOR’s inclusive design promotes equal learning opportunities for a diverse range of users.

6. Flexibility and Adaptability

MY TUTOR’s flexible and adaptable features cater to various learning styles and preferences. Users can customize their study materials and learning paths to suit their individual needs, making the platform a versatile tool for self-learners.

7. Global Applicability

The platform’s ability to cater to diverse educational needs and preferences makes it suitable for users worldwide. MY TUTOR’s global applicability ensures that learners from different regions and backgrounds can benefit from its features and resources.

# CHAPTER 3 SYSTEM REQUIREMENTS

The hardware and software requirements and also the platform description of the system are explained under sections 3.1, 3.2 and 3.3 respectively.

# HARDWARE REQUIREMENTS

Minimum Hardware Specifications

Server Requirements:

CPU: Quad-core processor or higher (e.g., Intel Core i5 or AMD Ryzen 5)

RAM: 8 GB or more

Storage: SSD with at least 100 GB of available space

Network: Reliable internet connection with minimum bandwidth of 1 Mbps

Operating System: Linux (Ubuntu or CentOS preferred) or Windows Server 2019

Client Requirements:

CPU: Dual-core processor (e.g., Intel Core i3 or AMD Ryzen 3)

RAM: 4 GB or more

Storage: Minimum 20 GB of free space

Network: Stable internet connection with at least 512 Kbps download speed

Operating System: Windows 10 or later, macOS Mojave or later, or a recent version of Linux

Additional Hardware Considerations

Backup Storage: External or cloud-based backup solutions to ensure data redundancy and recovery.

Development Environment: Development machines with higher specifications may be required for testing and development purposes, such as 16 GB RAM and multi-core processors

# SOFTWARE REQUIREMENTS

Server-Side Software

Operating System:

Linux (e.g., Ubuntu 20.04 LTS) or Windows Server 2019 for server environments.

Database:

MySQL: For managing relational data, ensuring robust data storage and retrieval capabilities.

Django: As the ORM (Object-Relational Mapping) tool for interacting with MySQL and managing database schemas. Backend Frameworks and Libraries:

Python: Primary programming language for server-side logic and API development.

FastAPI: For building high-performance RESTful APIs with Python, ensuring rapid and efficient communication between the frontend and backend.

Spring Boot: For building and managing microservices, enhancing modularity and scalability of the backend system.

Web Server:

Nginx or Apache: For handling HTTP requests and serving the application to users efficiently.

Containerization:

Docker: For creating, deploying, and running applications in containers, ensuring consistency across different environments.

Client-Side Software

Frontend Framework:

React: JavaScript library for building dynamic and responsive user interfaces, ensuring a seamless user experience.

Web Browser:

Modern web browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge for accessing the MY TUTOR application.

Development Tools

IDE/Editor:

VS Code: Recommended for coding in JavaScript (React) and Python.

PyCharm: For Python development, particularly with Django and FastAPI.

Version Control:

Git: For source code management and collaboration.

Build Tools:

Webpack or Vite: For bundling and optimizing frontend assets.

# SOFTWARE DESCRIPTION

Overview

This section provides detailed descriptions of the software components used in the MY TUTOR project, including frontend and backend technologies, and their specific roles in the system.

# FRONTEND

React

Purpose: React is used to build the user interface of MY TUTOR. It allows for the creation of dynamic and interactive web pages by leveraging reusable components.

Features:

Component-Based Architecture: Facilitates the development of modular and maintainable code by breaking down the user interface into smaller, reusable components.

Virtual DOM: Enhances performance by minimizing direct manipulation of the real DOM, resulting in faster rendering and updates.

State Management: Handles complex state management within the application using hooks or state management libraries like Redux.

Key Advantages:

Responsiveness: Ensures that the application is responsive and works well on various devices, including desktops, tablets, and smartphones.

Developer Efficiency: Streamlines the development process with tools and libraries that improve productivity and code maintainability.

# BACKEND

Python

Purpose: Python serves as the primary programming language for backend development, providing a flexible and powerful environment for server-side logic.

Features:

Ease of Learning and Use: Python’s simplicity and readability make it an ideal choice for backend development.

Rich Ecosystem: Extensive libraries and frameworks available for various backend needs.

FastAPI

Purpose: FastAPI is used to build and manage RESTful APIs, enabling efficient communication between the frontend and backend.

Features:

Performance: Provides high-performance API endpoints with minimal latency.

Automatic Documentation: Generates interactive API documentation automatically, enhancing developer productivity.

Spring Boot

Purpose: Spring Boot is utilized for developing microservices that handle specific functionalities within the backend.

Features:

Microservice Architecture: Facilitates the creation of modular and scalable backend systems.

Integration: Seamlessly integrates with other Java-based technologies and services.

Django

Purpose: Django is used as an ORM for interacting with the MySQL database, managing database schemas, and handling data migrations.

Features:

Built-in Admin Interface: Provides a ready-to-use interface for managing database records and user interactions.

Robust Security: Includes built-in security features to protect against common vulnerabilities.

MySQL

Purpose: MySQL is the relational database management system used to store and manage data for the MY TUTOR application.

Features:

Reliability: Known for its robustness and reliability in handling large volumes of data.

Scalability: Supports scaling to accommodate growing data needs.

# CHAPTER 4 SYSTEM DESIGN

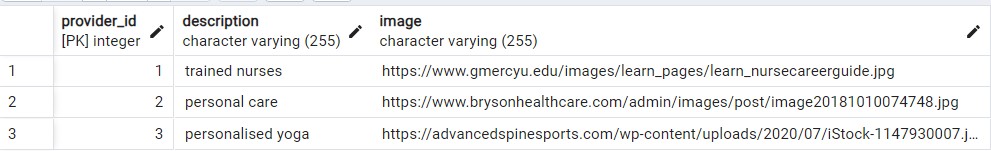
# MODULE DESCRIPTION

* + - Providers management
    - Service management
    - Booking management
    - User profile management

# PROVIDERS MANAGEMENT

The admin manages service providers by adding, editing, and deleting their profiles. This includes managing qualifications, availability, and contact information. This ensures that users have access to accurate and up-to-date information about service providers.

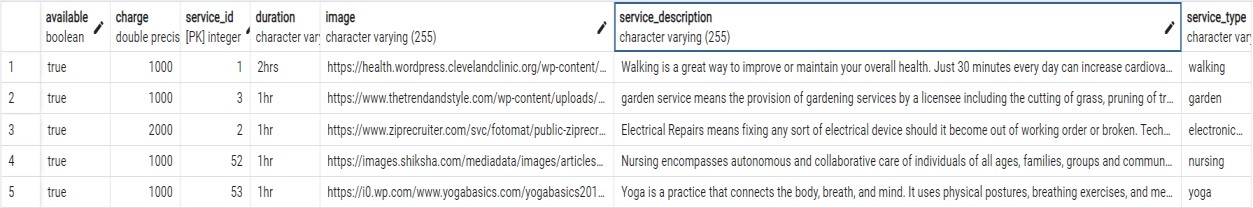
# Table 4.1. Providers Management



# SERVICE MANAGEMENT

The service management system organizes and controls various aspects of services to ensure efficient delivery and customer satisfaction. These aspects include specifying the service type, describing the service, setting the charge, defining the duration, and indicating service availability. Efficient management of these aspects helps ensure smooth service delivery, meets customer expectations, and maintains competitive pricing.

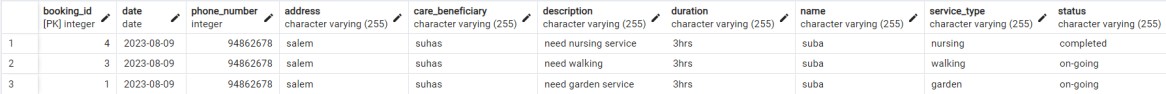
# Table 4.2. Service Management



# BOOKING MANAGEMENT

In the booking management, various details are managed to facilitate the scheduling and tracking of services. These details include the user's name and mobile number for identification, service type, date and timings for scheduling. Additionally, information about the care beneficiary, address, and booking status (which is managed by the admin) is also tracked. Efficient management of these details helps ensure that bookings are accurately scheduled, communicated, and executed, leading to improved customer service and satisfaction.

