CITY NAVIGATION

A PROJECT REPORT

Submitted by

DEEPTHI P

AKHILA MS

ARUNDATHY K

SNG23MCA-2035
SNG23MCA-2014
SNG23MCA-2025

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master Of Computer Applications

Department of Computer Applications

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 623811

November 2024

CITY NAVIGATION

PROJECT REPORT

Submitted by

DEEPTHI P (SNG23MCA-2035)
AKHILA M S (SNG23MCA-2014)
ARUNDATHY K(SNG23MCA-2025)

Under the Guidance of

Asst.Prof. ROHINI RAJU

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master Of Computer Applications

Department of Computer Applications

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 623811

November 2024

DECLARATION

I undersigned hereby declare that the project report "**CITY NAVIGATION**", submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under the supervision of **Assistant Professor. ROHINI.** This submission represents my ideas in my own words and where ideas or words of others have been included, i have adequately and accurately cited and referenced the original sources. I also declare that i have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricatedany data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu

Date:

DEEPTHI P

AKHILLA MS

ARUNDATHY K

Reg No: SNG23MCA-2035

Reg No: SNG23MCA-2014

Reg No: SNG23MCA-2025

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING, KADAYIRUPPU, KOLENCHERY-623811

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report entitled "**CITY NAVIGATION**" submitted by **DEEPTHI P,AKHILA MS,ARUNDATHY K** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications is a bona fide record of the project work carried out by us under ourguidance and supervision. This reportin any form has not been submitted to any other University or Institute for any purpose**.**

Asst.Prof. ROHINI RAJU
Project Guide
Department of Computer Applications

Prof.Dr. Sandhya R
Head of the Department
Department of Computer Applications

ACKNOWLEDGEMENT

I thank GOD almighty for guiding us throughout the project. I would like to thank allthose who have supported to the completion of the project and helped me with valuable suggestions for improvement.I owe my deep sense of gratitude to the Management of Sree Narayana Gurukulam College of Engineering, providing the best suited academic environments for the fulfilment of our project. I would like to place on record our sincere thanks to **Dr. S JOSE** Principal of our institution, for his valuable comments and suggestions to this project. I wish to express my sincere thanks to our Head of the Department **Prof. (Dr.) Sandhya R**, for providing me the opportunity to undertake this project. We deeply indebted to our project guide **Asst.Prof ROHINI RAJU**, and project coordinator **Asst.Prof ALBY** in the Department of Computer Applications for providing me with valuable advice and guidance during the course of the project. I thank all faculty members of our departmentand friends for extending their cooperation during my project.Finally, I convey my thanks to each and every one, who helped me directly andindirectly to carry out this project successfully

DEEPTHI P
AKHILA M S
ARUNDATHY K

ABSTRACT

Elevate your urban experience with City Life, an innovative web application designed to transform your city adventures. Our mission is to empower you to explore the city to its fullest potential effortlessly. City Life enhances your daily activities with a comprehensive range of features. Start by satisfying your dining needs with ease through our extensive restaurant recommendations, where you can discover diverse culinary delights, browse menus, read honest reviews, and secure reservations hassle-free. Beyond dining, City Life offers a wide range of hotel services for locals seeking new experiences or travelers needing reliable accommodation. With transparent pricing, real-time availability, and valuable user reviews, finding the perfect place to stay is just a few clicks away. Transportation is essential to a seamless city experience, and City Life provides detailed information on taxis, rideshares, and public transport schedules, ensuring convenience with real-time updates and travel tips. What sets City Life apart is its commitment to customization; you can tailor your urban experience using our user-friendly interface to filter restaurants, hotels, and travel options based on your preferences. At City Life, your satisfaction is paramount. Our platform ensures you find what you need when you need it, supported by detailed information, contact details, ratings, and reviews. Safety is a top priority, and City Life equips you with the tools and knowledge to navigate the city confidently, backed by our 24/7 customer support team ready to assist with any questions and ensure your city exploration is both exciting and secure.

CONTENTS

CITY NAVIGATION

1 INTRODUCTION

1.1 Project Background

Elevate your urban experience with City Life, an innovative web application designed to transform your city adventures. Our mission is to empower you to explore the city to its fullest potential effortlessly.. Beyond dining, City Life offers a wide range of hotel services for locals seeking new experiences or travelers needing reliable accommodation. With transparent pricing, real-time availability, and valuable user reviews, finding the perfect place to stay is just a few clicks away. Transportation is essential to a seamless city experience, and City Life provides detailed information on taxis, rideshares, and public transport schedules, ensuring convenience with real-time updates and travel tips. At City Life, your satisfaction is paramount. Our platform ensures you find what you need when you need it, supported by detailed information, contact details, ratings, and reviews. Safety is a top priority, and City Life equips you with the tools and knowledge to navigate the city confidently, backed by our 24/7 customer support team ready to assist with any questions and ensure your city exploration is both exciting and secure.

1.2 Existing System

Existing systems for urban exploration and city navigation provide a variety of functionalities but often fall short in integration and comprehensiveness. Restaurant and dining apps like **Yelp** and **TripAdvisor**. Accommodation booking platforms such as **Booking.com** and **Airbnb** Transportation apps like **Uber**, **Lyft**, and **Google Maps** are crucial for navigating the city with real-time information on taxis, rideshares, and public transport. Local experience platforms such as **Eventbrite** and **Meetup** City guide apps like **Foursquare** and **Triposo** provide curated recommendations for attractions and activities but may not offer the integrated, real-time features necessary for a cohesive urban experience. Overall, while these systems provide valuable information and services, they often lack a unified platform that combines dining, accommodation, and transportation features into a seamless user experience.

1.3 Proposed System

The proposed system, **CityNavigator**, is designed to offer a comprehensive and integrated solution for urban exploration, combining dining, accommodation, and transportation features into a single, user-friendly platform. **CityNavigator** enhances the city experience by providing a seamless interface for discovering and enjoying various aspects of city life.

Users can effortlessly browse a wide range of restaurant options, view detailed menus, read authentic reviews, and make reservations with ease. The platform also integrates extensive hotel services, allowing users to search for and book accommodations based on transparent pricing, real-time availability, and user feedback.

For transportation, **CityNavigator** offers detailed information on taxis, rideshares, and public transit schedules, including real-time updates and travel tips to ensure smooth and safe navigation through the city. The application stands out with its robust customization options, enabling users to tailor their city experience by filtering and selecting dining, lodging, and travel options based on personal preferences and budget.

In addition to these core features, **CityNavigator** prioritizes user safety and convenience. It includes real-time safety alerts, contact details for immediate assistance, and a 24/7 customer support team available to address queries and provide help. The platform also allows users to share their experiences by posting photos and reviews, further enriching the community-driven content. By combining these elements into one integrated system, **CityNavigator** aims to transform urban exploration into a more efficient, enjoyable, and personalized experience, making it an indispensable tool for both locals and travelers.

2 SYSTEM ANALYSIS

2.1 Identification of Need

System analysis is the reduction of the entire system by studying various operations and their relationships with the system and the requirements of bit successor. A system can be defined as an orderly grouping of interdependent components linked together according to plan to achieve a specific objective. The idea of the system has become most practical and necessary in conceptualizing the interrelationships and integrations of operations especially when using computers. Organizing consists of several interrelated and interacting components. Analysis is the detailed study of various operations performed by the system and their relations within and outside the system. During analysis, data are connected on the available files, decision points and is handled by the present system.

2.2 Preliminary Investigation

Preliminary analysis is the initial process at the start of the project that determines whether the concept is viable. It looks at economic, market, industry and social trends that influence the success of business endeavours associated with a proposed strategy. Preliminary analysis is repeated in situations where primary investigations trigger updates to plan. Conducting a preliminary analysis of a business strategy allows the organization to see the viability of an intended goal. It creates a comprehensive idea of the enterprise objective and states the outcome is meant to be expressed.

2.3Feasibility Study

During the system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the organization or not. The existing system is manual. Some data are currently recorded in books. The books have to be referred every time when a new item is added or an item is removed. The existing system is compared with the merits of the new system. If there is no loss for the organization then the proposed system is considered as financially feasible.

The results of the feasibility study are:-

1. Technical feasibility

2. Economical feasibility

3. Behavior feasibility

2.3.1 Technical Feasibility

The objective of technical feasibility is to establish the reason for developing the software that is, acceptable to users and adaptable to change and conformable to established standards.

In this project to accomplish the requirement is practical and workable in the software.it is technically easy to use.

2.3.2 Economical Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the system and is commonly known as cost benefit analysis, the procedure made costs. The result of a comparison is found out and changed if needed. This is an on-going effort that improves the accuracy at each phase of the system life cycle. If a benefit outweighs costs, then decision is made to decide and implement the system. Otherwise, further justification or alternation in the proposed system will have to be made and the process is repeated.

It has been proven that the proposed system is economically feasible since it provides several cost benefits.This project is economically feasible because it is less expensive by production and free of cost for the users. It requires a little much expense for the hosting of the product in the web server and it will be profitable from the merits which the users gaining from this project. So it is feasible economically by usage.

2.3.3 Operational Feasibility

This assessment involves undertaking a study to analyse and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

This project undergoes study of operational feasibility and the result states that it doesn't require additional technical knowledge for the operation and use of this project. Any user

with basic knowledge of internet and related activities can perfectly use this project for that, the development team ensures simple and guided user-interfaces.

One of the main problems faced during development of a new system is getting acceptance from the user. There is support from the management of fortune, towards the development of the project. All the operational aspects are considered carefully. Thus, the project is operationally feasible. The system was found to be technically, economically and operationally feasible. The system developed is user friendly, needs less training and improve the working environment.

2.4 Project Planning

The project will go through the following stages of development in its Software Development Life Cycle.

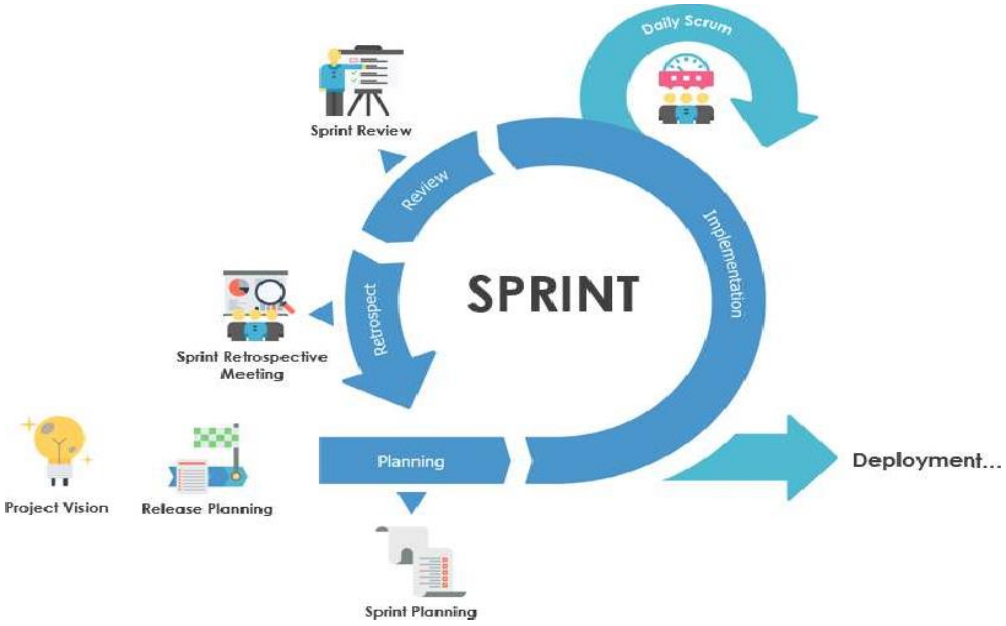


Figure 2.1: Project Planning

2.5 Development Environment

2.5.1 ABOUT THE FRONT END

The system is created using HTML, CSS and JavaScript as front end.

2.5.1.1 HTML

HTML is a computer language devised to allow Website creation. These Websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the W3C, the organization charged with designing and maintaining the language.

HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as a html file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages.

2.5.1.2 CSS

Stands for “Cascading Style Sheet". Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and 8 other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

2.5.1.3 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of Web pages, whose implementations allow clientside script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. 9 JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other Web browsers.

2.6 About the Back End

The system is created Node js

Nodejs

Node.js is a JavaScript runtime environment that allows developers to run JavaScript code on the server side, outside the browser. Built on Google Chrome's V8 JavaScript engine, Node.js provides a fast and efficient platform for building scalable applications. It uses an event-driven, non-blocking I/O model, meaning it can handle multiple tasks concurrently without waiting for each to finish, making it ideal for I/O-heavy operations like web requests, database queries, and file handling. This non-blocking, asynchronous nature helps Node.js manage a large number of simultaneous connections with minimal overhead.One of the key strengths of Node.js is its single-threaded architecture, where an event loop manages incoming requests and delegates tasks efficiently. Despite being single-threaded, Node.js can handle numerous connections at once, thanks to its event-driven approach. It also has access to the Node Package Manager (NPM), which offers a rich ecosystem of open-source libraries and tools that streamline development. Node.js is widely used for building web servers, APIs, real-time applications like chat apps or streaming services, and microservices architectures.

2.6.1 MICROSOFT SQL Server 2008

SQLServer2008 (formerly codenamed "Katmai") was released on August 6, 2008 and aims to make data management self-tuning, self organizing, and self maintaining with the development of SQL Server always on technologies, to provide near-zero down time. SQL Server 2008 also includes support for structured and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as BLOBs (binary large objects), but they are generic bit streams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them. According to Paul Flessner, senior Vice President, Server Applications, Microsoft Corp., SQLServer2008 can be a data storage back end for different varieties of data: XML, email, time/calendar, file, document, spatial, etc as well as perform search, query, analysis, sharing, and synchronization across all data types.

A database system is an overall collection of different database software components and database containing the part viz. Database application programs, front-end components, Database management systems and Database.

A database system must provide the following features:

• A variety of user interfaces

• Physical data independence

• Logical data independence

• Query optimization

• Data integrity

• Concurrency control

• Backup and recovery

• Security and authentication

When creating a database, the main concept is to know how the database is structured in SQL. SQL stands for Structured Query Language. It is a language that enables us to create and operate on relational database, which are sets of related information stored in tables. Because of its elegance and independence.

1.1. Requirement Specification

2.7.1 Software Requirement

The software requirements specification (SRS) is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties–the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development.

Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client's problem and application area. SRS bridges this communication gap. Problem analysis is done to obtain a clear understanding of the needs of the clients and the users, and what exactly is desired from the software. Analysis leads to the actual specification. People performing the analysis called analysts, are also responsible for specifying the requirements. The software project is initiated by the client's needs .In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify their requirements by talking to these people and understanding their needs. These people and the existing documents about the current mode of operation are the basic source of information for the analyst.