# Table 4.3 Booking Management



# USER PROFILE MANAGEMENT

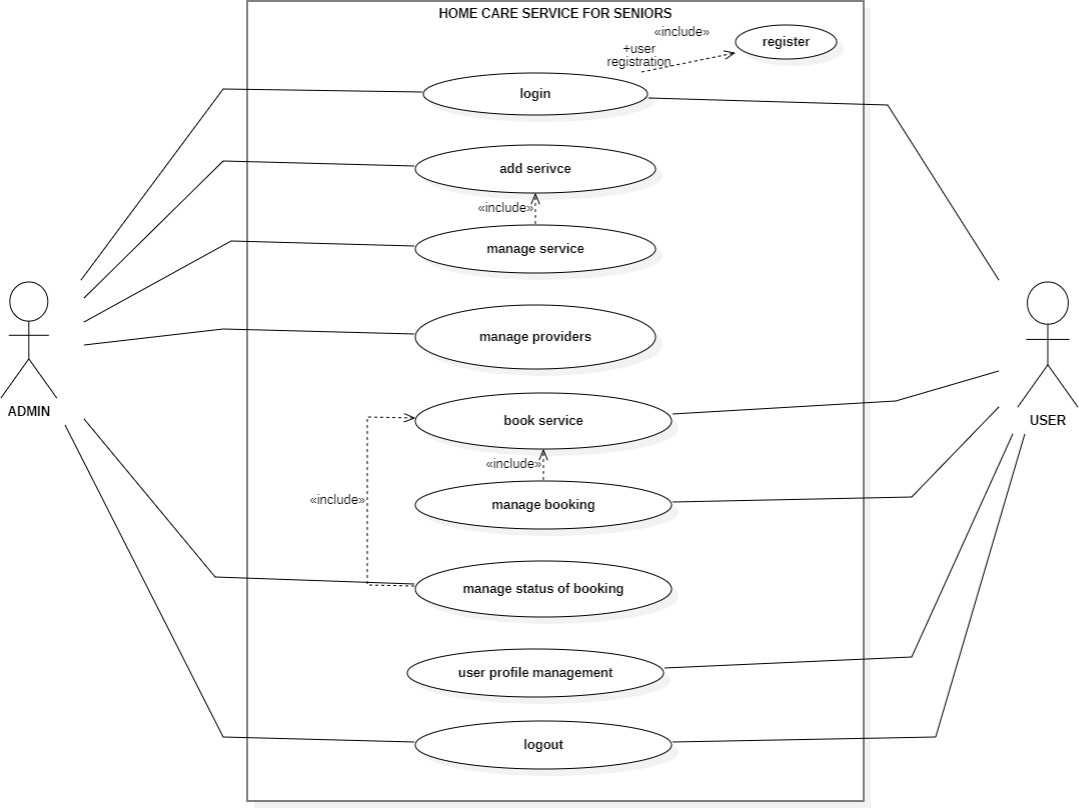
In the user profile management system, users can manage their details such as their name, email ID, and mobile number. However, while users can update their mobile number, they cannot edit their name and email ID. This setup ensures that critical user information remains accurate and secure, while still allowing users some control over their profiles.

# Table 4.4. User Profile Management



# USE CASE DIAGRAM

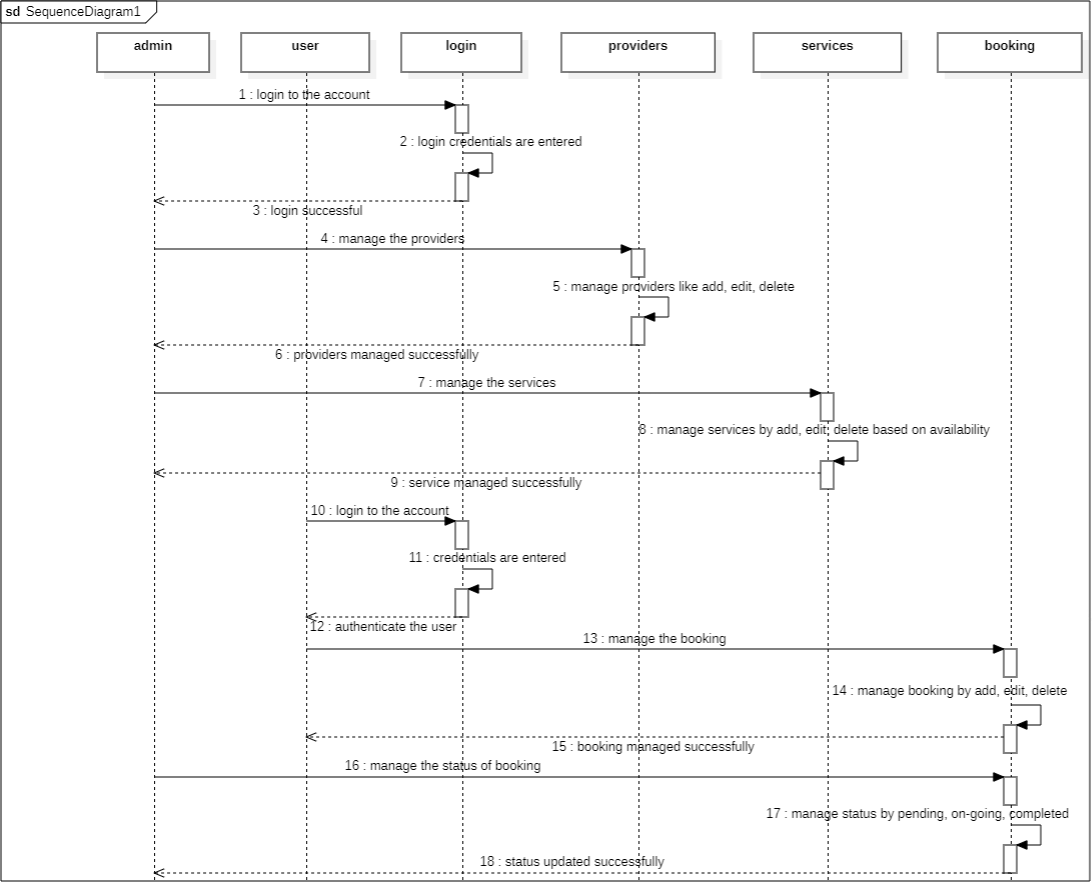
A use case diagram is a visual representation in UML (Unified Modelling Language) that illustrates the interactions between actors and a system or software application. It is used to depict the various ways users or external entities can interact with the system and the specific functionalities or use cases it offers.



# Fig. 4.1. Use Case Diagram

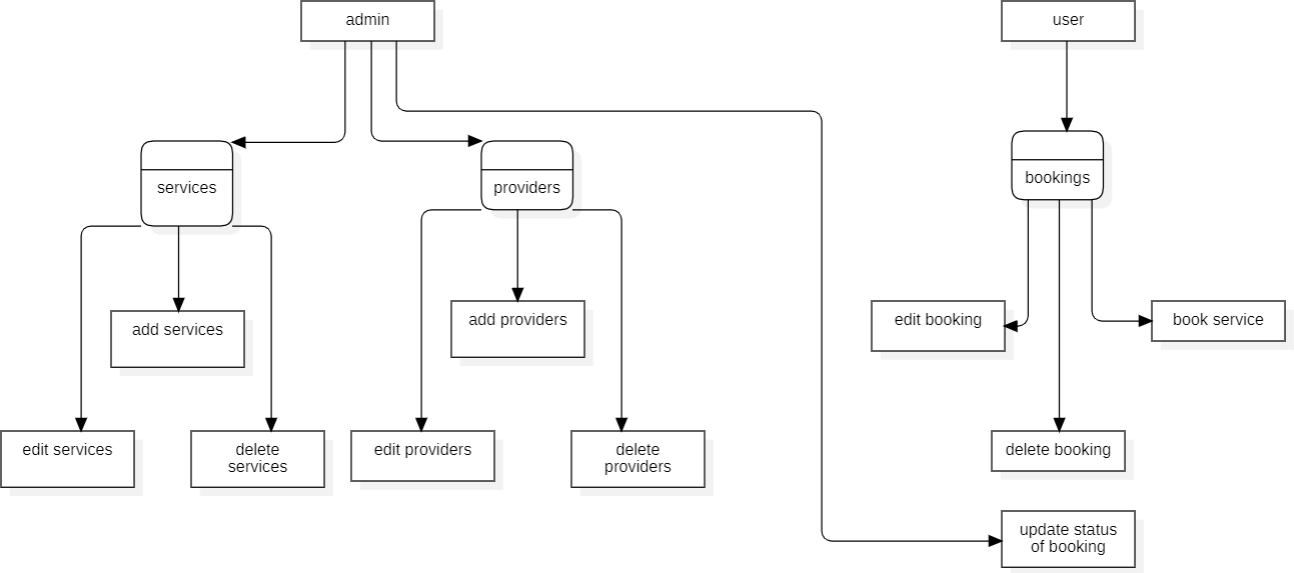
# SEQUENCE DIAGRAM

A sequence diagram is a visual representation used in software engineering to illustrate the interactions and communication between different objects or components in a system over a specific period of time. It shows the chronological order of messages or method calls exchanged between these entities, helping to depict the dynamic behaviour of a system or a particular scenario. In essence, it provides a time-ordered view of how various parts of a system collaborate to achieve a particular task or functionality.



# Fig. 4.2. Sequence Diagram

# DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a visual representation that illustrates the flow of data within a system or process. It uses symbols to depict processes, data stores, data flow, and external entities. DFD helps to analyse, design, and document information systems, showing how data is input, processed and output, while emphasizing the interactions between different elements in the system.

# Fig. 4.3. Data Flow Diagram

**CHAPTER 5 TESTING**

**5.TESTING**

Testing is a crucial phase in the development lifecycle of MY TUTOR, ensuring that the application performs as expected, is secure, and meets user requirements. This section provides an overview of the testing strategies employed, including unit testing, integration testing, security and authentication testing, and specific test cases.

# UNIT TESTING

Overview

Unit testing involves testing individual components or functions of the application in isolation. The goal is to validate that each unit of the application performs as expected and adheres to its specifications. For MY TUTOR, unit tests cover various aspects of the frontend and backend systems.

Frontend Unit Testing

Tools Used: Jest, React Testing Library

Components Tested:

Button Components: Ensure that buttons, such as "Start Read Aloud" and "Submit," respond correctly to user interactions.

Form Validation: Verify that form inputs for sign-up and login are validated properly and handle edge cases.

Rendering: Check that components render correctly with different states and props.

Backend Unit Testing

Tools Used: Pytest for Python, Django Test Framework

Components Tested:

API Endpoints: Validate that endpoints for user registration, login, and content retrieval return the correct responses and handle errors appropriately.

Business Logic: Test the core logic for content processing and quiz generation to ensure accuracy.

Database Operations: Verify that database queries and updates work as expected without causing data corruption.

Examples:

Button Click Test: Verify that clicking the "Start Read Aloud" button initiates the correct action.

Form Submission Test: Check that submitting the sign-up form creates a new user in the database.

# INTEGRATION TESTING

Overview

Integration testing focuses on the interactions between different modules or components of the system. It ensures that these components work together as intended and that data flows correctly between them. For MY TUTOR, integration tests validate the interaction between the frontend, backend, and third-party services.

Frontend and Backend Integration Testing

Tools Used: Postman for API testing, Cypress for end-to-end testing

Scenarios Tested:

Sign-Up Process: Test the complete flow from form submission on the frontend to user creation in the backend.

Quiz Generation: Verify that the frontend can request a quiz, receive it from the backend, and display it correctly to the user.

Content Processing: Ensure that files uploaded via the frontend are processed by the backend and that the results are returned accurately.

Examples:

End-to-End Sign-Up Test: Simulate a user signing up, logging in, and accessing the main features of MY TUTOR to ensure a seamless user experience.

Content Upload and Processing Test: Upload a file through the frontend, check that it is processed correctly by the backend, and confirm that the processed data is displayed as expected.

# SECURITY AND AUTHENTICATION

Overview

Security and authentication testing ensures that the application is protected against unauthorized access and data breaches. It verifies that user data is secure and that authentication mechanisms are robust.

Testing Areas:

Authentication: Validate that the authentication system works correctly, including password hashing, session management, and token validation.

Authorization: Ensure that users have appropriate access levels and that restricted resources are protected.

Data Encryption: Verify that sensitive data, such as user credentials and personal information, is encrypted both at rest and in transit.

Vulnerability Assessment: Perform penetration testing to identify potential security vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Examples:

Login Attempt with Invalid Credentials: Check that users cannot log in with incorrect passwords or usernames.

Access Control Test: Verify that users cannot access restricted areas or perform actions beyond their permissions.

# TEST CASES

Overview

Test cases are specific scenarios used to validate the functionality and performance of the application. They provide a structured approach to testing and help ensure that all aspects of the system are thoroughly evaluated.

# TEST CASE I

Objective: Verify that the user registration process works correctly.

Preconditions: User is on the registration page.

Steps:

Enter valid user details (username, email, name, password).

Click the "Sign Up" button.

Expected Result: User is successfully registered, and a confirmation message is displayed. User data is stored in the database.

# TEST CASE II

Objective: Ensure that quiz responses are processed and scored correctly.

Preconditions: User is logged in and has access to a quiz.

Steps:

Answer all questions in the quiz.

Submit the quiz.

Expected Result: The quiz is submitted, and the user receives a score with feedback on their performance.

# CHAPTER 6

**CONCLUSION AND FUTURE WORK**

# CONCLUSION

MY TUTOR is a cutting-edge educational application designed to enhance the self-learning experience for students. Through a suite of innovative features, including summarization tools, interactive quizzes, and customizable flashcards, MY TUTOR addresses key challenges faced by self-learners, such as lack of immediate feedback, engagement issues, and limited interaction.

Key Achievements:

Enhanced Learning Efficiency: MY TUTOR’s summarization capabilities allow students to quickly grasp and review complex topics, making study sessions more effective. The interactive quizzes and flashcards provide a streamlined approach to reinforcing learning and tracking progress.

Increased Engagement: The application’s interactive elements, such as quizzes and flashcards, help maintain student motivation and engagement. The integration of these tools ensures that students stay actively involved in their learning journey.

Personalized Feedback: Immediate feedback from quizzes and progress tracking features help students identify their strengths and areas needing improvement. This personalized approach guides students towards achieving their educational goals more effectively.

Technological Integration: The use of advanced technologies such as the Gemini API for file processing enhances the app's capabilities, ensuring efficient content handling and user-friendly interactions.

Overall, MY TUTOR successfully provides a comprehensive platform for self-learners, improving their study habits and overall educational experience. The application combines innovative technology with user-centric design to deliver an effective learning tool that addresses common challenges in self-education.

# FUTURE WORK

As technology and user needs evolve, there are several opportunities for enhancing MY TUTOR to provide even more value to its users. One of the primary areas for future development includes:

Audio-to-Text Functionality for Practice Recitation:

Objective: Implementing audio-to-text functionality will allow users to practice recitation by converting their spoken responses into text. This feature will enable students to improve their pronunciation, fluency, and overall verbal skills.

Features:

Speech Recognition: Utilize advanced speech recognition technology to transcribe audio input into text accurately.

Real-Time Feedback: Provide immediate feedback on pronunciation and clarity to help students refine their speaking skills.

Practice Mode: Offer a practice mode where users can record their recitations and compare their spoken text with the original content to assess their performance.

Progress Tracking: Implement a system to track improvements in speech accuracy and fluency over time, offering personalized recommendations for further practice.

Integration: This feature will be integrated seamlessly with existing functionalities, such as quizzes and flashcards, to offer a holistic learning experience that includes both written and spoken practice.

By incorporating audio-to-text functionalities, MY TUTOR will expand its capabilities and offer a more comprehensive tool for language learning and verbal practice. This enhancement will address additional aspects of self-learning, further supporting students in achieving their educational objectives.

Other Future Enhancements:

Enhanced AI Features: Explore the integration of more advanced AI algorithms to provide personalized learning paths and adaptive content based on user performance.

Mobile Application: Develop a mobile version of MY TUTOR to offer greater accessibility and convenience for users on the go.

Collaborative Learning Tools: Introduce features that enable collaboration and interaction among users, such as study groups or peer review systems.

These future enhancements aim to build upon MY TUTOR’s current success, ensuring that it continues to meet the evolving needs of self-learners and provides a state-of-the-art educational experience.