2.7.2Hardware Requirement

Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This step acquiring all the facts problem specification such as identifying the desired result determining what information is needed to produce these results and figuring out what process must be carried out to proceed to get the accurate result.

2.8 Software and Hardware Specification

2.8.1 Software Specification

Operating System	**Windows 10 or 11**
Technology	**Nodejs**
Tools	**Express generator**
Client side Technologies	**HTML,CSS,JAVASCRIPT**
Server side Technologies	**Nodejs**
Database Server	**Mongodb**
IDE	**vs code**

2.8.2 Hardware Specification

Processor	**Intel core i3 or above**
Display	**13.5 inch or above**
Hard Disk space	**250 GB minimum**
RAM	**256MB minimum**

3. SYSTEM MODELLING

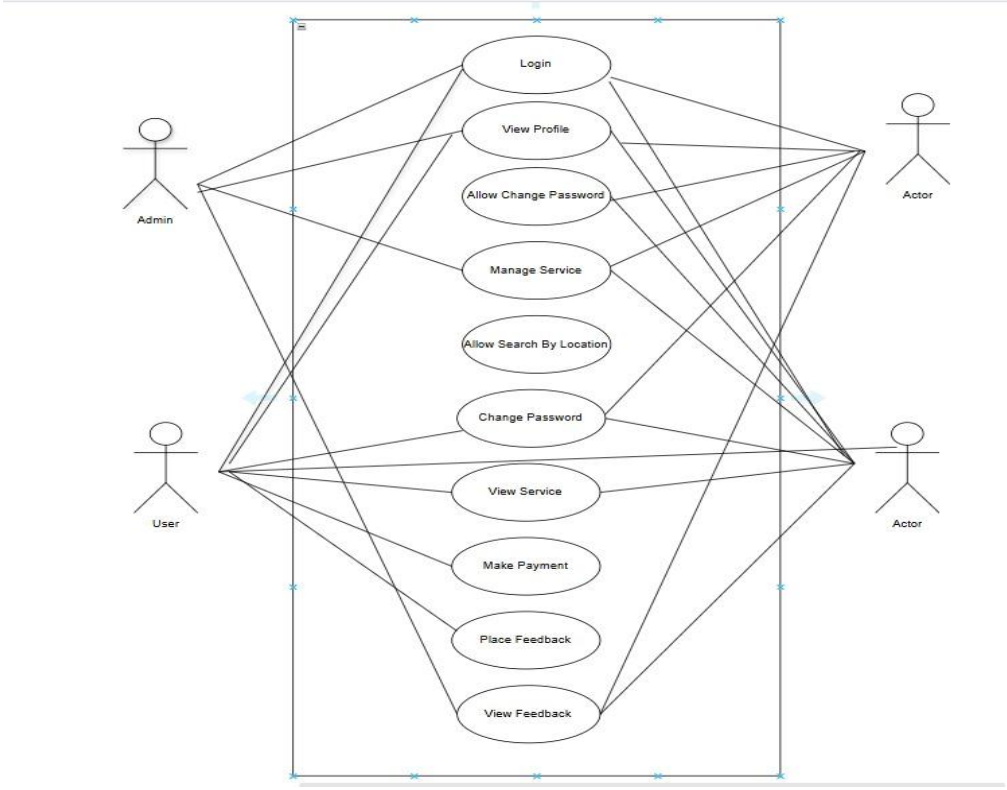
3.1 Design Methodology

Design methodology refers to the development of a system or method for a unique situation. Today the term is most often applied to technological fields in reference to web design, software or information system design. The key design methodology is finding the best solution for each design situation, whether it be in industrial design, architecture or technology. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and wants of the end user is the most critical concern. Design methodology also employs basic research methods, such as analysis and testing.

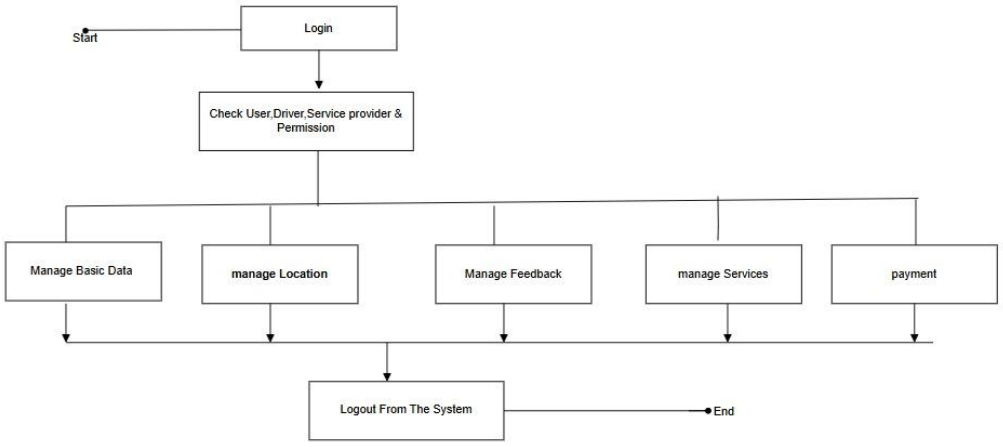
3.2 Use case diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not

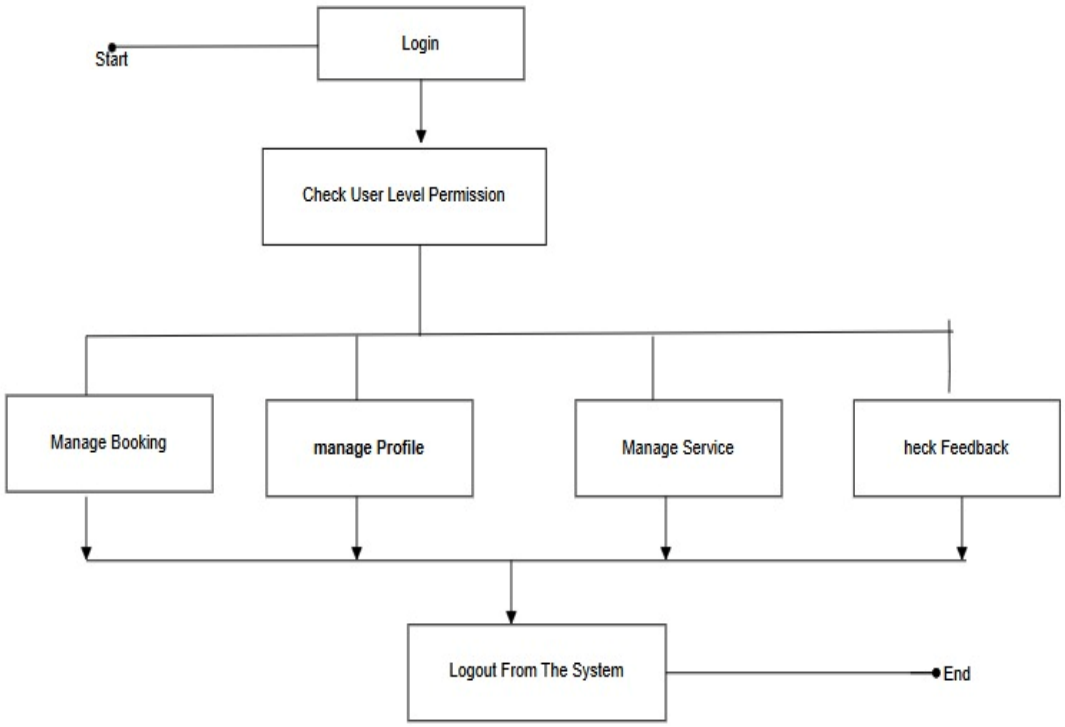
the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour.

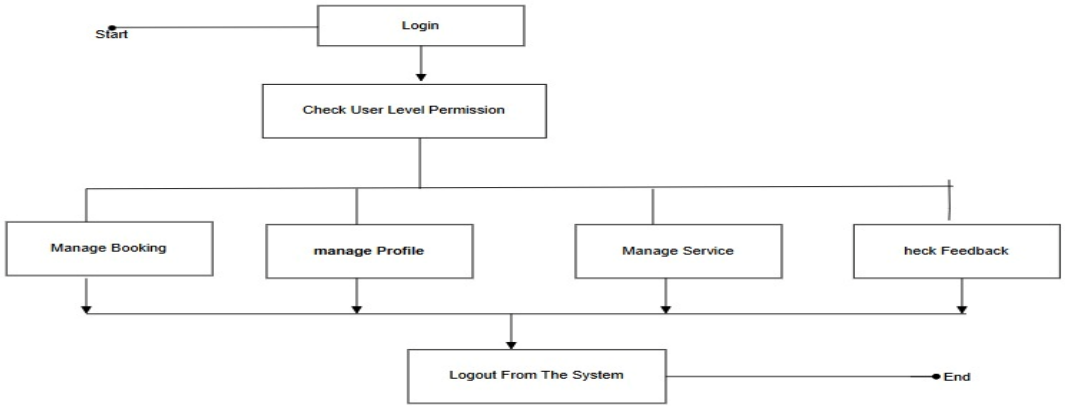


Admin

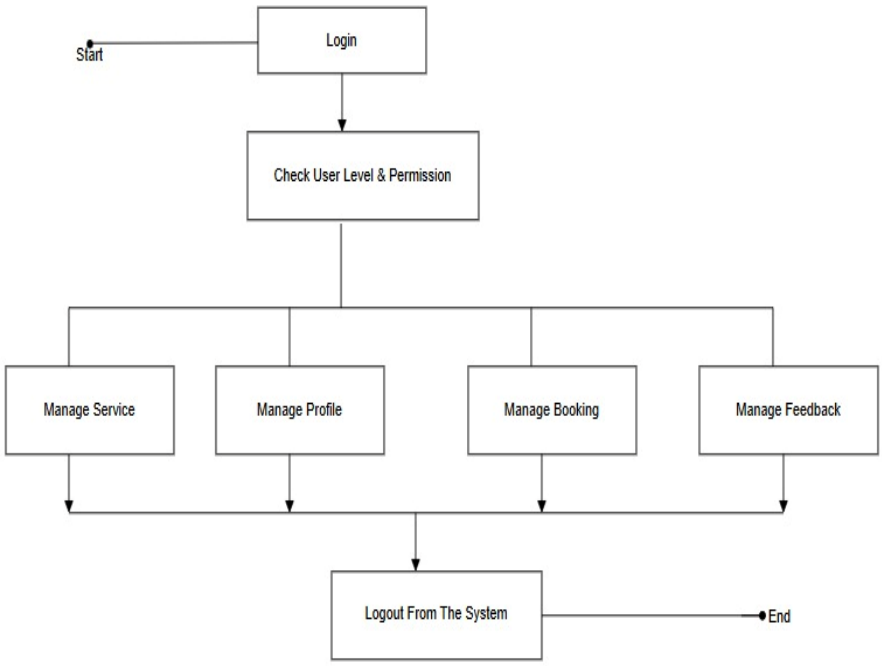


Drivers


Start
Login
Check User Level Permission
Manage Booking
manage Profile
Manage Service
heck Feedback
Logout From The System
End



ACTIVITY DIAGRAM – USER



4. SYSTEM DESIGN

System design is the first step in the development phase for any engineering product or system. The term “Design” is defined as “The process of principles for the purpose of defining a processor a system insufficient detail to permit its realization”. And design is most creative and challenging phase of system development life cycle. It is an approach for the creation of the proposed system is designed, which will help in the system coding. System design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system. A number of sub- systems are to be identified which constitute the whole system.

System design high-end decisions are taken regarding the basic system architecture, platforms and tools to be used. The system design transforms a logical representation of what a given system is required to be in the physical specification. It is an approach for the creation of the proposed system is designed, which will help in the system coding. Design starts with system requirement specification and converts it to a physical reality during the development. Important design fact or such as reliability, response time, throughput of the system maintainability, expand-ability etc. should be taken into account.

System design is the process of developing specifications for the proposed system that meet the needs established in the structured analysis. A major step in the structured design is the preparation of input and output design which will be acceptable to the user. Structured design is the process of planning a new system to replace the old one. Characteristics of well-defined system is:

(1) Acceptability

(2) Decision making ability

(3) Economy

(4) Flexibility

(5) Reliability

(6) Simplicity

a. Module Description

1 Admin Module

Roles:

- Login
- Verify driver
- Admin has to manage the service provider and the user for booking details
- Manage the events and driver allocation.
- Provide vehicle rental facility
- To allow the user to search services through location.
- to allow the user to search services through location.

2. USER Modules

Roles:

- Register
- Login
- Manage Profile
- Search and book service
- Complaint and Feedback
- Change password
- User can make payment

3. Driver ModulesRoles:

- Registration
- Login
- Manage Profile
- Manage Service
- Manage Booking
- Complaint and Feedback

4 .Service provider moduleRoles:

- Registration

b. Input Design

Input design converts user-oriented inputs to computer- based format, which requires careful attention. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via some map support or links. The input design should require careful attention. In this project we use the input design as forms format. We give all the input in this forms format and the data is accepted for computer processing. All the data can be given very easily.

c. Database Design

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

– Determine the data to be stored in the database.

– Determine the relationships between the different data elements.

– Superimpose a logical structure upon the data on the basis of these relationships.

Normalization:

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF. Fifth normal form is very rarely seen.

– **First Normal Form(1NF):** First normal form (1NF) sets the very basic rules for an organized database. Create separate tables for each group of related data and identify each row with the primary key.

– **Second Normal Form(2NF):** Second normal form (2NF) further addresses the concept of removing duplicate data.

– **Third Normal Form(3NF):** Third normal form (3NF) goes one large step further. Meet all the requirements of the second normal form and remove columns that are not dependent upon the primary key.

– **Boyce-Codd Normal Form (BCNF or 3.5NF):** The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form.

– **Fourth Normal Form(4NF):** Finally, fourth normal form (4NF) has one additional requirement. A relation is in 4NF if it has no multi-valued dependencies.