# APPENDIX I SOURCE CODE

**Signup.jsx :**

**import { useNavigate } from 'react-router-dom';**

**import { useState } from 'react';**

**import axios from 'axios';**

**import signinImage from './undraw\_sign\_up\_n6im.svg';**

**import { GoogleLogin } from 'react-google-login';**

**import { MicrosoftLogin } from 'react-microsoft-login';**

**function Signin() {**

**const [isChecked, setIsChecked] = useState(false);**

**const [signupData, setSignupData] = useState({**

**username: '',**

**email: '',**

**name: '',**

**password: '',**

**termsAccepted: false,**

**});**

**const [loginData, setLoginData] = useState({**

**email: '',**

**password: '',**

**});**

**const navigate = useNavigate();**

**const toggleCheck = () => {**

**setIsChecked(!isChecked);**

**};**

**const handleSignupChange = (e) => {**

**const { name, value, type, checked } = e.target;**

**setSignupData((prevState) => ({**

**...prevState,**

**[name]: type === 'checkbox' ? checked : value,**

**}));**

**};**

**const handleLoginChange = (e) => {**

**const { name, value } = e.target;**

**setLoginData((prevState) => ({**

**...prevState,**

**[name]: value,**

**}));**

**};**

**const handleSignupSubmit = async (e) => {**

**e.preventDefault();**

**const { username, email, name, password, termsAccepted } = signupData;**

**if (username && email && name && password && termsAccepted) {**

**try {**

**await axios.post('http://localhost:8090/register', signupData);**

**alert('User Created');**

**toggleCheck();**

**} catch (error) {**

**alert('Error creating user');**

**}**

**} else if (!termsAccepted) {**

**alert('You must accept the terms and conditions');**

**} else {**

**alert('Please fill all the fields');**

**}**

**};**

**const handleLoginSubmit = async (e) => {**

**e.preventDefault();**

**const { email, password } = loginData;**

**if (email && password) {**

**try {**

**const response = await axios.get('http://localhost:8090/get');**

**const userExist = response.data.some(**

**(data) => data.email === email && data.password === password**

**);**

**if (userExist) {**

**alert('Login successful');**

**navigate('/home');**

**} else {**

**alert('User Not Found');**

**}**

**} catch (error) {**

**console.error('Error fetching users', error);**

**alert('Error logging in');**

**}**

**} else {**

**alert('Please fill all the fields');**

**}**

**};**

**const handleGoogleResponse = (response) => {**

**console.log('Google response', response);**

**// Handle Google login response here**

**};**

**const handleMicrosoftResponse = (response) => {**

**console.log('Microsoft response', response);**

**// Handle Microsoft login response here**

**};**

**return (**

**<div style={styles.page}>**

**<div style={styles.background}></div>**

**<div style={styles.left}>**

**<h1 style={styles.header}>**

**Welcome to <span style={styles.highlight}>MY TUTOR</span>! Your personal AI-based assistance.**

**</h1>**

**<img src={signinImage} alt="learning" style={styles.pageImage} />**

**</div>**

**<div style={styles.right}>**

**<div style={styles.authContainer}>**

**<div**

**style={!isChecked ? { ...styles.formContainer, ...styles.fadeIn } : styles.hidden}**

**>**

**<div style={styles.signup}>**

**<form onSubmit={handleSignupSubmit} style={styles.form}>**

**<label htmlFor="chk" aria-hidden="true" style={styles.formLabel}>**

**Sign up**

**</label>**

**<input**

**type="text"**

**name="username"**

**placeholder="User name"**

**required**

**style={styles.inputField}**

**onChange={handleSignupChange}**

**/>**

**<input**

**type="email"**

**name="email"**

**placeholder="Email"**

**required**

**style={styles.inputField}**

**onChange={handleSignupChange}**

**/>**

**<input**

**type="text"**

**name="name"**

**placeholder="Name"**

**required**

**style={styles.inputField}**

**onChange={handleSignupChange}**

**/>**

**<input**

**type="password"**

**name="password"**

**placeholder="Password"**

**required**

**style={styles.inputField}**

**onChange={handleSignupChange}**

**/>**

**<div style={styles.termsContainer}>**

**<input**

**type="checkbox"**

**name="termsAccepted"**

**required**

**style={styles.checkbox}**

**onChange={handleSignupChange}**

**/>**

**<label htmlFor="termsAccepted" style={styles.termsLabel}>**

**By clicking this you agree to our <a href="/privacy" style={styles.termsLink}>Privacy Policy</a>.**

**</label>**

**</div>**

**<button type="submit" style={styles.submitButton}>**

**Sign up**

**</button>**

**</form>**

**<div style={styles.bottomPrompt}>**

**<span style={styles.promptText}>Already a user?</span>**

**<button onClick={toggleCheck} style={styles.toggleButton}>**

**Login**

**</button>**

**</div>**

**</div>**

**</div>**

**<div**

**style={isChecked ? { ...styles.formContainer, ...styles.fadeIn } : styles.hidden}**

**>**

**<div style={styles.login}>**

**<form onSubmit={handleLoginSubmit} style={styles.form}>**

**<label htmlFor="chk" aria-hidden="true" style={styles.formLabel}>**

**Login**

**</label>**

**<input**

**type="email"**

**name="email"**

**placeholder="Email"**

**required**

**style={styles.inputField}**

**onChange={handleLoginChange}**

**/>**

**<input**

**type="password"**

**name="password"**

**placeholder="Password"**

**required**

**style={styles.inputField}**

**onChange={handleLoginChange}**

**/>**

**<button type="submit" style={styles.submitButton}>**

**Login**

**</button>**

**</form>**

**<div style={styles.bottomPrompt}>**

**<span style={styles.promptText}>New user?</span>**

**<button onClick={toggleCheck} style={styles.toggleButton}>**

**Sign up**

**</button>**

**</div>**

**</div>**

**</div>**

**<div style={styles.socialLoginContainer}>**

**<GoogleLogin**

**clientId='YOUR\_GOOGLE\_CLIENT\_ID'**

**buttonText="Login with Google"**

**onSuccess={handleGoogleResponse}**

**onFailure={handleGoogleResponse}**

**/>**

**</div>**

**</div>**

**</div>**

**</div>**

**);**

**}**

**// Inline styles**

**const styles = {**

**page: {**

**display: 'flex',**

**justifyContent: 'center',**

**alignItems: 'center',**

**height: '100vh',**

**overflow: 'hidden',**

**position: 'relative',**

**},**

**background: {**

**position: 'absolute',**

**top: 0,**

**left: 0,**

**width: '100%',**

**height: '100%',**

**background: 'linear-gradient(45deg, #1e1e1e 25%, #121212 75%)',**

**zIndex: -1,**

**overflow: 'hidden',**

**animation: 'backgroundAnimation 10s linear infinite',**

**},**

**left: {**

**flex: 1,**

**textAlign: 'center',**

**padding: '20px',**

**position: 'relative',**

**zIndex: 1,**

**},**

**right: {**

**flex: 1,**

**padding: '20px',**

**display: 'flex',**

**justifyContent: 'center',**

**alignItems: 'center',**

**flexDirection: 'column',**

**backgroundColor: '#1e1e1e',**

**borderRadius: '10px',**

**boxShadow: '0px 4px 6px rgba(0, 0, 0, 0.5)',**

**height: '80%',**

**position: 'relative',**

**zIndex: 1,**

**},**

**header: {**

**color: '#1e90ff',**

**marginBottom: '20px',**

**fontSize: '2rem',**

**fontWeight: 'bold',**

**animation: 'headerAnimation 2s ease-in-out infinite',**

**},**

**highlight: {**

**color: '#ff69b4',**

**},**

**pageImage: {**

**width: '80%',**

**height: 'auto',**

**maxWidth: '400px',**

**margin: '0 auto',**

**animation: 'imageAnimation 3s ease-in-out infinite',**

**},**

**hidden: {**

**display: 'none',**

**},**

**formContainer: {**

**width: '90%',**

**maxWidth: '350px',**

**transition: 'opacity 0.5s ease-in-out',**

**},**

**fadeIn: {**

**opacity: 1,**

**},**

**fadeOut: {**

**opacity: 0,**

**},**

**signup: {**

**width: '100%',**

**},**

**login: {**

**width: '100%',**

**},**

**formLabel: {**

**display: 'block',**

**color: '#e0e0e0',**

**marginBottom: '10px',**

**fontSize: '24px',**

**fontWeight: 'bold',**

**textAlign: 'center',**

**},**

**inputField: {**

**width: '100%',**

**padding: '15px',**

**margin: '10px 0',**

**borderRadius: '5px',**

**border: '1px solid #333',**

**backgroundColor: '#292929',**

**color: '#e0e0e0',**

**boxSizing: 'border-box',**

**fontSize: '16px',**

**transition: 'background-color 0.3s, transform 0.3s',**

**},**

**submitButton: {**

**width: '100%',**

**padding: '15px',**

**borderRadius: '5px',**

**border: 'none',**

**backgroundColor: '#1e90ff',**

**color: '#ffffff',**

**fontSize: '18px',**

**fontWeight: 'bold',**

**cursor: 'pointer',**

**transition: 'background-color 0.3s, transform 0.3s',**

**},**

**bottomPrompt: {**

**marginTop: '20px',**

**textAlign: 'center',**

**},**

**promptText: {**

**color: '#e0e0e0',**

**marginRight: '10px',**

**},**

**toggleButton: {**

**background: 'none',**

**border: 'none',**

**color: '#ff69b4',**

**cursor: 'pointer',**

**fontSize: '16px',**

**fontWeight: 'bold',**

**transition: 'color 0.3s, transform 0.3s',**

**},**

**termsContainer: {**

**display: 'flex',**

**alignItems: 'center',**

**margin: '10px 0',**

**},**