In the Department Management System most of the tables satisfies up to the 3NF.

d. Output Design

Outputs are the most important a direct source of information to the user and to the department. Intelligent output design will improve the systems relationship with the user and help much in decision-making. Outputs are so used to provide a permanent hard copy of the results for later uses. Computer output is the most important and direct source of information the user. Efficient, intelligible output design should improve the systems relationship with the user and help in decision making. In this project we get the output design as forms format. All the input data given is get under processing and we get the output as forms format. These data can be easily accessible to the users.

e. User Interface Design

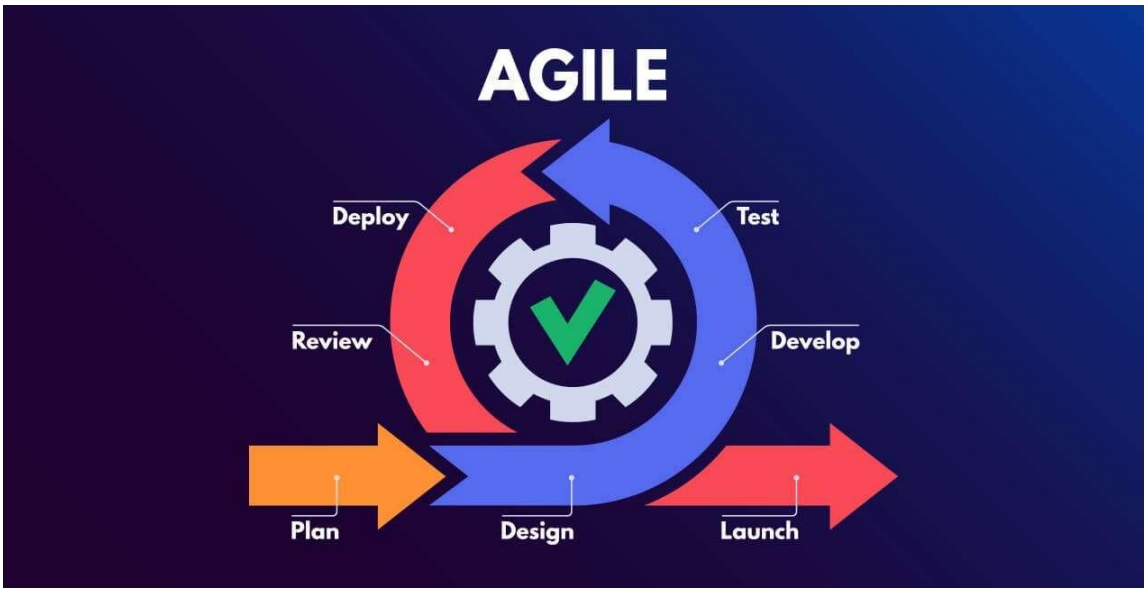
User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements to create a system that is not only operational but also usable and adaptable to changing user needs.

5. AGILE TECHNOLOGY OVERVIEW

5.1 Introduction to Scrum

Scrum is a subset of agile. Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum itself is a simple framework for effective team collaboration on complex products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The

Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum.



AGILE METHOD

5.2 Principles or Methodology Used

The SCRUM methodology is defined by team rules, events(ceremonies),artifacts and roles.

5.2.1 Scrum Team

A scrum team is a collection of individuals (typically between five and nine members) working together to deliver the requested and committed product increments. To work effectively it is important for a scrum team that everyone within the team follow a common goal. The Scrum Team share different tasks and responsibilities related to the delivery of the product. Each role are closely inter-related. It is recommended for Scrum team members work together in the same location whenever possible. There are 3 roles in a scrum team:

1. The Product Owner: The product owner is a project's key stakeholder-Usually an internal or external customer. or a scope person for the customer. There is only one product owner who conveys the overall mission and vision of the product which the team is building. The product owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

2. The Scrum Master: The scrum master is the servant leader to the product owner, development team and organization. With no hierarchical authority over the team but rather more of a facilitator, The scrum master ensures that the team adheres to scrum theory, practices and rules. The scrum master protect the team by doing anything possible to help the team perform of the highest level. This may include removing impediments, facilitating meetings and helping the product owner groom the backlog.

3. The Development Team: is a self organizing, cross functional group armed with all of the skills to deliver shippable increments at the completion of each sprint.

5.2.2 Scrum Events

Scrum events are time-boxed events that means in a project, every scrum events has a predefined maximum duration. These events enable transparency on the project progress to all who are involved in the projects. The vital events of scrum are -the Sprint.

1. The Sprint: A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprint are usually 2-4 weeks long but can be as short as 1 week .

2. Sprint Planning: Sprint Planning team meetings are time-boxed events that determine which product backlog items will be delivered and have the work will be achieved.

3. The daily Stand-up: The daily stand-up is a short communication meeting in which each team member quickly and transparently covers progress since the last stand-up ,planed work before the next meeting and any impediments that may be blocking his or her progress.

4. The Sprint Review: The sprint review is the ”show and tell” of demonstration events for the team to present the work completed during the sprint. The product owner checks the work against predefined acceptance criteria or either accept or reject the work. The stakeholder or client give feedback to ensure that the delivered incremental must the business model.

5. Retrospective: The retrospective or retro is the final team meeting in the sprint to determine what went well, what didn’t go well and how the team can improve the next sprint. Attended by the team and the scrum master, the retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its process.

5.2.3 Scrum Artifacts

Scrum artifacts are designed to increase transparency of information related to the delivery of the project, and provide opportunities for inspection and adaptation. They are management products useful for the creation of the specialist product of the project. There are 3 artifacts in scrum

1. Product Backlog An extended list of everything that might be needed in the final product.

2. Sprint Backlog Selected items for the product backlog to be delivered through a sprint, along with the task for delivering the item and realizing the sprint goal.

3. The Sprint Review Increment set of all the product backlog items so far in the project.

5.2.4 Scrum Rules

The rules of Agile Scrum Should be completely up to the team and governed for what works best for their processes. The best agile will tell teams to start with basic scrum events listed above and then inspect and adopt based on your teams unique needs so there is continuous improvement in the way teams work together.

5.3 Sprint

5.3.1 Product Backlog

In the simplest definition the Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user centric e.g. in the form of user stories. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the development Team collaborate on the details of Product Backlog items. A Scrum product backlog contains descriptions of the functionality desired in an end product. Agile backlog prioritization is the next step. The Product Backlog is the tool used by the product owner to keep track of all of the features that stakeholders would like to see implemented in the product whereas the Sprint Backlog is a subset of the Product Backlog representing the current active Sprint iteration. However, the product owner prioritizes it.

Feature ID	User Story	Estimated (Hr/Day)	Priority
f01	As an admin,I want to to login to the system,so I can manage users , services and verify drivers	12	High
f02	As a user I want to login ,so I can create an account and use the service and update informations	5	High
f03	As a user, I want to search services by location, so I can find services near me	6	High
f04	As a service provider, I want to add new services so I can expand the offering	6	High
f05	As a driver I want to register, so I can join the system and offer transportation	5	High
f06	As a driver I want to view/update my profile	5	Medium
f07	User can view the approximate rate of vehicles	6	High

Table 5.1 Product Backlog

5.3.2 Sprint Planner

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box. The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. As the Development Team works, it does so with the Sprint Goal always in mind.

Feature ID	Sprint	Start Date	End Date	Estimated Work(hrs)	Duration (Days)	Sprint Goal
f05,6	1	18/9/23	30/9/24	8	14	Create an environm-ent for manage and a-

						vailability of sevices
f06,4	2	2/10/23	14/10/24	8	14	User authentication And registeration , User profile
f07	3	16/10/23	28/10/24	8	14	Payment integration
f02,3	4	30/10/23	11/11/24	8	14	Buy car spare parts
f01	5	13/11/23	25/11/24	8	14	Location tracking

Table 5.2 Sprint Planner

5.3.3 Ideal Burn Down Chart

A burndown chart is a graphic representation of how quickly the team is working through a customer's user stories, an agile tool that is used to capture a description of a feature from an end-user perspective. The burndown chart shows the total effort against the amount of work for each iteration. The quantity of work remaining is shown on a vertical axis, while the time that has passed since beginning the project is placed horizontally on the chart, which shows the past and the future. The burndown chart is displayed so everyone on the team can see it and is updated regularly to keep it accurate.

There are two variants that exist for a burndown chart. A sprint burndown is for work remaining in the iteration. When illustrating the work remaining for the entire project, the chart is called a product burndown. The burndown chart has several points. There's an x-axis, which is the project or iteration timeline. The y-axis is the work that needs to get done in the project. The story point estimates for the work that remains is represented by this axis. The project starting point is the farthest point to the left of the chart and occurs on day zero of the project or iteration. The project end point is farthest to the right and marks the final day of the project or iteration.

There is an ideal work remaining line, which is a straight line connecting the start point to the end point. It shows the sum of the estimates for all the tasks that need to be completed. At the end point, the ideal line crosses the x-axis and shows there is no work left to be done. This line is based on estimates and therefore not always accurate.

Then there is the actual work remaining line that shows the actual work that remains in the project or iteration. At the beginning the actual work remaining and the ideal work remaining are the same, but as the project or iteration progresses the actual work line will fluctuate above and below the ideal work line. Each day a new point is added to this line until the project or iteration is done to make sure it's as accurate as possible. If the actual work line is above the ideal work line, it means there is more work left than originally thought. In other words, the project is behind schedule. However, if the actual work line is below the ideal work line, there is less work left than had been predicted and the project is ahead of schedule.

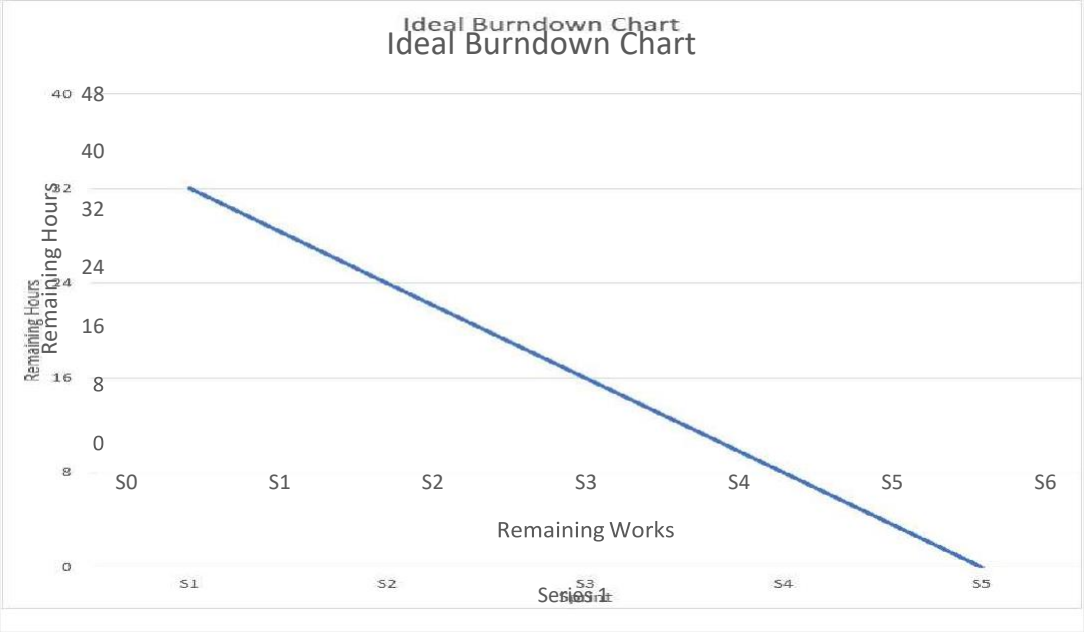


Figure 5.1 : Ideal Burndown Chart

5.3.4 Git Hub Registration

GitHub is an online-browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more. GitHub offers free accounts for users and organizations working on public and open sourceprojects, as well as paid accounts that offer unlimited private repositories and optional

user management and security features. Git hub account creation includes the following steps:

– Go to the GitHub sign up page, then Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters.

– Review carefully the GitHub Terms of Service and Privacy Policy before continuing and Choose a plan. Hereby anyone can finish the account creation procedure.

– You can store a variety of projects in GitHub repositories, including open source projects.

– In the upper-right corner of any page, click , and then click New repository.

– Type a short, memorable name for your repository followed by Optionally, add a description of your repository, public or private repository.

– Select Initialize this repository with a README. finally Click Create repository

– After creation, need to collaborate members by the admin.

– In the left sidebar, click Collaborators and teams.

– Under ”Collaborators”, type the name of the person you’d like to give access to the repository, then click Add collaborator.

– Next to the new collaborators name, choose the appropriate permission level: Write, Read, or Admin.

– The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.