**checkbox: {**

**marginRight: '10px',**

**accentColor: '#1e90ff', // modern checkbox color**

**},**

**termsLabel: {**

**color: '#e0e0e0',**

**fontSize: '14px',**

**},**

**termsLink: {**

**color: '#1e90ff',**

**textDecoration: 'underline',**

**},**

**socialLoginContainer: {**

**marginTop: '20px',**

**display: 'flex',**

**flexDirection: 'column',**

**alignItems: 'center',**

**},**

**socialButton: {**

**margin: '10px 0',**

**width: '100%',**

**maxWidth: '300px',**

**},**

**};**

**// Keyframes for animations**

**const animations = `**

**@keyframes backgroundAnimation {**

**0% { background-position: 0% 0%; }**

**50% { background-position: 100% 100%; }**

**100% { background-position: 0% 0%; }**

**}**

**@keyframes headerAnimation {**

**0% { transform: scale(1); }**

**50% { transform: scale(1.1); color: #ff69b4; }**

**100% { transform: scale(1); color: #1e90ff; }**

**}**

**@keyframes imageAnimation {**

**0% { transform: scale(1); }**

**50% { transform: scale(1.05); }**

**100% { transform: scale(1); }**

**}**

**`;**

**// Inject keyframes into the document head**

**const styleSheet = document.createElement("style");**

**styleSheet.type = "text/css";**

**styleSheet.innerText = animations;**

**document.head.appendChild(styleSheet);**

**export default Signin;**

# Home.jsx :

# import React from 'react';

# import styled from 'styled-components';

# import { motion } from 'framer-motion';

# import { Link, useNavigate } from 'react-router-dom';

# import { useInView } from 'react-intersection-observer';

# import { useSpring, animated } from '@react-spring/web';

# import FileUpload from '../Frontend/FileUpload/FileUpload';

# import { ReactTyped } from 'react-typed';

# const Container = styled.div`

# font-family: 'Poppins', sans-serif;

# display: flex;

# flex-direction: column;

# align-items: center;

# width: 100vw;

# background-color: #282c34; // Dark blue-gray background

# color: #f5f5f5;

# overflow-x: hidden;

# `;

# const Header = styled(motion.header)`

# width: 100vw;

# display: flex;

# justify-content: space-between;

# align-items: center;

# padding: 40px 20px;

# background-color: #1f232a; // Darker gray for header

# box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.5);

# position: relative;

# z-index: 1000;

# color: #fff;

# overflow: hidden; // To ensure the sliding effect is contained within the header

# &::after {

# content: '';

# position: absolute;

# bottom: 0;

# left: 0;

# width: 100%;

# height: 4px; // Thin line height

# background: linear-gradient(90deg, transparent, #3b82f6, transparent);

# background-size: 200% 100%;

# background-position: -100% 0;

# animation: slide 2s infinite;

# }

# @keyframes slide {

# 0% {

# background-position: -100% 0;

# }

# 100% {

# background-position: 100% 0;

# }

# }

# `;

# const Logo = styled.h1`

# font-size: 2.5em;

# color: #00bcd4; // Cyan color for logo

# font-weight: 700;

# `;

# const Nav = styled.nav`

# display: flex;

# gap: 50px;

# `;

# const NavLink = styled(Link)`

# text-decoration: none;

# font-weight: 600;

# color: #00bcd4; // Cyan color for navigation links

# font-size: 1.2em;

# transition: color 0.3s ease, text-decoration 0.3s ease;

# &:hover {

# color: #ff5722; // Orange color on hover

# text-decoration: underline;

# }

# `;

# 

# const HeroSection = styled(motion.section)`

# width: 100vw;

# padding: 100px 20px;

# text-align: center;

# background: linear-gradient(135deg, #1a1a1a 0%, #000 100%);

# color: #ffffff;

# position: relative;

# `;

# const TypingText = styled.div`

# font-size: 3.5em;

# margin-bottom: 20px;

# font-weight: 700;

# color: #00bcd4; // Cyan color for typing effect

# text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

# `;

# const SubHeadline = styled.p`

# font-size: 2em;

# margin-bottom: 40px;

# font-weight: 300;

# color: #ffffff;

# `;

# const CTAButton = styled(motion.button)`

# padding: 12px 24px;

# font-size: 1.5em;

# color: #ffffff;

# background-color: #00bcd4; // Cyan color for CTA button

# border: none;

# border-radius: 8px;

# cursor: pointer;

# transition: background-color 0.3s ease, transform 0.3s ease;

# &:hover {

# background-color: #0097a7; // Darker cyan on hover

# transform: scale(1.05);

# }

# `;

# const FeaturesSection = styled.section`

# width: 100vw;

# padding: 100px 20px;

# display: flex;

# flex-direction: column;

# align-items: center;

# background-color: #2c2c2c; // Darker