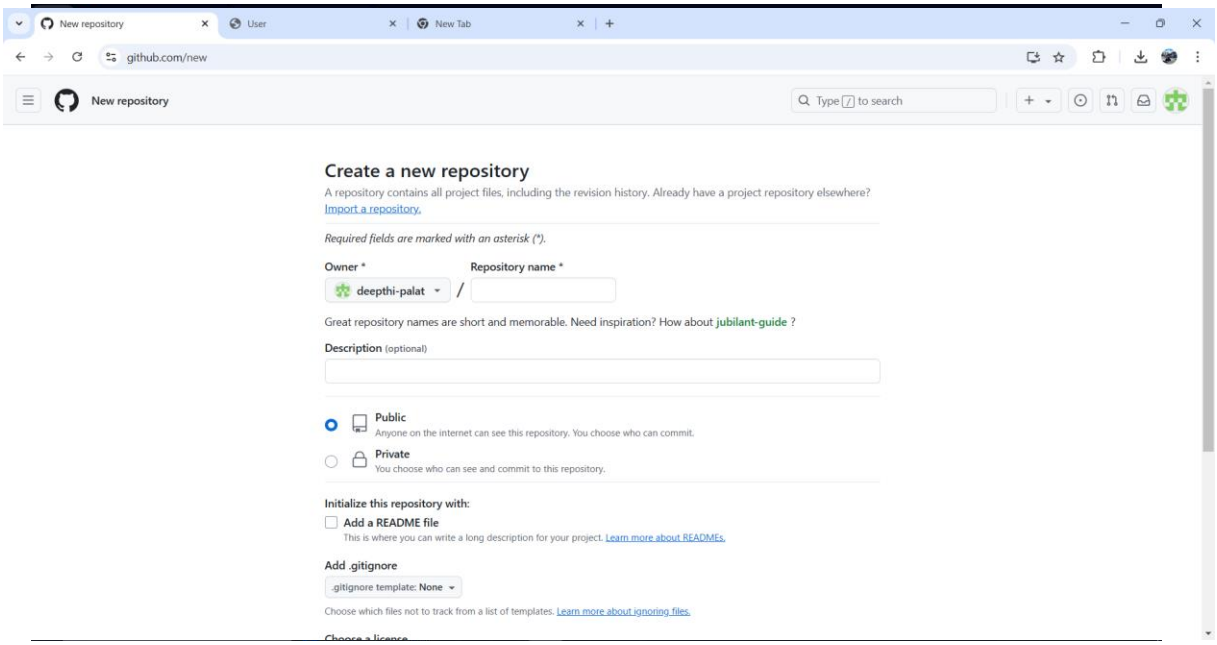


Figure 5.2 : Repository Creation

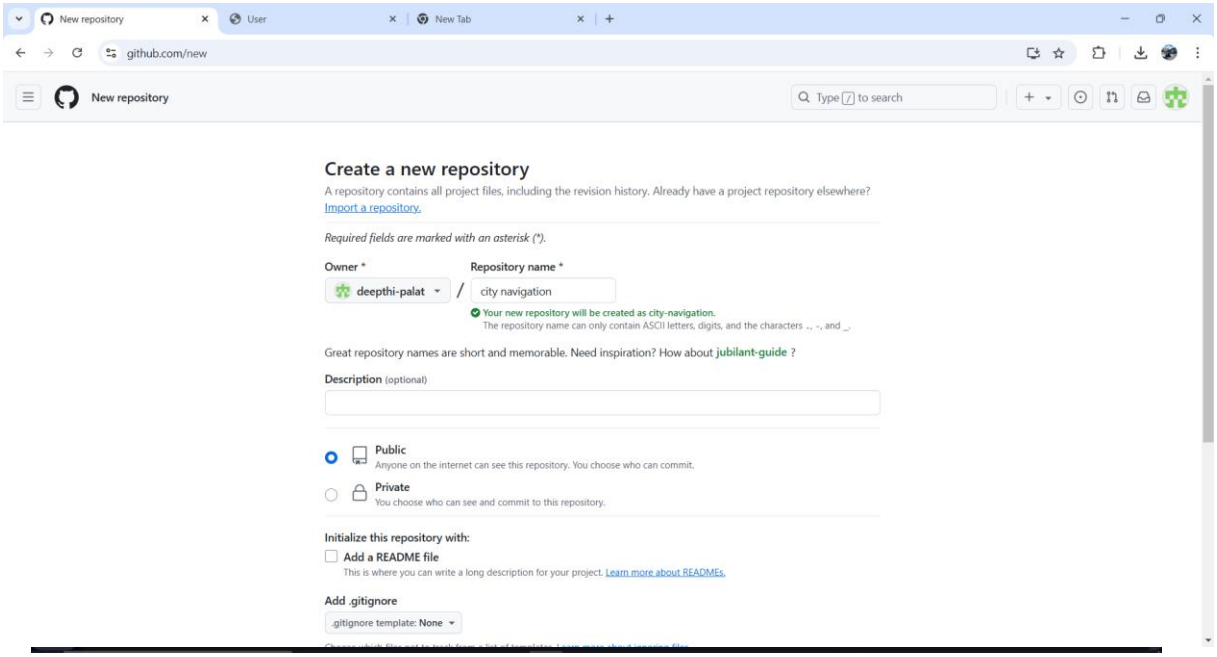


Figure 5.3 : Adding Collaborator

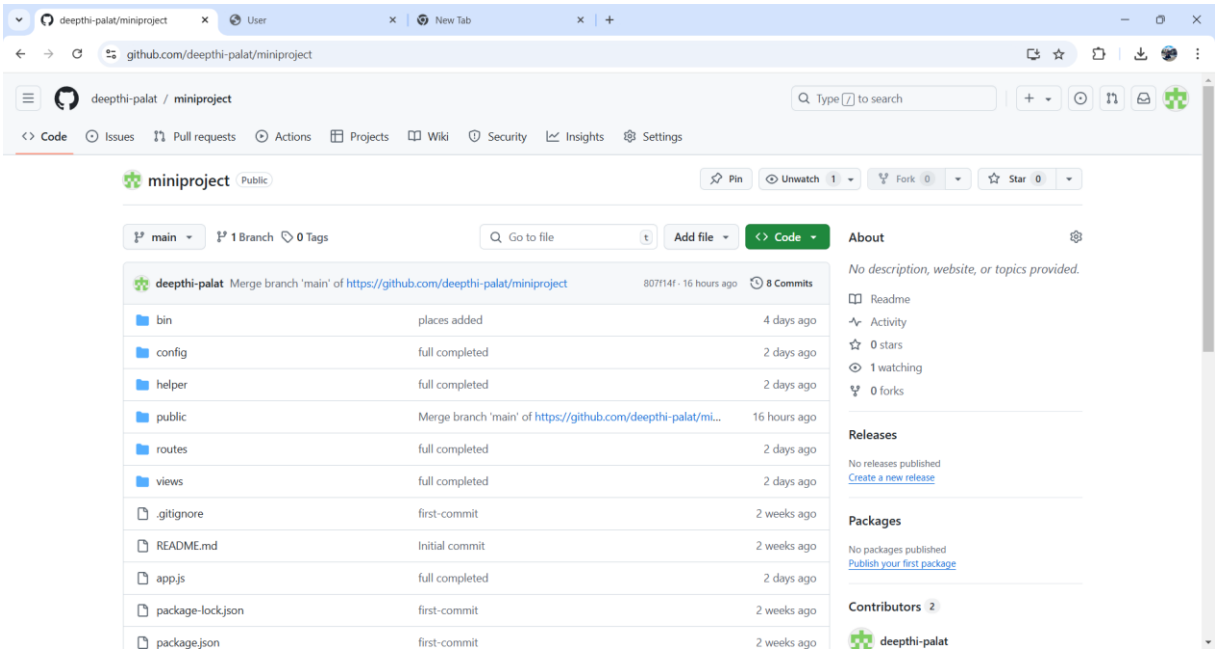


Figure 5.4 : Poject uploading

6. CODING

a. Coding Standards Followed

PHP follows few rules and maintains its style of coding. As there are many developers all over the world, if each of them follows different coding styles and standards this will raise great confusion and difficulty for a developer to understand another developer's code. It will be very hard to manage and store the code for future reference.

The coding standards come into play for all the reasons

Reasons for maintaining coding standards:

- People working in different modules can easily manage the source code due to familiarity.
- Acts as a blueprint for other team members to understand the code for management or changes.
- Clarity saves a lot of time doing common mistakes as it's easy to understand.
- Coding standards or industry standards improve the quality and consistency of software.

Coding standards or industry standards improve the quality and consistency of software.

- PHP tags: One must use the PHP standard tags, rather than the shorthand tags to delimit the PHP code.
- Comments: Use of standard C and C++ commenting style i.e., (//) for single line and (/* */) for multiple-line, is highly encouraged and use of Python or Perl style of commenting i.e., (#), is discouraged.
- Line length and Indentation: It is a standard recommendation to not exceed more than 75-85 characters per line of code. One must not use tabs for indentation instead use spaces as it is the standard indenting method in most programming languages. One statement per line or else split lines for long multiple ones.
- Control structures: The control flow or conditional statements must be written in such a way so that they could be differentiated from function call statements. While

writing if, for, else, foreach, do, while, switch, and other controls flow statements there must be one space between the keyword and the opening parenthesis. The function definitions do not have any space between the function name and opening parenthesis.

- Function calls: There should be no spaces between parameters, comma, last parameter, and semicolon. Always use full PHP tags and avoid shorthand tags. Also, give one space between both sides of '=' operator.
- Naming variables: There are few conventions that one must follow in order to name the variables:
- Use of lower case letters to name the variables.
- Use of '_' to separate the words in a variable.
- Static variable names may be started with the letter 's'.
- Global variable names must start with the letter 'g'.
- Use of upper-case letters to define global constants with '_' as a separator.
- Block alignment : Every block of code and curly braces must be aligned.
- Short Functions: All functions and methods must limit themselves to a single page and must not be lengthy.

7. CODE REVIEW AND TESTING

7.1 Code Review

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to Do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases.

Testing Objectives include:

– Testing is a process of executing a program with the intent offending an error.

– A good test case is one that has a probability of finding an as yet undiscovered error.

– A successful test is one that uncovers an undiscovered error.

Testing Principles are:

– All tests should be traceable to end user requirements

– Tests should be planned long before testing begin

– Testing should begin on a small scale and progress towards testing in large

– Exhaustive testing is not possible

– To be most effective testing should be conducted by an independent third party.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can cause chaos and confusion. The implementation stage involves the following tasks :

– Careful planning.

– Investigation of system and constrains. of methods to achieve the changeover.

– Training of staff in the changeover phase. – Evaluation of the changeover method.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

7.2 Testing Process

Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program. Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing present interesting anomaly for the software engineer.

Unit testing

This is the first of testing. In this different modules are tested against the specification produces during the design of the modules. It refers to the verification of single program module in an isolated environment. Unit testing focuses on the modules independently of one another to locate errors.

In our project we test each module and each forms individually. Each forms may tested using appropriate values. The input screens need to be designed very carefully and logically. While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered.

Validation Checks

As a web application developer, form validation is a crucial part of your work, and it should not be underrated as it could lead to security flaws in your application. You should consider it a must if you're striving to provide a professional end user experience.

– Basic form validation

– Custom error messages

I always prefer to load common libraries and helpers in the constructor of the controller itself as it's a nice habit to avoid code duplication elsewhere in controller methods. We load the form and url helpers so that we can use the utility methods provided by those helpers throughout the rest of the application. The form validation library in the constructor, you can access it using the form validation convention.

8. IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users, that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints implementation, design of methods to achieve the change over, an evaluation, of change over methods.

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after through testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

At the beginning of the development phase a preliminary implementation plan is created to schedule and manage the many different activities must be integrated into plan. The implementation plan is updated throughout the development phase, culminating in a changeover plan for the phase. The major elements of implementation plan are test plan, equipment installation plan and a conversion plan.

There are three types of implementation:

• Implementation of a computer system to replace a manual system.

• Implementation of a new computer system to replace an existing one.

• Implementation of a modified application to replace an existing one, using the same computer.

9. CONCLUSION

The project entitled “CAR BAY” was completed on time. This project provided maximum interaction and flexibility. The system was tested and the performance of the system was provided to be much efficient and data maintenance is achieved partially. The system has been developed in attractive fashion.

The modules in the system help in faster development, implementation and maintenance of the software. This system has been developed as versatile and user friendly as possible keeping in mind the advanced features. Using HTML, CSS, JavaScript, PHP and MS SQL SERVER 2008, the system was developed and tested with all possible samples of data. As a whole, the system was well planned and designed. The performance of the system is proved to be efficient. And it already provides all the objectives, I have identified before.

All modules are tested separately and put together to form the main system. Finally the system is tested with the real data and everything worked successfully. Thus the system has fulfilled the entire objective identified. The system required least hardware requirement to work on. So we can state, we have developed such a good environment for communication, to connect with more people. And it provides a number of advantages too as we have described in the previous sections.

To conclude this, i thank all people who helped me to complete this project work successfully.

10. BIBLIOGRAPHY

SYSTEM ANALYSIS AND DESIGN: Elias M Award,

Second Edition,1999,

Galgotia Publications, Delhi.

DATABASE SYSTEM CONCEPTS: Silberschatz, Korth,

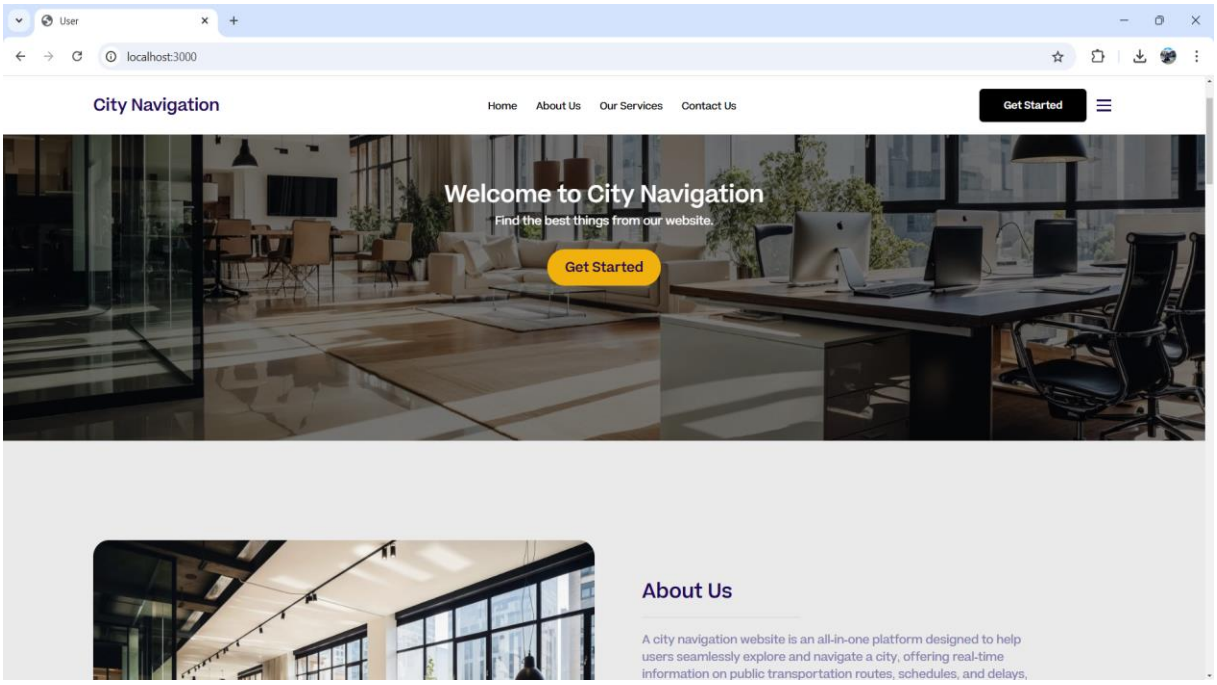
Sudarshan, Fourth Edition.

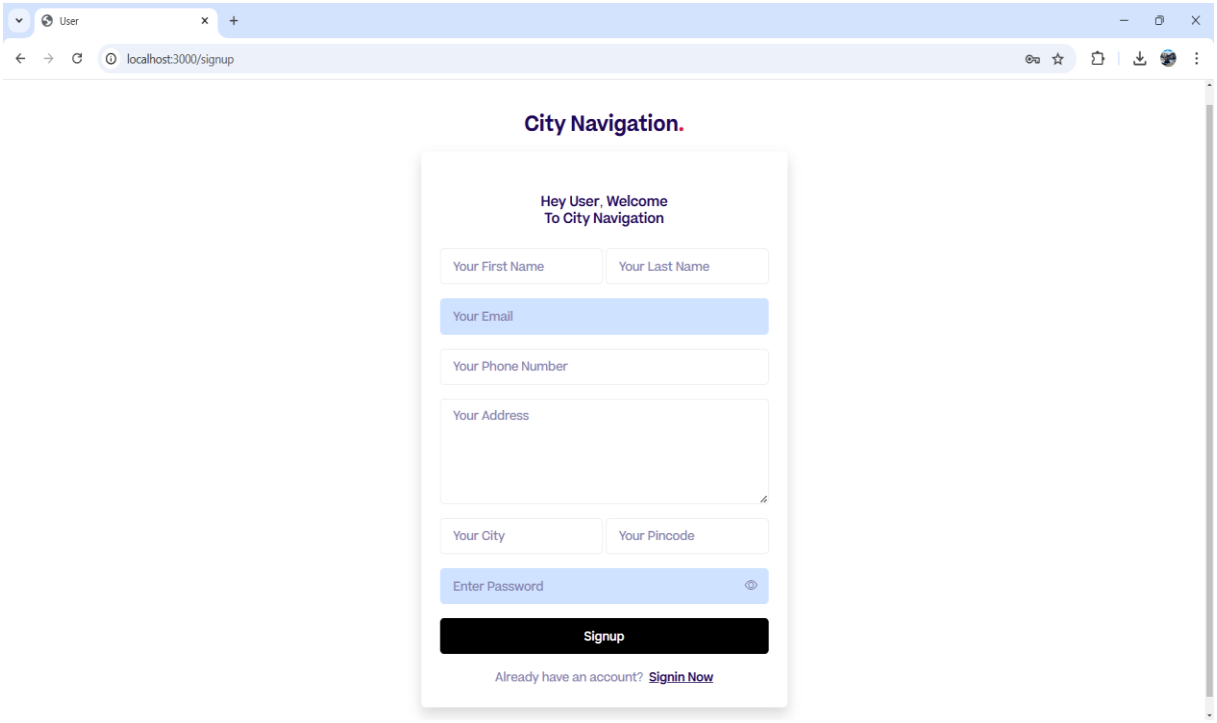
WEB PROGRAMMING USING PHP: Dr. Jeeva Jose.