background for features section

# position: relative;

# overflow: hidden;

# // Background Graphics

# &::before {

# content: '';

# position: absolute;

# top: 0;

# left: 0;

# width: 200%;

# height: 100%;

# background: radial-gradient(circle, rgba(255, 255, 255, 0.1) 20%, transparent 70%);

# background-size: 200% 200%;

# background-position: -50% -50%;

# animation: backgroundAnimation 10s linear infinite;

# z-index: 0;

# }

# 

# @keyframes backgroundAnimation {

# 0% {

# background-position: -50% -50%;

# }

# 100% {

# background-position: 50% 50%;

# }

# }

# // Adjust content to be above background graphics

# & > \* {

# position: relative;

# z-index: 1;

# }

# `;

# // Title

# const FeaturesTitle = styled.h3`

# font-size: 2.5em;

# margin-bottom: 40px;

# color: #ffffff;

# font-weight: 700;

# text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

# `;

# // Features List Container

# const FeaturesList = styled.div`

# display: flex;

# flex-wrap: wrap;

# gap: 30px;

# justify-content: center;

# `;

# // Feature Item

# const FeatureItem = styled(motion.div)`

# width: 300px;

# padding: 20px;

# background-color: #3a3a3a; // Slightly lighter gray for feature items

# border: 1px solid #333;

# border-radius: 10px;

# text-align: center;

# box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.3);

# transition: transform 0.3s ease, box-shadow 0.3s ease;

# position: relative;

# // Adding a subtle glow effect on hover

# &:hover {

# transform: translateY(-10px);

# box-shadow: 0px 8px 20px rgba(0, 0, 0, 0.5);

# }

# 

# // Add an animation to the feature items

# @keyframes fadeInUp {

# from {

# opacity: 0;

# transform: translateY(20px);

# }

# to {

# opacity: 1;

# transform: translateY(0);

# }

# }

# 

# animation: fadeInUp 0.6s ease-out;

# `;

# const TestimonialsSection = styled.section`

# width: 100vw;

# padding: 60px 20px;

# background-color: #1a1a1a; // Dark background for testimonials

# `;

# const TestimonialsTitle = styled.h3`

# font-size: 2.5em;

# margin-bottom: 30px;

# text-align: center;

# color: #ffffff;

# font-weight: 700;

# `;

# const Testimonial = styled.div`

# width: 100%;

# max-width: 600px;

# margin: 0 auto;

# padding: 20px;

# background-color: #2c2c2c; // Slightly lighter gray for testimonials

# border: 1px solid #333;

# border-radius: 10px;

# text-align: center;

# color: #e0e0e0;

# box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.3);

# `;

# const Footer = styled.footer`

# width: 100vw;

# padding: 20px;

# background-color: #1f232a; // Darker gray for footer

# box-shadow: 0px -4px 10px rgba(0, 0, 0, 0.5);

# display: flex;

# justify-content: space-between;

# align-items: center;

# color: #ffffff;

# `;

# const SocialLinks = styled.div`

# display: flex;

# gap: 15px;

# `;

# const FooterLink = styled.a`

# text-decoration: none;

# color: #00bcd4; // Cyan color for footer links

# transition: color 0.3s ease;

# &:hover {

# color: #ff5722; // Orange color on hover

# text-decoration: underline;

# }

# `;

# const handleContactClick = () => {

# window.open('/contact', '\_blank');

# };

# const Home = ({ setfunc }) => {

# const navigate = useNavigate();

# const { ref: featuresRef, inView: featuresInView } = useInView({

# triggerOnce: true,

# threshold: 0.1,

# });

# const { ref: testimonialsRef, inView: testimonialsInView } = useInView({

# triggerOnce: true,

# threshold: 0.1,

# });

# const featuresSpring = useSpring({

# opacity: featuresInView ? 1 : 0,

# transform: featuresInView ? 'translateY(0)' : 'translateY(50px)',

# });

# const testimonialsSpring = useSpring({

# opacity: testimonialsInView ? 1 : 0,

# transform: testimonialsInView ? 'translateY(0)' : 'translateY(50px)',

# });

# return (

# <Container>

# <Header

# initial={{ opacity: 0, y: -20 }}

# animate={{ opacity: 1, y: 0 }}

# transition={{ duration: 0.6 }}

# >

# <Logo>MY TUTOR</Logo>

# <Nav>

# <NavLink to="/">Home</NavLink>

# <NavLink to="/summarize">Summarize</NavLink>

# <NavLink to="/flashcards">Flashcards</NavLink>

# <NavLink to="/quiz">Quiz</NavLink>

# <NavLink onClick={handleContactClick}>Contact</NavLink>

# </Nav>

# </Header>

# <HeroSection

# initial={{ opacity: 0, y: -50 }}

# animate={{ opacity: 1, y: 0 }}

# transition={{ duration: 1 }}

# >

# <TypingText>

# <ReactTyped

# strings={['Welcome to MY TUTOR', 'Your AI-powered study companion']}

# typeSpeed={40}

# backSpeed={50}

# loop

# />

# </TypingText>

# <SubHeadline>Transforming your study experience</SubHeadline>

# <FileUpload setfunc={setfunc} />

# </HeroSection>

# <FeaturesSection id="features" ref={featuresRef}>

# <FeaturesTitle>Features</FeaturesTitle>

# <FeaturesList>

# <animated.div style={featuresSpring}>

# <FeatureItem>

# <h4>Smart Flashcards</h4>

# <p>Automatically generated flashcards from your study materials.