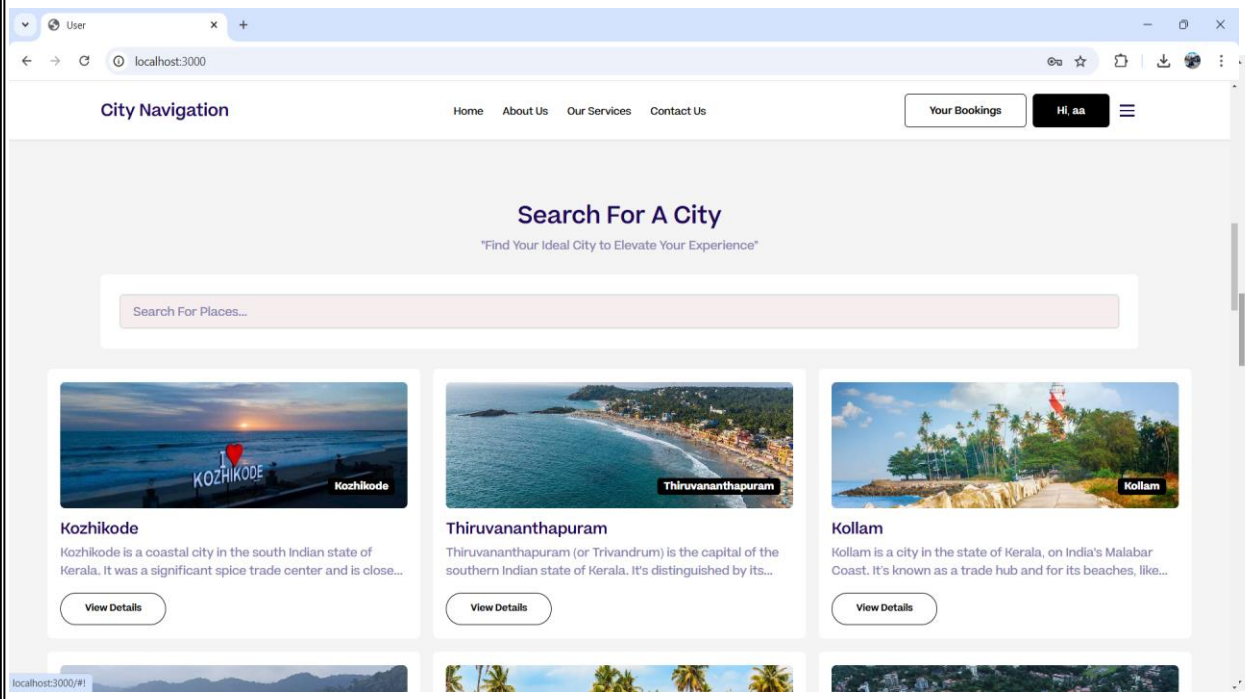
Kalyani publications

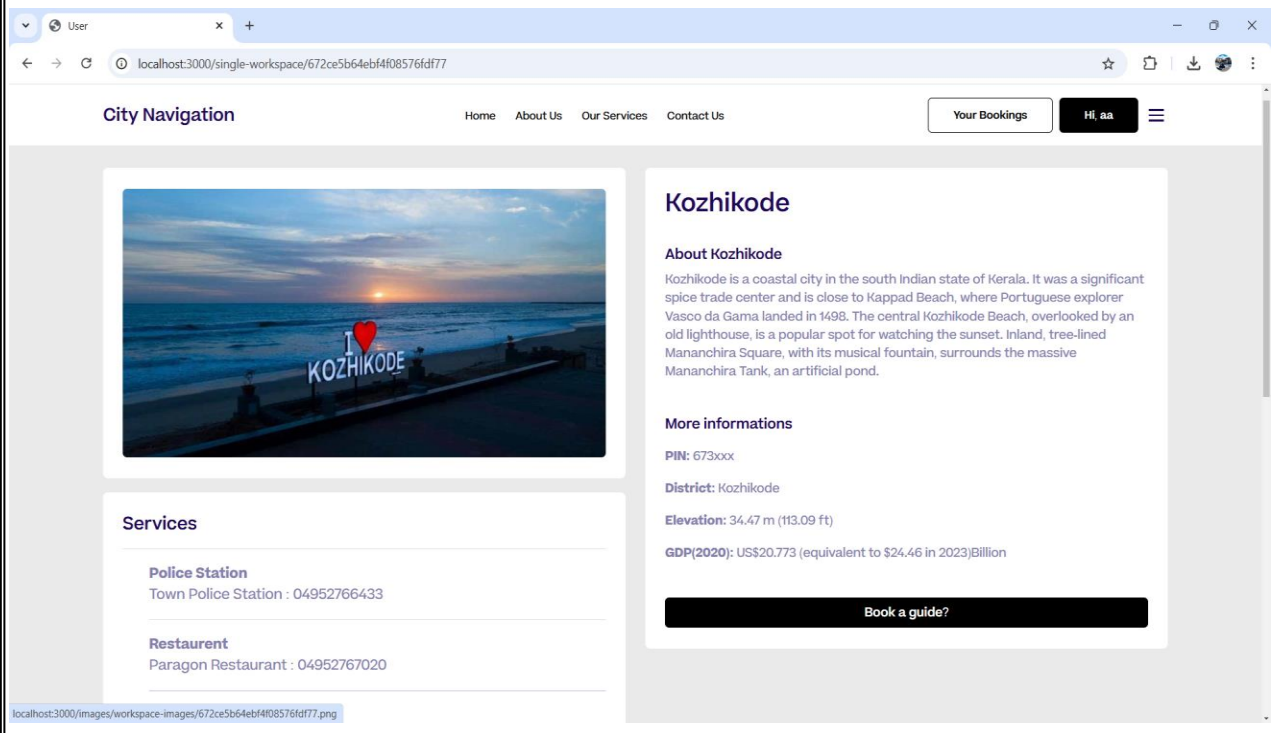
10 Screenshots

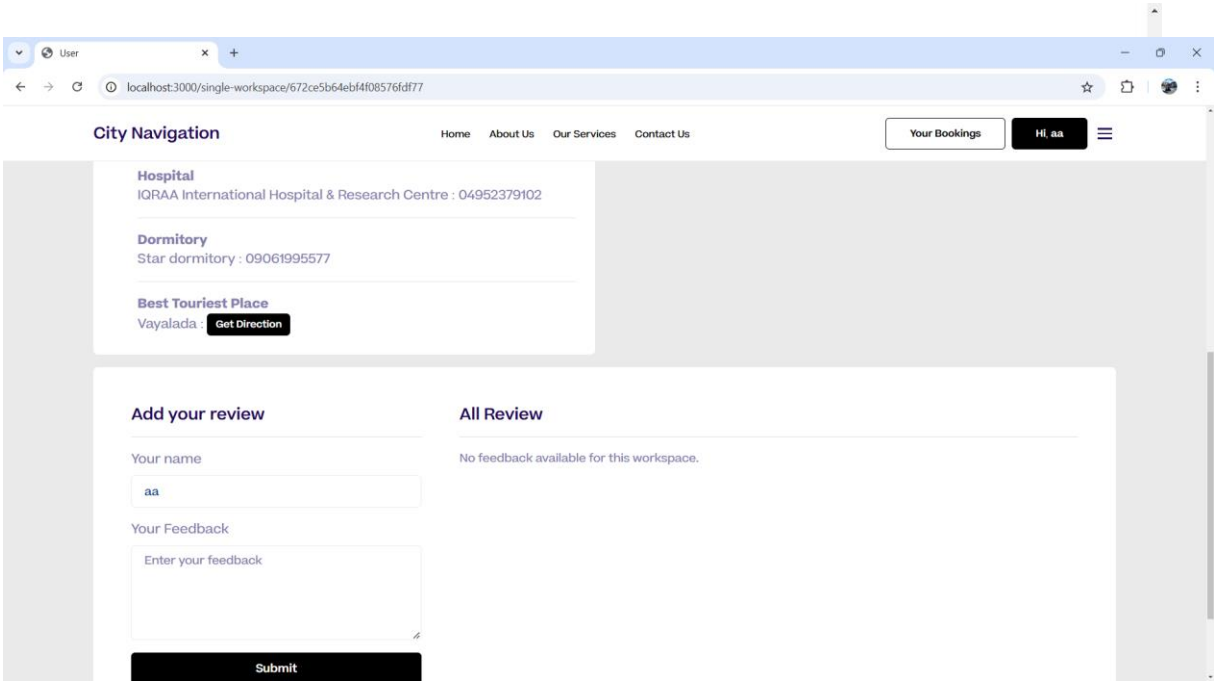
11.1 Screen Shots of Forms

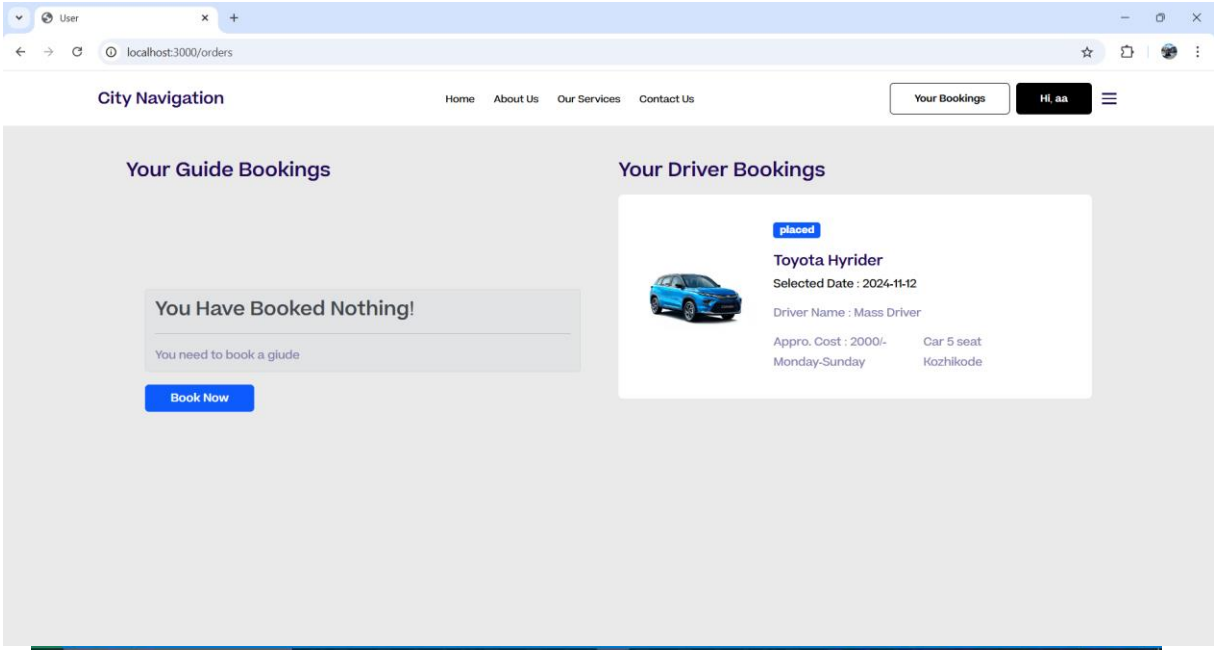




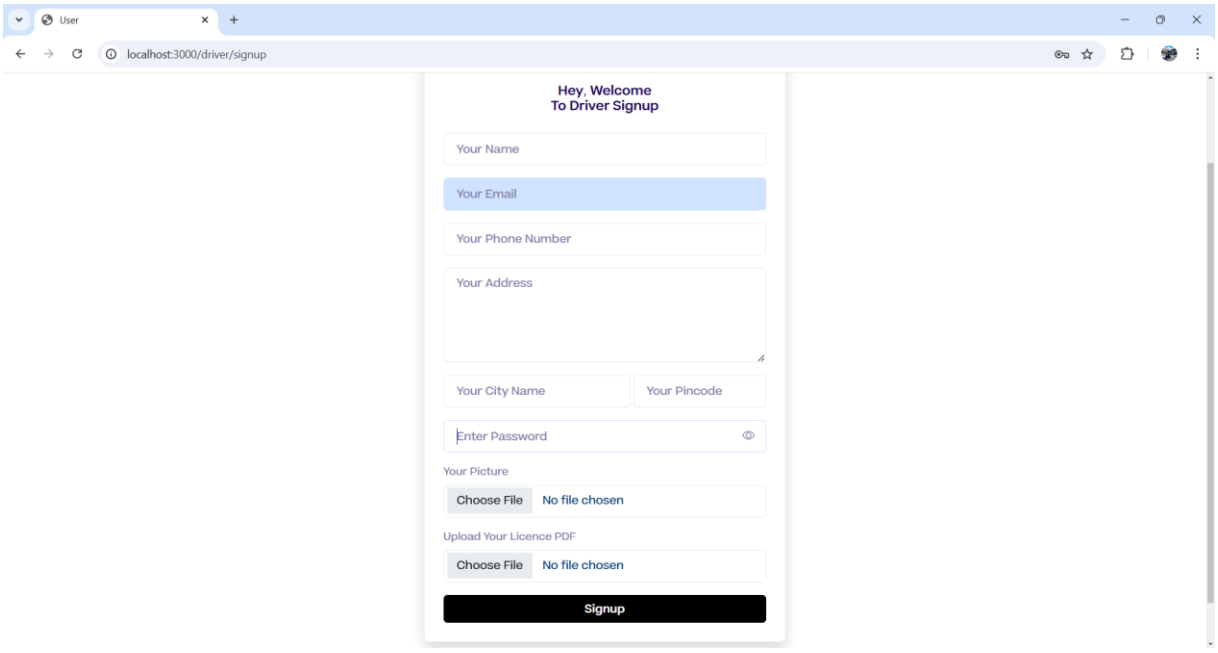

User
localhost:3000
City Navigation
Home
About Us
Our Services
Contact Us
Your Bookings
Hi, aa
Search For A City
"Find Your Ideal City to Elevate Your Experience"
Search For Places...
Kozhikode
Kozhikode is a coastal city in the south Indian state of Kerala. It was a significant spice trade center and is close...
View Details
Thiruvananthapuram
Thiruvananthapuram (or Trivandrum) is the capital of the southern Indian state of Kerala. It's distinguished by its...
View Details
Kollam
Kollam is a city in the state of Kerala, on India's Malabar Coast. It's known as a trade hub and for its beaches, like...
View Details
localhost:3000/#!

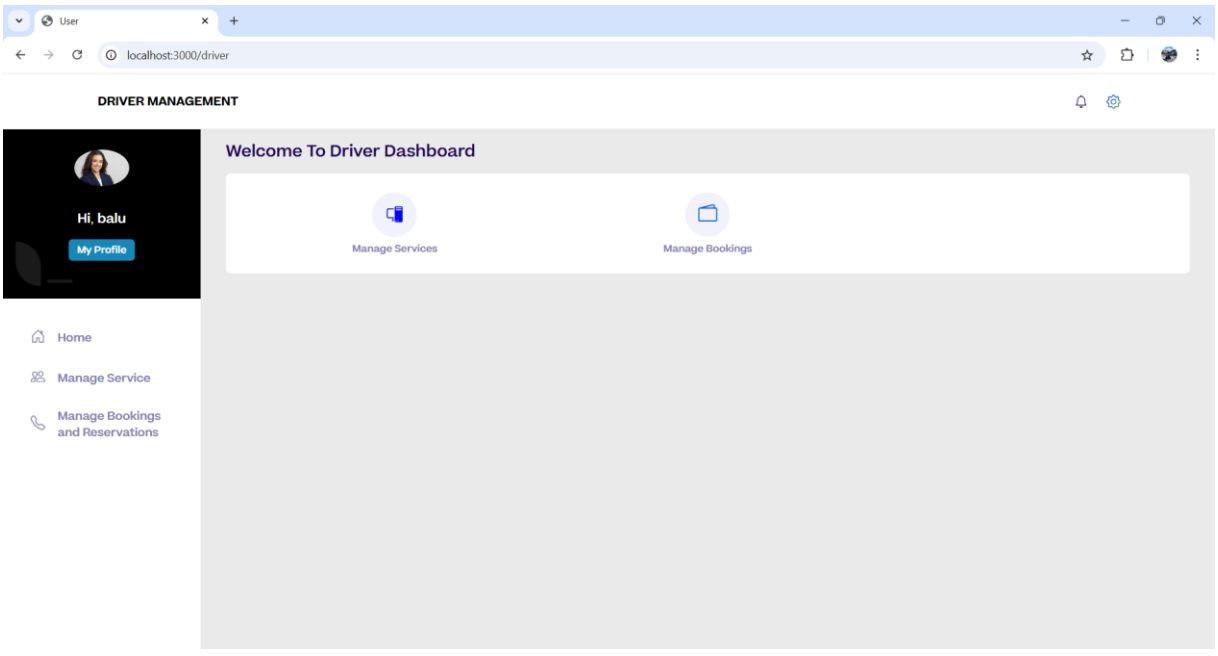

User
localhost:3000/single-workspace/672ce5b64ebf4f08576fdf77
City Navigation
Home
About Us
Our Services
Contact Us
Your Bookings
Hi, aa
Kozhikode
About Kozhikode
Kozhikode is a coastal city in the south Indian state of Kerala. It was a significant spice trade center and is close to Kappad Beach, where Portuguese explorer Vasco da Gama landed in 1498. The central Kozhikode Beach, overlooked by an old lighthouse, is a popular spot for watching the sunset. Inland, tree-lined Mananchira Square, with its musical fountain, surrounds the massive Mananchira Tank, an artificial pond.
More informations
PIN: 673xxx
District: Kozhikode
Elevation: 34.47 m (113.09 ft)
GDP(2020): US$20.773 (equivalent to $24.46 in 2023)Billion
Book a guide?
Services
Police Station
Town Police Station : 04952766433
Restaurent
Paragon Restaurant : 04952767020
localhost:3000/images/workspace-images/672ce5b64ebf4f08576fdf77.png

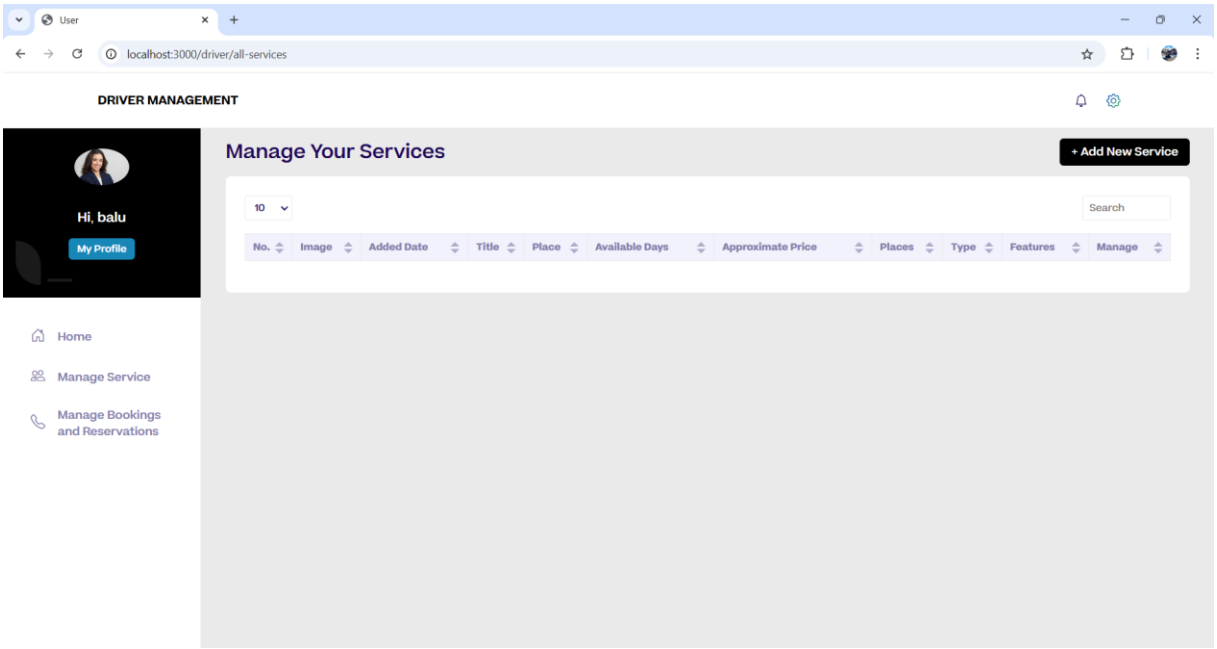

User
localhost:3000/single-workspace/672ce5b64ebf4f08576fdf77
City Navigation
Home
About Us
Our Services
Contact Us
Your Bookings
Hi, aa
Hospital
IQRAA International Hospital & Research Centre : 04952379102
Dormitory
Star dormitory : 09061995577
Best Touriest Place
Vayalada : Get Direction
Add your review
Your name
aa
Your Feedback
Enter your feedback
Submit
All Review
No feedback available for this workspace.

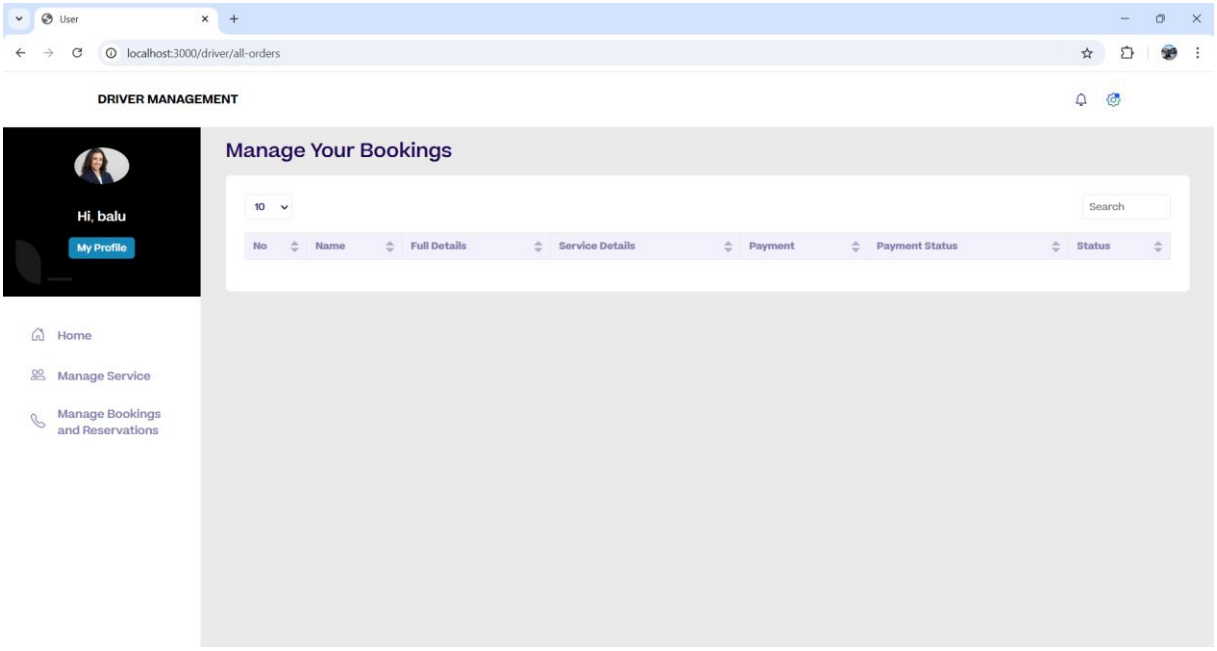

User
localhost:3000/orders
City Navigation
Home
About Us
Our Services
Contact Us
Your Bookings
Hi, aa
Your Guide Bookings
You Have Booked Nothing!
You need to book a giude
Book Now
Your Driver Bookings
placed
Toyota Hyrider
Selected Date : 2024-11-12
Driver Name : Mass Driver
Appro. Cost : 2000/-
Monday-Sunday
Car 5 seat
Kozhikode

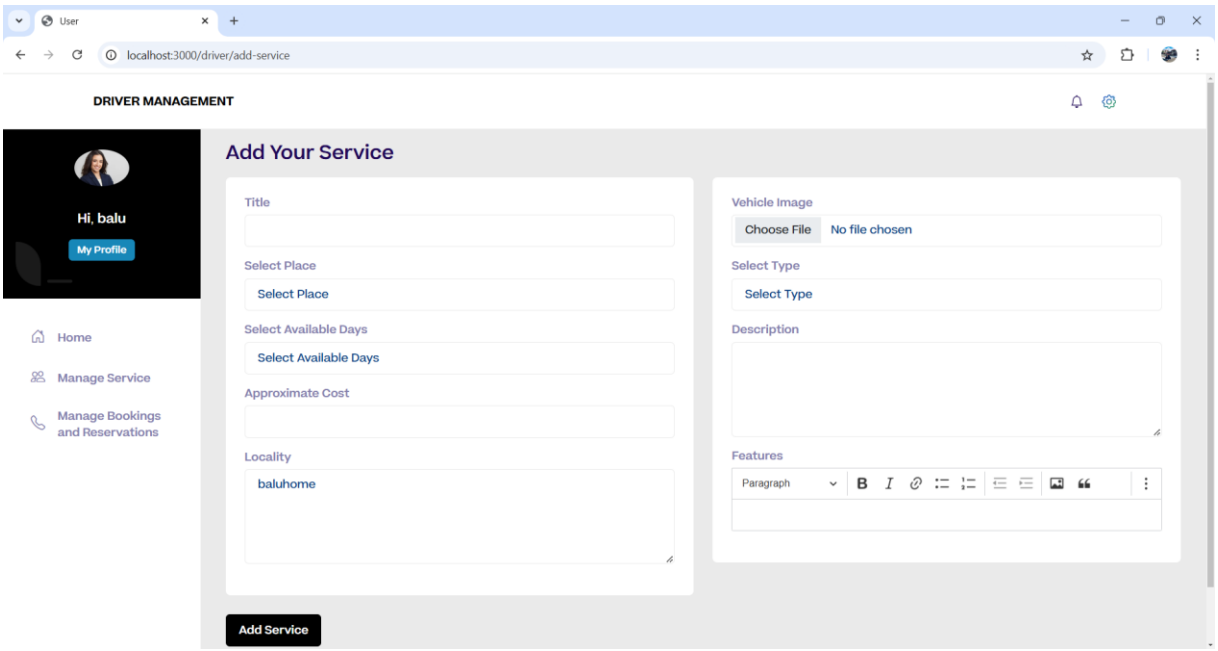
k

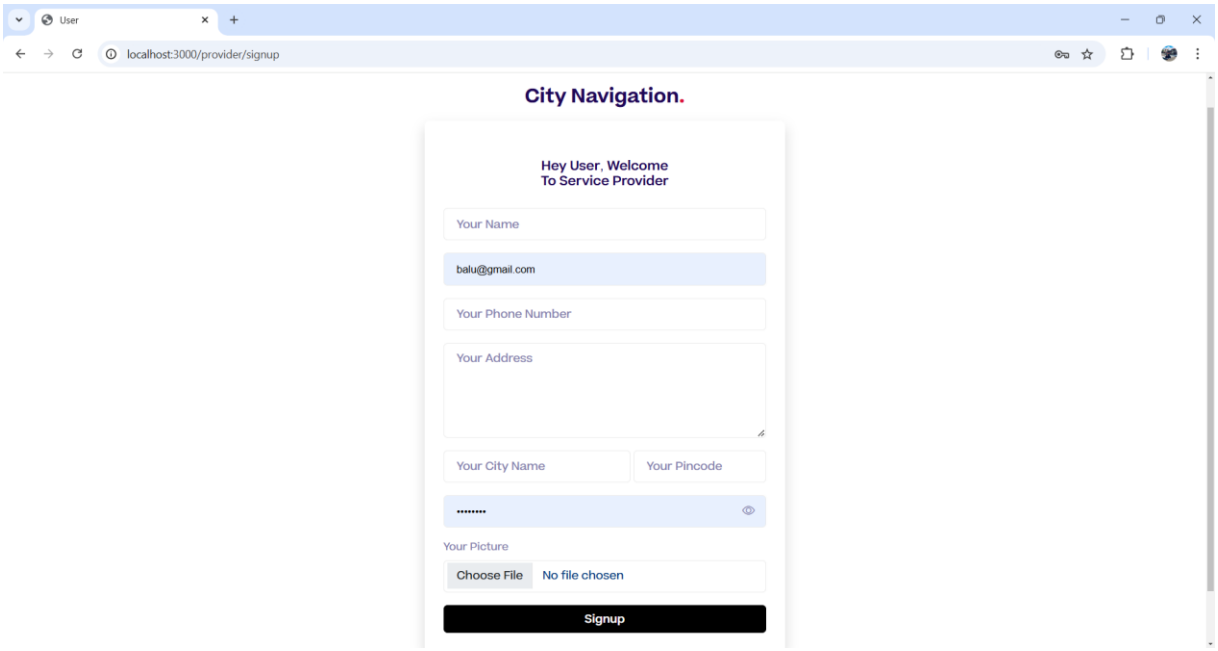

User
localhost:3000/driver/signup
Hey, Welcome
To Driver Signup
Your Name
Your Email
Your Phone Number
Your Address
Your City Name
Your Pincode
Enter Password
Your Picture
Choose File
No file chosen
Upload Your Licence PDF
Choose File
No file chosen
Signup