</p>

# <CTAButton

# whileHover={{ scale: 1.1 }}

# whileTap={{ scale: 0.9 }}

# onClick={() => navigate('/flashcards')}

# >

# Generate Flashcards

# </CTAButton>

# </FeatureItem>

# </animated.div>

# <animated.div style={featuresSpring}>

# <FeatureItem>

# <h4>Personalized Quiz</h4>

# <p>Quizzes tailored to your learning progress and needs.</p>

# <CTAButton

# whileHover={{ scale: 1.1 }}

# whileTap={{ scale: 0.9 }}

# onClick={() => navigate('/quiz')}

# >

# Start Quiz

# </CTAButton>

# </FeatureItem>

# </animated.div>

# <animated.div style={featuresSpring}>

# <FeatureItem>

# <h4>Short Summarizing</h4>

# <p>Summarize your entire study material in brief.</p>

# 

# <CTAButton

# whileHover={{ scale: 1.1 }}

# whileTap={{ scale: 0.9 }}

# onClick={() => navigate('/summarization')}

# >

# Get Summarize

# </CTAButton>

# 

# </FeatureItem>

# </animated.div>

# </FeaturesList>

# </FeaturesSection>

# <TestimonialsSection ref={testimonialsRef}>

# <TestimonialsTitle>What Our Users Say</TestimonialsTitle>

# <animated.div style={testimonialsSpring}>

# <Testimonial>

# <p>"My Tutor has completely transformed the way I study. The AI-generated flashcards are a game-changer!"</p>

# <p>- Alex R.</p>

# </Testimonial>

# </animated.div>

# <animated.div style={testimonialsSpring}>

# <Testimonial>

# <p>"I Never Expected Studying to be this much fun and easy!"</p>

# <p>- Madhu M.</p>

# </Testimonial>

# </animated.div>

# 

# </TestimonialsSection>

# <Footer>

# <div>&copy; 2024 My Tutor</div>

# <SocialLinks>

# <FooterLink href="https://twitter.com/mytutor" target="\_blank">Twitter</FooterLink>

# <FooterLink href="https://facebook.com/mytutor" target="\_blank">Facebook</FooterLink>

# <FooterLink href="https://instagram.com/mytutor" target="\_blank">Instagram</FooterLink>

# </SocialLinks>

# </Footer>

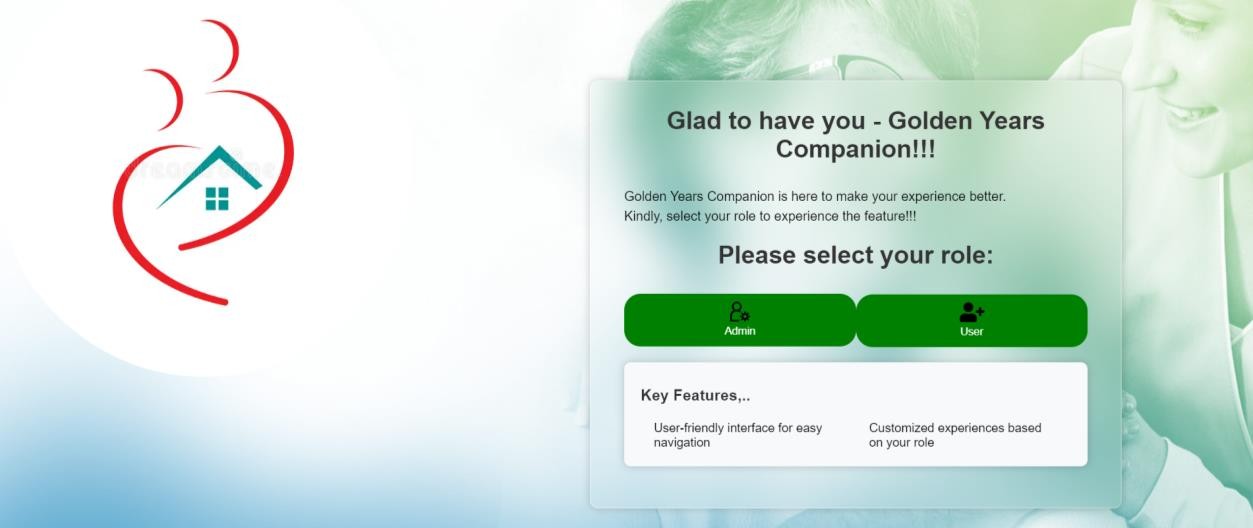
# </Container>

# );

# };

# export default Home;

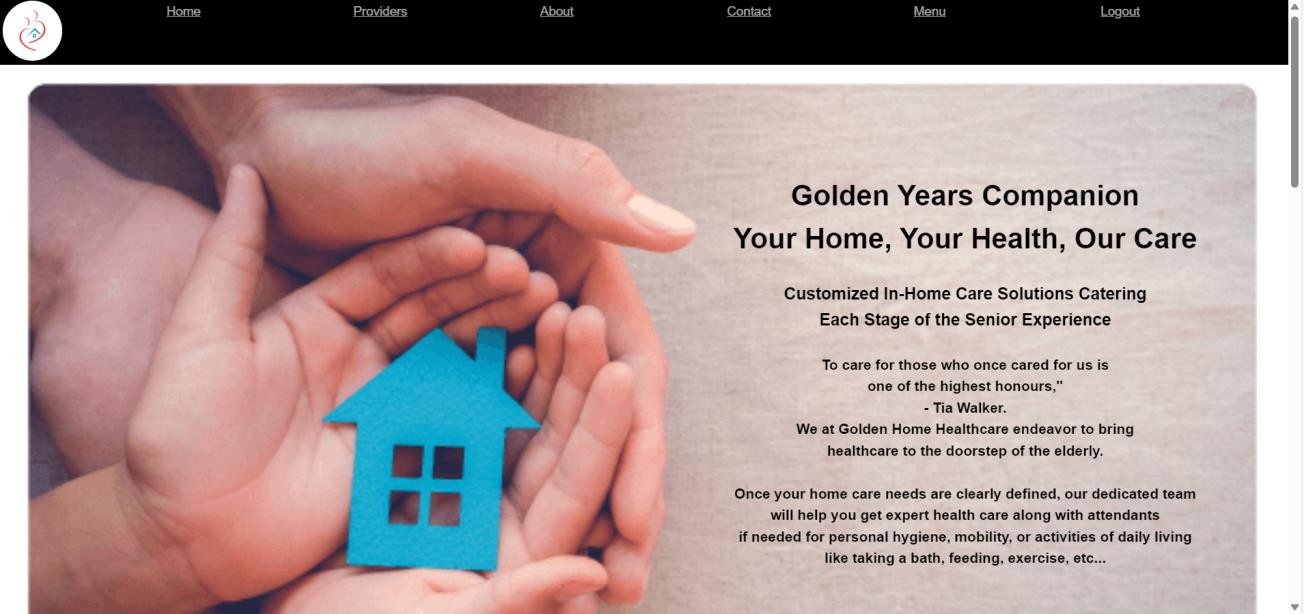
# APPENDIX II SCREENSHOTS



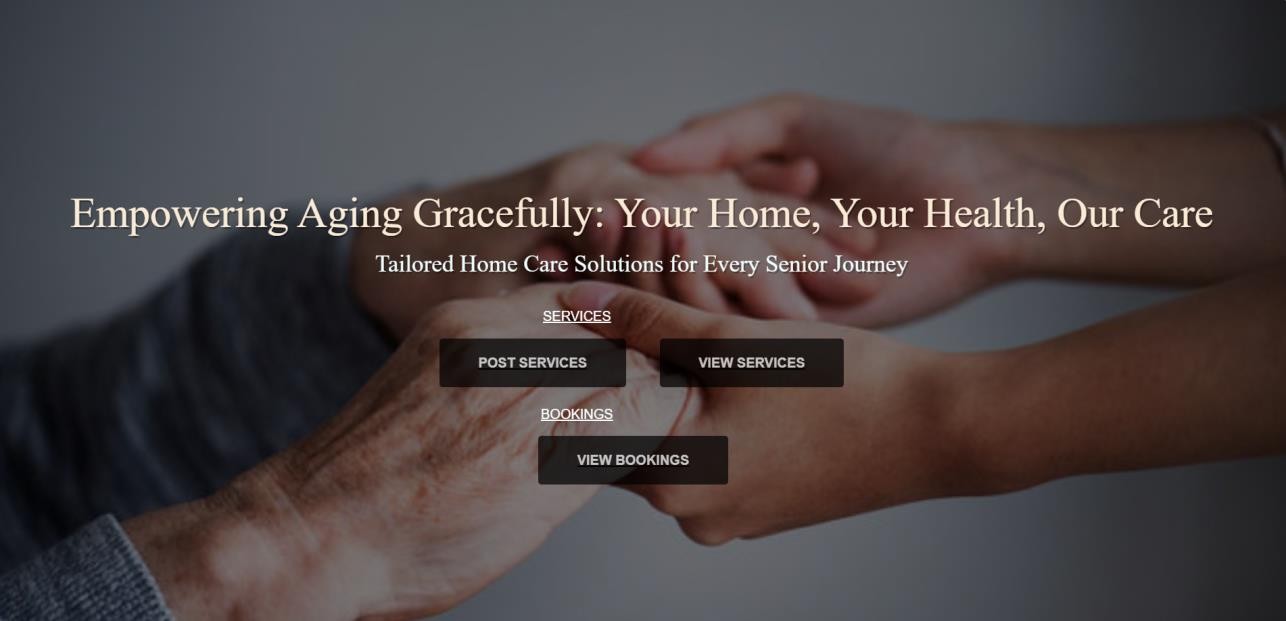
**Fig. A.2.1. Main Page**



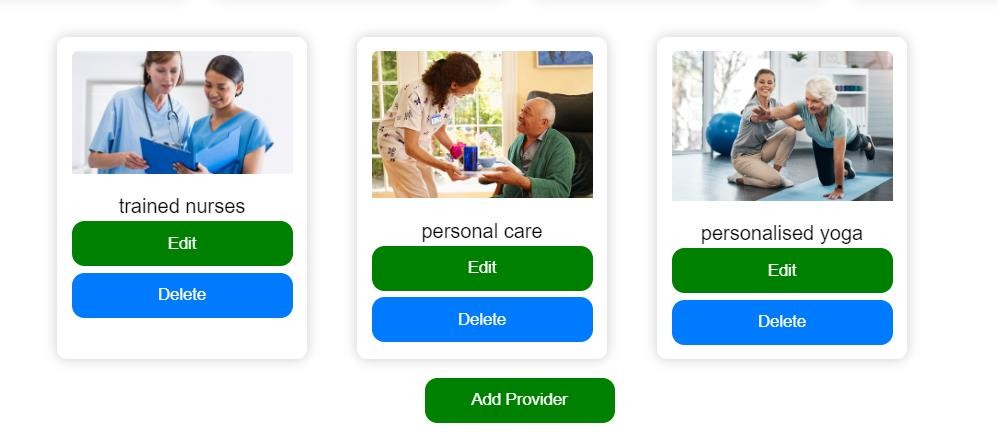
# Fig. A.2.2. Admin Login



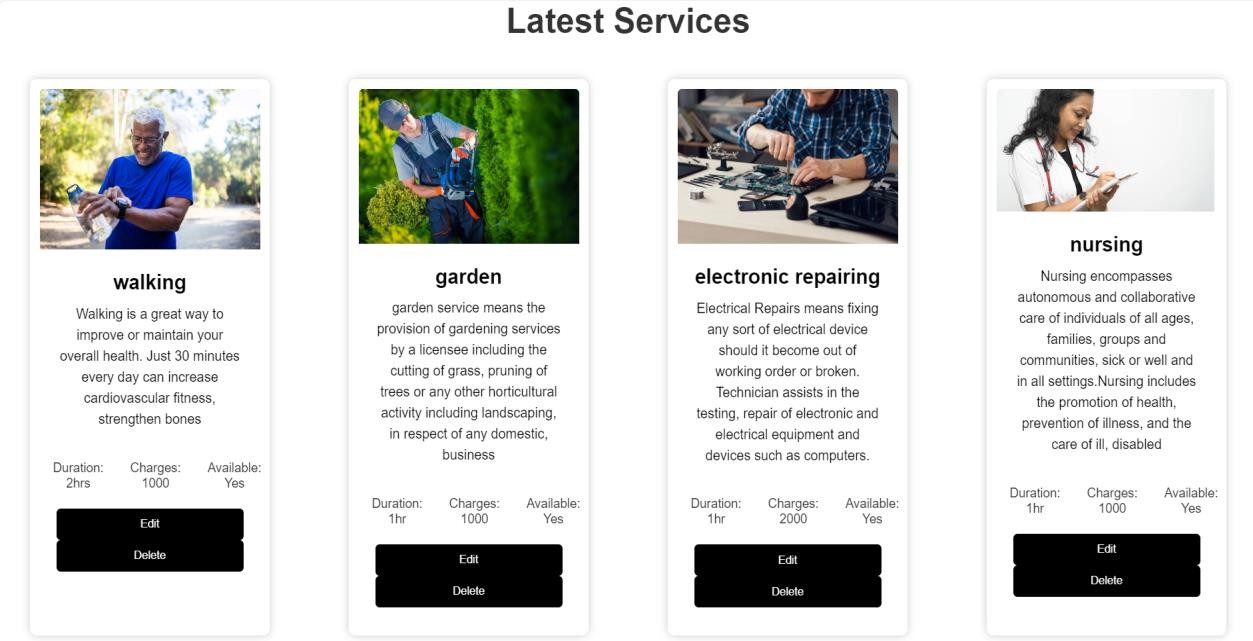
**Fig. A.2.3. Admin Landing Page**



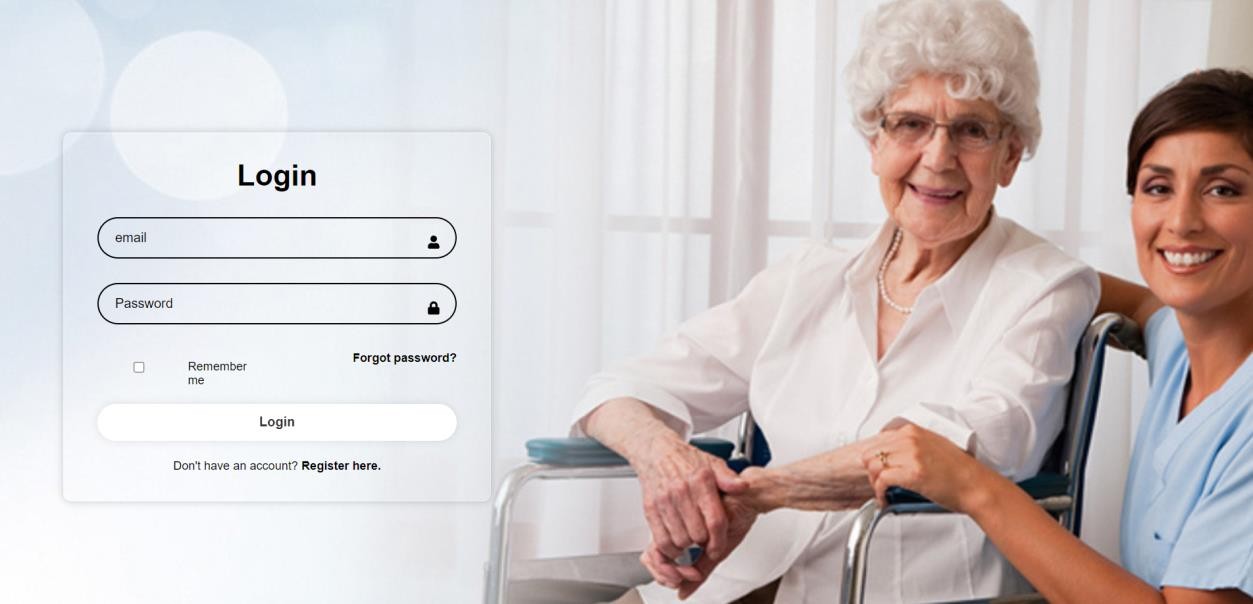
# Fig. A.2.4. Admin Dashboard



**Fig. A.2.5. Providers Management**



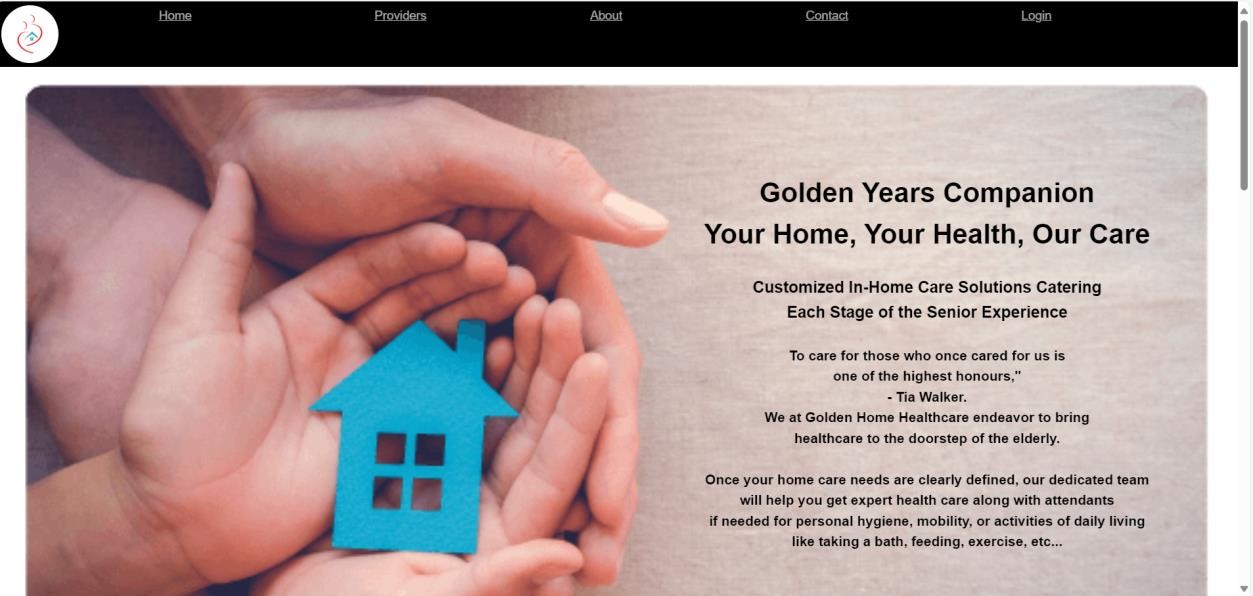
# Fig. A.2.6. Service Management



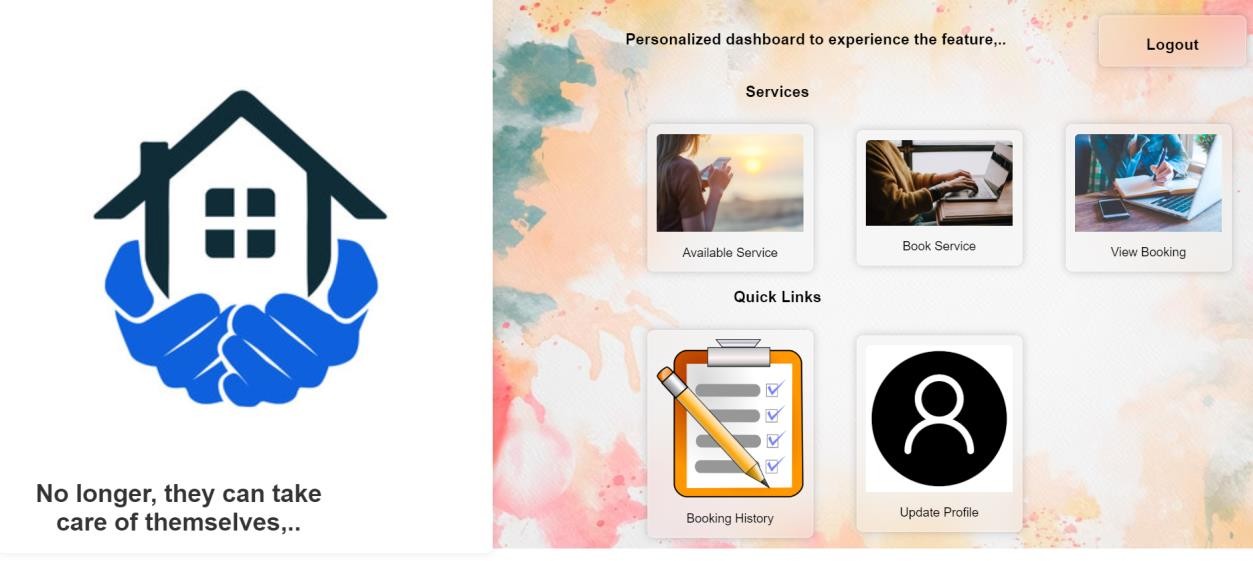
**Fig. A.2.7. User Login Page**



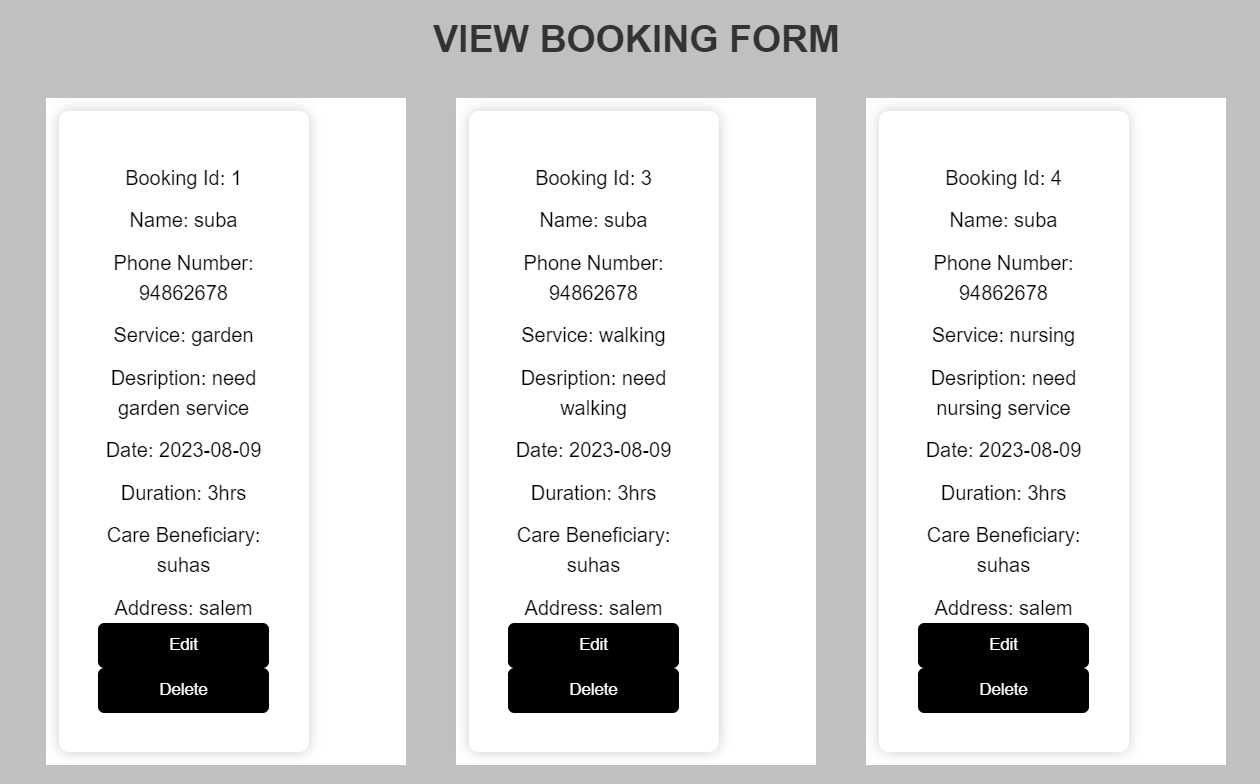
# Fig. A.2.8. User Registration Page



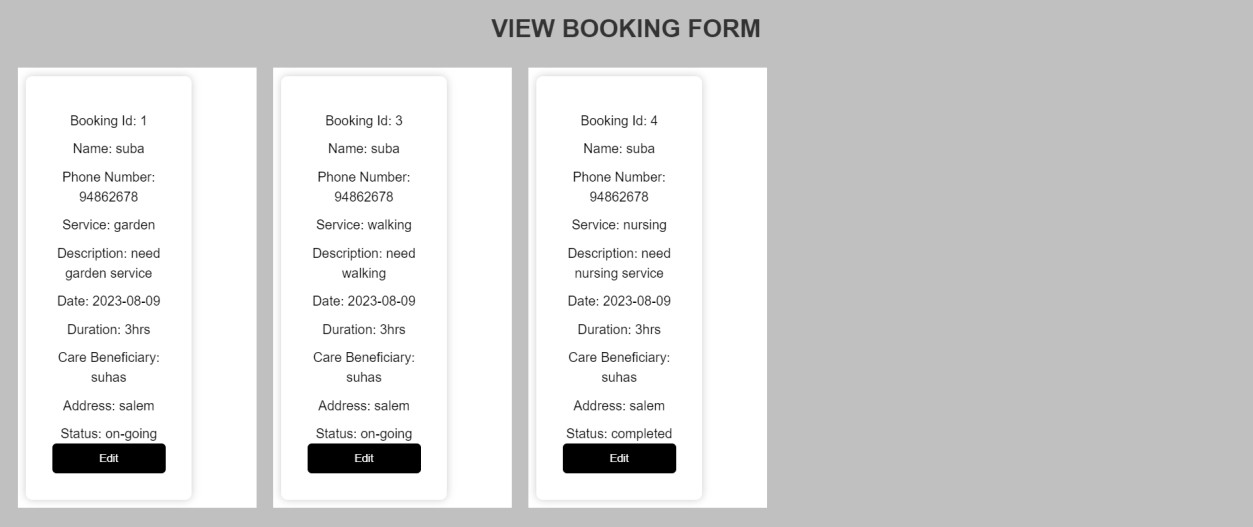
**Fig. A.2.9. User Landing Page**



# Fig. A.2.10. User Dashboard



**Fig. A.2.11. Booking Management**



# Fig. A.2.12. Managing the Status of Booking



**Fig. A.2.13. User Profile Management**

# REFERENCES

**Web references:**

1. PostgreSQL Official Documentation: <https://www.postgresql.org/docs/>
2. React Official Documentation: <https://reactjs.org/docs/getting-started.html>
3. Swagger Official Documentation: <https://swagger.io/tools/swagger-ui/>
4. Java Extensions for Visual Studio Code Official Documentation: <https://code.visualstudio.com/docs/java/extensions>

# Book references:

1. Amanda Davis (2022), “The Ultimate Guide to Home Care for Seniors:

Everything You Need to Know”, Senior Care Publications

1. Emily Wilson (2022), “The Complete Guide to Home Care for Seniors”,

Elderly Care Books

1. Jennifer Brown (2021), “Aging in Place: A Comprehensive Guide to Home Care Services”, Senior Care Publishers
2. Jessica Adams (2022), “Home Care Solutions for Seniors: A Complete Handbook”, Caregiver Press
3. Kimberly Johnson (2021), “Home Care for Seniors: A Practical Guide”,

Senior Care Publishers

1. Laura Miller (2021), “Home Care Services for Seniors: A Comprehensive Manual”, Caregiver Resources
2. Rachel Smith (2021), “Senior Home Care: A Step-by-Step Guide to Caring for Your Loved One at Home”, Family Care Guides
3. Sarah Thompson (2022), “Caring for Aging Parents: A Practical Guide to Home Care Services”, Aging Well Books