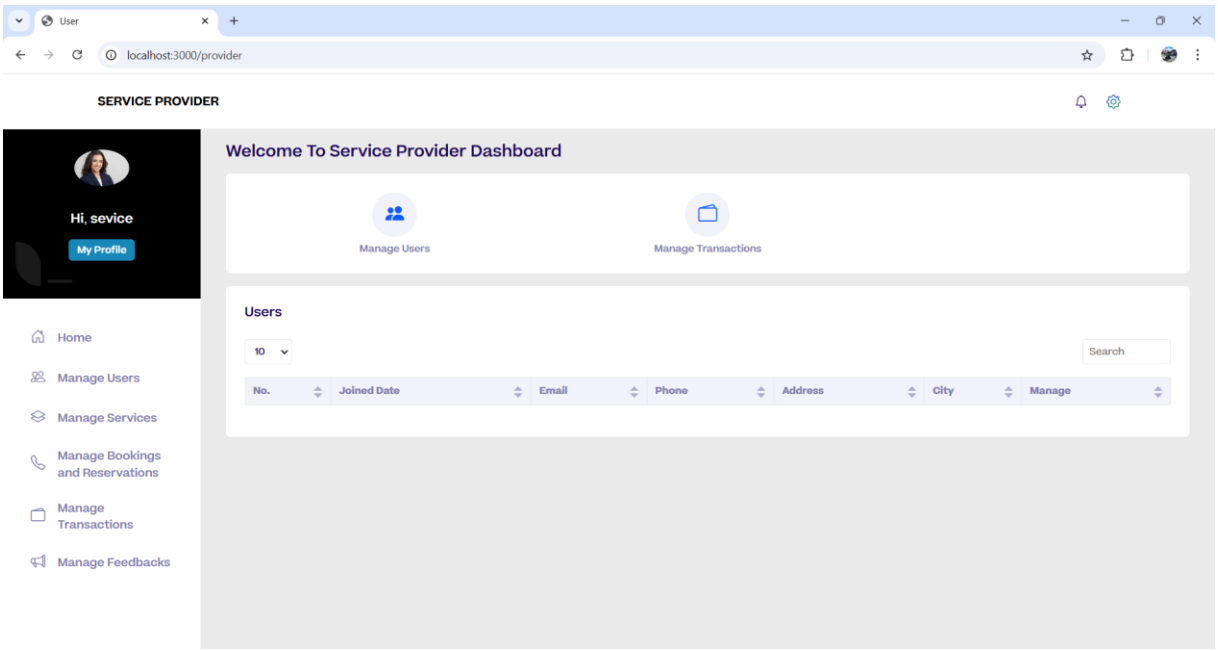

User
localhost:3000/driver
DRIVER MANAGEMENT
Hi, balu
My Profile
Welcome To Driver Dashboard
Manage Services
Manage Bookings
Home
Manage Service
Manage Bookings and Reservations

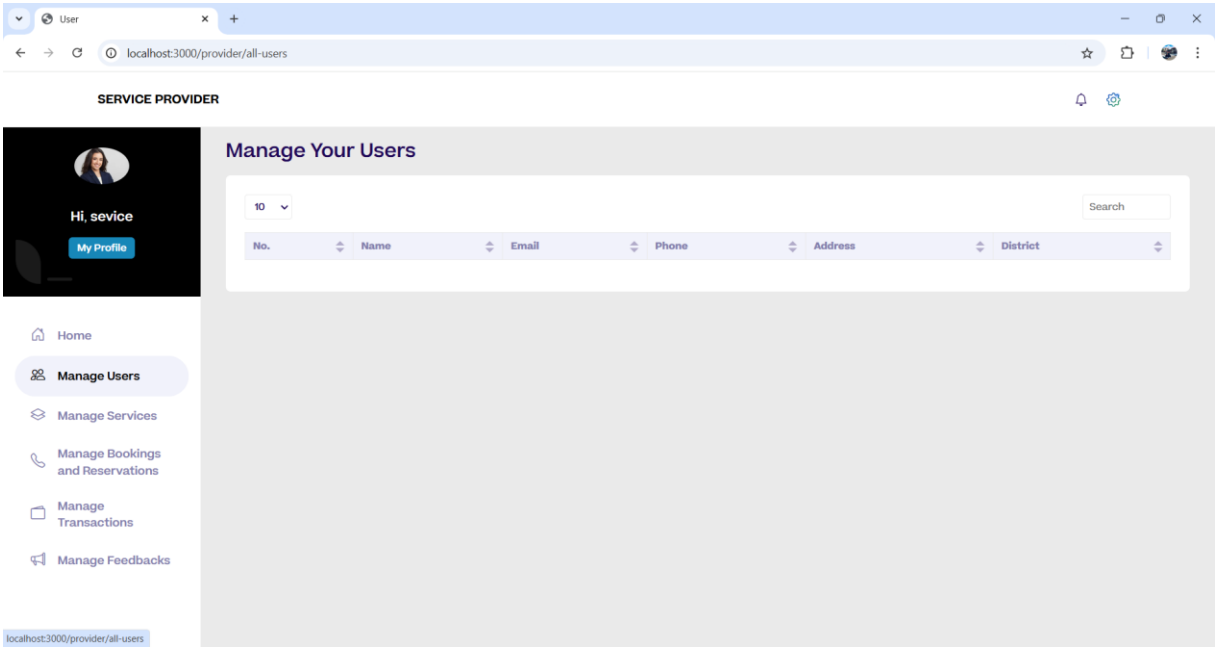

User
localhost:3000/driver/all-services
DRIVER MANAGEMENT
Hi, balu
My Profile
Home
Manage Service
Manage Bookings and Reservations
Manage Your Services
+ Add New Service
10
Search
No.
Image
Added Date
Title
Place
Available Days
Approximate Price
Places
Type
Features
Manage

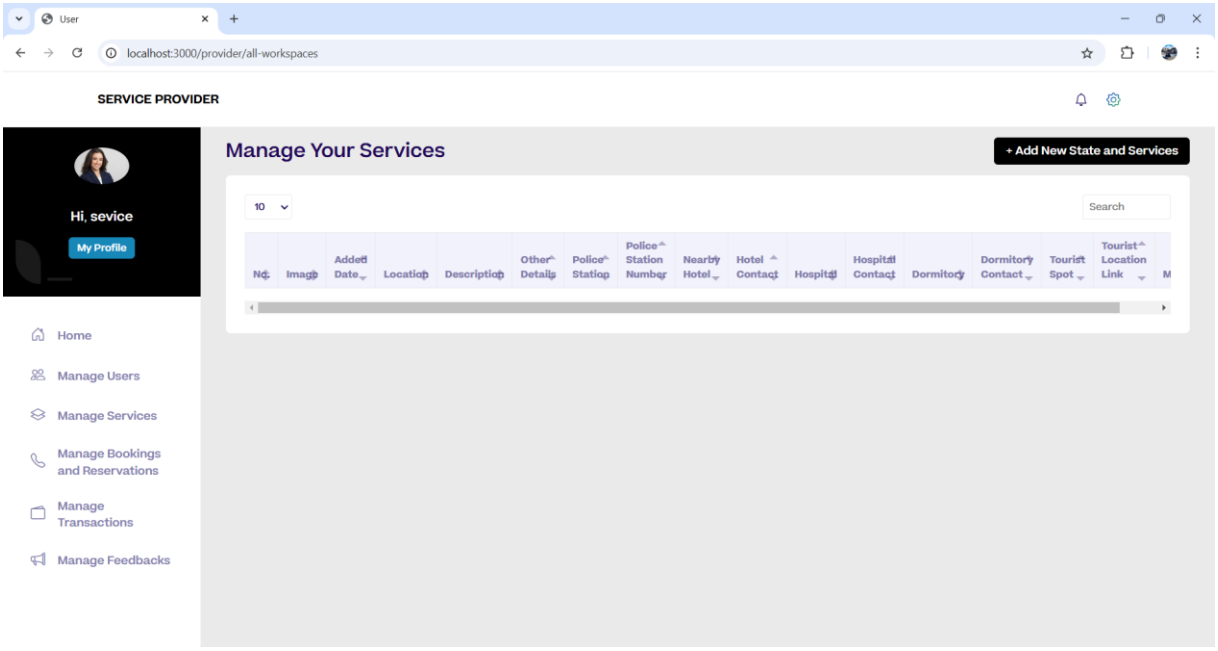

User
localhost:3000/driver/all-orders
DRIVER MANAGEMENT
Hi, balu
My Profile
Home
Manage Service
Manage Bookings and Reservations
Manage Your Bookings
10
Search
No
Name
Full Details
Service Details
Payment
Payment Status
Status

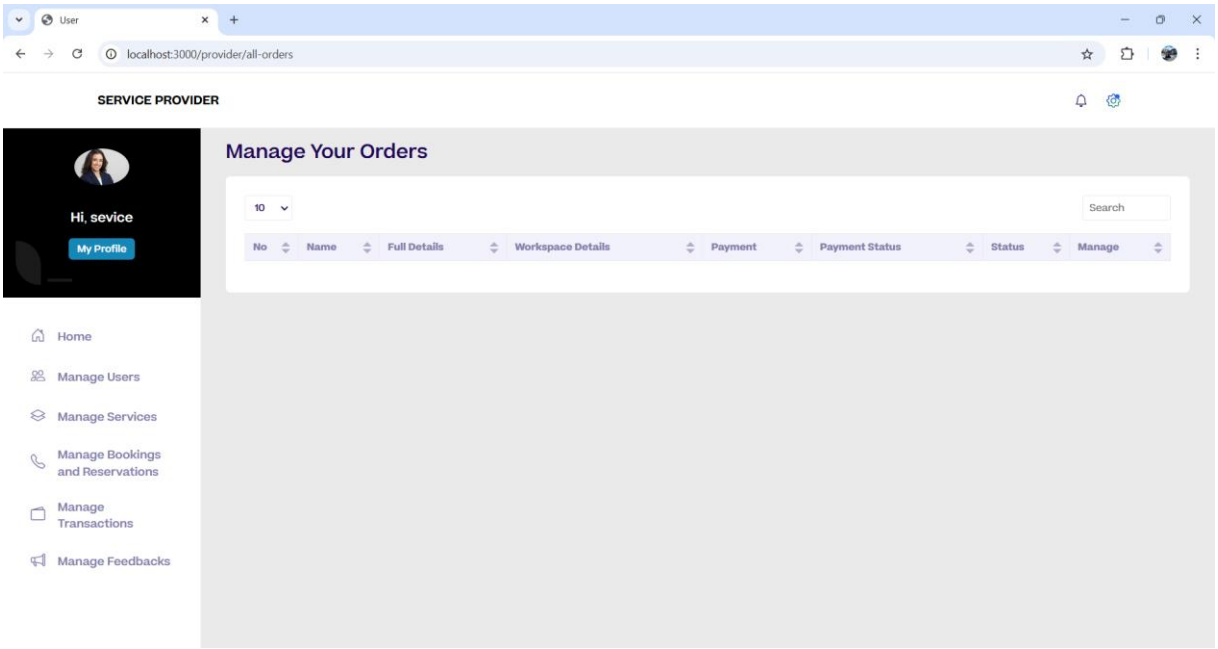

User
localhost:3000/driver/add-service
DRIVER MANAGEMENT
Hi, balu
My Profile
Home
Manage Service
Manage Bookings and Reservations
Add Your Service
Title
Select Place
Select Place
Select Available Days
Select Available Days
Approximate Cost
Locality
baluhome
Vehicle Image
Choose File
No file chosen
Select Type
Select Type
Description
Features
Paragraph
Add Service

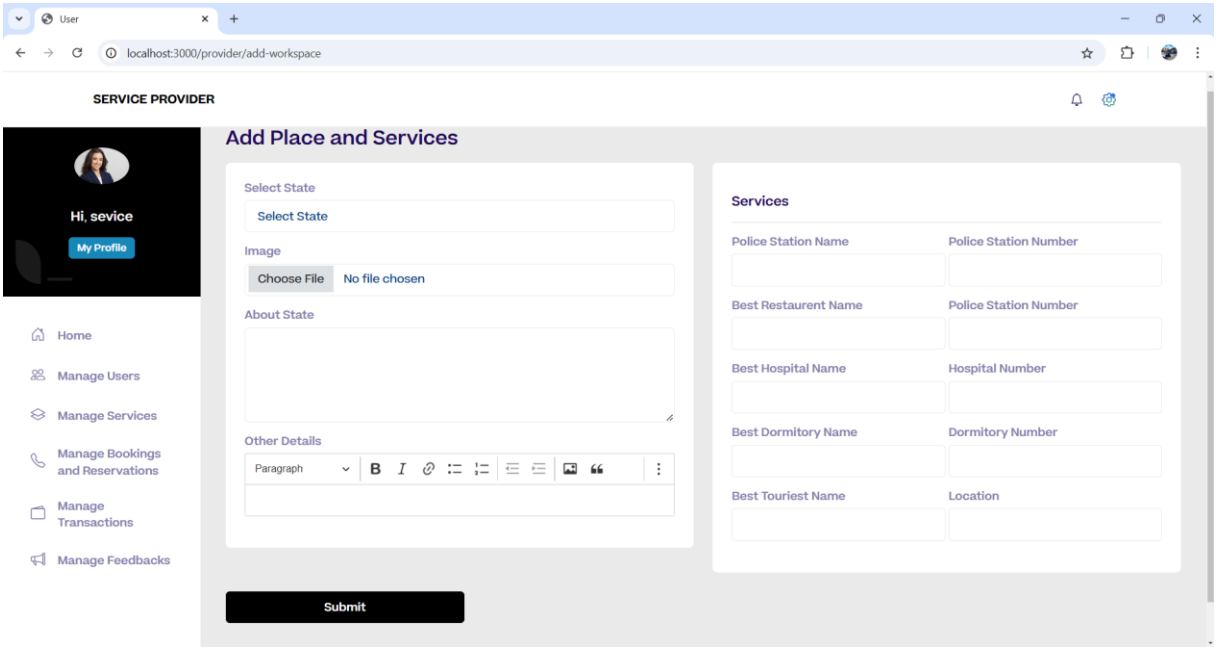

User
localhost:3000/provider/signup
City Navigation.
Hey User, Welcome
To Service Provider
Your Name
balu@gmail.com
Your Phone Number
Your Address
Your City Name
Your Pincode
Your Picture
Choose File
No file chosen
Signup

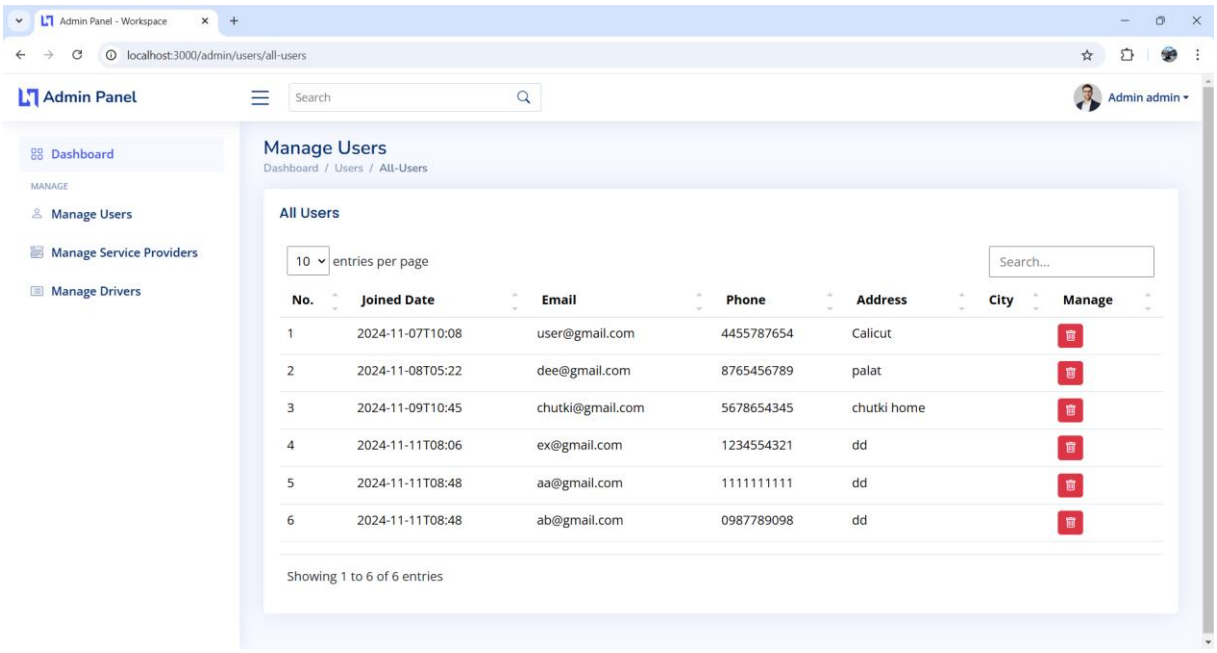

User
localhost:3000/provider
SERVICE PROVIDER
Hi, sevice
My Profile
Welcome To Service Provider Dashboard
Manage Users
Manage Transactions
Users
10
Search
No.
Joined Date
Email
Phone
Address
City
Manage
Home
Manage Users
Manage Services
Manage Bookings and Reservations
Manage Transactions
Manage Feedbacks

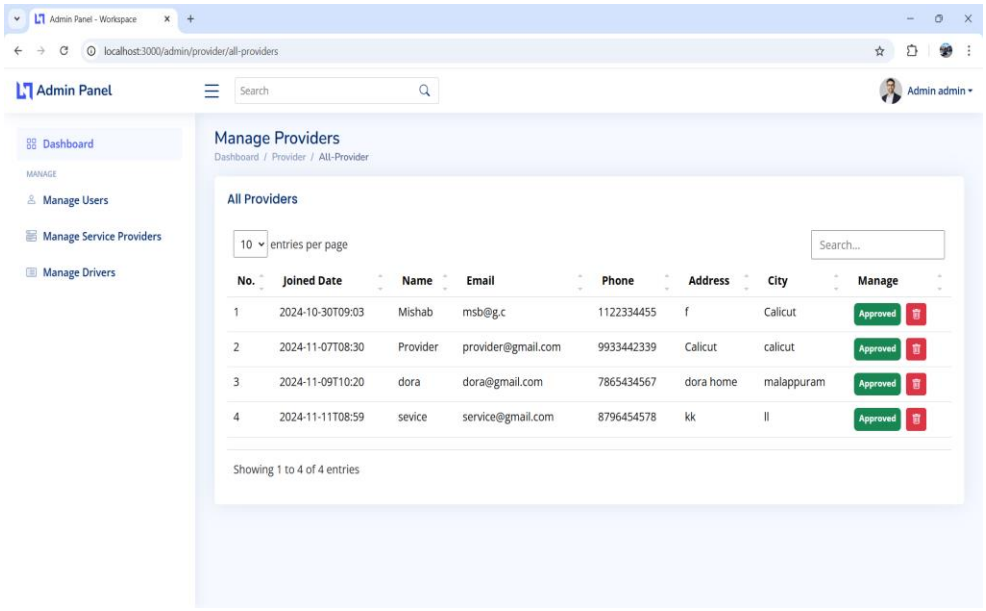

User
localhost:3000/provider/all-users
SERVICE PROVIDER
Hi, sevice
My Profile
Manage Your Users
10
Search
No.
Name
Email
Phone
Address
District
Home
Manage Users
Manage Services
Manage Bookings and Reservations
Manage Transactions
Manage Feedbacks
localhost:3000/provider/all-users

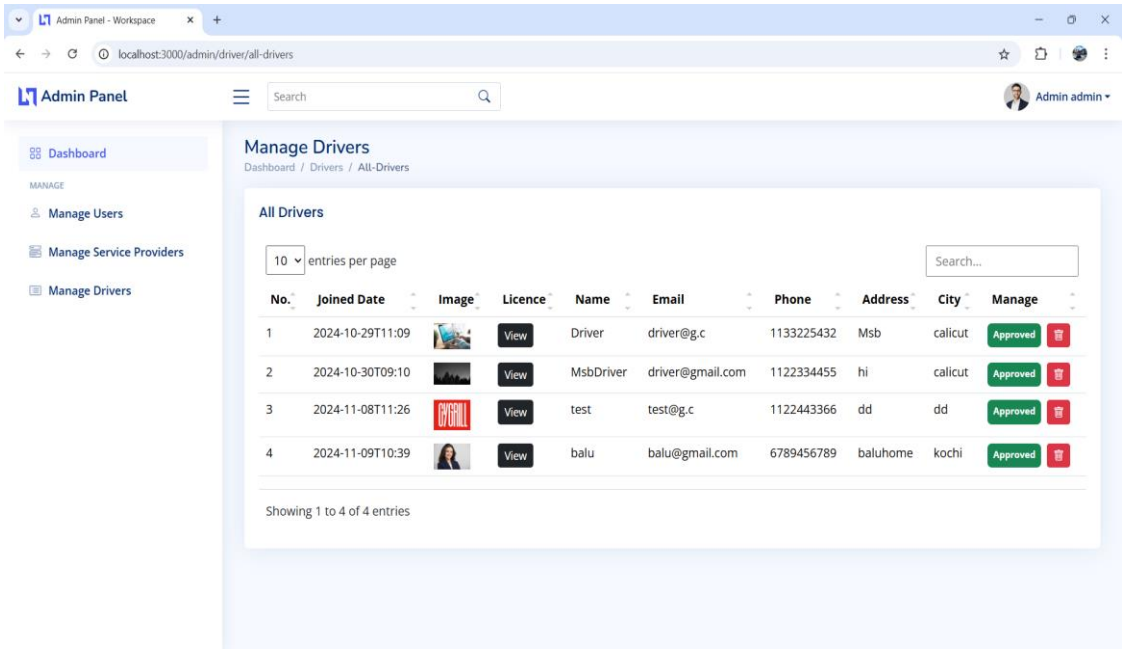

User
localhost:3000/provider/all-workspaces
SERVICE PROVIDER
Hi, sevice
My Profile
Home
Manage Users
Manage Services
Manage Bookings and Reservations
Manage Transactions
Manage Feedbacks
Manage Your Services
+ Add New State and Services
10
Search
No
Image
Added Date
Location
Description
Other Details
Police Station
Police Station Number
Nearby Hotel
Hotel Contact
Hospital
Hospital Contact
Dormitory
Dormitory Contact
Tourist Spot
Tourist Location Link


User
localhost:3000/provider/all-orders
SERVICE PROVIDER
Hi, sevice
My Profile
Home
Manage Users
Manage Services
Manage Bookings and Reservations
Manage Transactions
Manage Feedbacks
Manage Your Orders
10
Search
No
Name
Full Details
Workspace Details
Payment
Payment Status
Status
Manage


User
localhost:3000/provider/add-workspace
SERVICE PROVIDER
Hi, sevice
My Profile
Home
Manage Users
Manage Services
Manage Bookings and Reservations
Manage Transactions
Manage Feedbacks
Add Place and Services
Select State
Select State
Image
Choose File
No file chosen
About State
Other Details
Paragraph
Services
Police Station Name
Police Station Number
Best Restaurent Name
Police Station Number
Best Hospital Name
Hospital Number
Best Dormitory Name
Dormitory Number
Best Touriest Name
Location
Submit


Admin Panel - Workspace
localhost:3000/admin/users/all-users
Admin Panel
Search
Admin admin
Dashboard
MANAGE
Manage Users
Manage Service Providers
Manage Drivers
Manage Users
Dashboard / Users / All-Users
All Users
10 entries per page
Search...
No. Joined Date Email Phone Address City Manage
1 2024-11-07T10:08 user@gmail.com 4455787654 Calicut
2 2024-11-08T05:22 dee@gmail.com 8765456789 palat
3 2024-11-09T10:45 chutki@gmail.com 5678654345 chutki home
4 2024-11-11T08:06 ex@gmail.com 1234554321 dd
5 2024-11-11T08:48 aa@gmail.com 1111111111 dd
6 2024-11-11T08:48 ab@gmail.com 0987789098 dd
Showing 1 to 6 of 6 entries


Admin Panel - Workspace
localhost:3000/admin/provider/all-providers
Admin Panel
Search
Admin admin
Dashboard
MANAGE
Manage Users
Manage Service Providers
Manage Drivers
Manage Providers
Dashboard / Provider / All-Provider
All Providers
10 entries per page
Search...
No. Joined Date Name Email Phone Address City Manage
1 2024-10-30T09:03 Mishab msb@g.c 1122334455 f Calicut Approved
2 2024-11-07T08:30 Provider provider@gmail.com 9933442339 Calicut calicut Approved
3 2024-11-09T10:20 dora dora@gmail.com 7865434567 dora home malappuram Approved
4 2024-11-11T08:59 sevice service@gmail.com 8796454578 kk ll Approved
Showing 1 to 4 of 4 entries


Admin Panel - Workspace
localhost:3000/admin/driver/all-drivers
Admin Panel
Search
Admin admin
Dashboard
MANAGE
Manage Users
Manage Service Providers
Manage Drivers
Manage Drivers
Dashboard / Drivers / All-Drivers
All Drivers
10 entries per page
Search...
No. Joined Date Image Licence Name Email Phone Address City Manage
1 2024-10-29T11:09 View Driver driver@g.c 1133225432 Msb calicut Approved
2 2024-10-30T09:10 View MsbDriver driver@gmail.com 1122334455 hi calicut Approved
3 2024-11-08T11:26 View test test@g.c 1122443366 dd dd Approved
4 2024-11-09T10:39 View balu balu@gmail.com 6789456789 baluhome kochi Approved
Showing 1 to 4 of 4 entries

USERHELPER

```
doSignup: (userData) => {
    return new Promise(async (resolve, reject) => {
      userData.Password = await bcrypt.hash(userData.Password, 10);
      db.get()
        .collection(collections.USERS_COLLECTION)
        .insertOne(userData)
        .then((data) => {
          resolve(data.ops[0]);
        });
    });
  },

  doSignin: (userData) => {
    return new Promise(async (resolve, reject) => {
      let response = {};
      let user = await db
        .get()
        .collection(collections.USERS_COLLECTION)
        .findOne({ Email: userData.Email });
      if (user) {
        bcrypt.compare(userData.Password, user.Password).then((status) => {
          if (status) {
            console.log("Login Success");
            response.user = user;
            response.status = true;
            resolve(response);
          } else {
            console.log("Login Failed");
            resolve({ status: false });
          }
        });
      } else {
        console.log("Login Failed");
        resolve({ status: false });
      }
    });
  },

  getUserDetails: (userId) => {
    return new Promise((resolve, reject) => {
      db.get()
        .collection(collections.USERS_COLLECTION)
        .findOne({ _id: objectId(userId) })
        .then((user) => {
          resolve(user);
        })
        .catch((err) => {
          reject(err);
        });
    });
  },

  updateUserProfile: (userId, userDetails) => {
    return new Promise((resolve, reject) => {
      db.get()
        .collection(collections.USERS_COLLECTION)
        .updateOne(
          { _id: objectId(userId) },
          {
            $set: {
              Fname: userDetails.Fname,
              Lname: userDetails.Lname,
              Email: userDetails.Email,
              Phone: userDetails.Phone,
              Address: userDetails.Address,
              District: userDetails.District,
              Pincode: userDetails.Pincode,
            },
```

```
        }
      )
      .then((response) => {
        resolve();
      })
      .catch((err) => {
        reject(err);
      });
  });
};
```

USERROUTER

```
router.get("/signup", function (req, res) {
  if (req.session.signedIn) {
    res.redirect("/");
  } else {
    res.render("users/signup", { admin: false, layout: 'empty' });
  }
});


router.post("/signup", async function (req, res) {
  const { Fname, Email, Phone, Address, Pincode, District, Password } = req.body;
  let errors = {};

  // Check if email already exists
  const existingEmail = await db.get()
    .collection(collections.USERS_COLLECTION)
    .findOne({ Email });

  if (existingEmail) {
    errors.email = "This email is already registered.";
  }

  // Validate phone number length and uniqueness

  if (!Phone) {
    errors.phone = "Please enter your phone number.";
  } else if (!/^\d{10}$/.test(Phone)) {
    errors.phone = "Phone number must be exactly 10 digits.";
  } else {
    const existingPhone = await db.get()
      .collection(collections.USERS_COLLECTION)
      .findOne({ Phone });

    if (existingPhone) {
      errors.phone = "This phone number is already registered.";
    }
  }
  // Validate Pincode
  if (!Pincode) {
    errors.pincode = "Please enter your pincode.";
  } else if (!/^\d{6}$/.test(Pincode)) {
    errors.pincode = "Pincode must be exactly 6 digits.";
  }

  if (!Fname) errors.fname = "Please enter your first name.";
  if (!Email) errors.email = "Please enter your email.";
  if (!Address) errors.address = "Please enter your address.";
  if (!District) errors.district = "Please enter your city.";

  // Password validation
  if (!Password) {
    errors.password = "Please enter a password.";
  } else {
    const strongPasswordRegex = /^(?=.[a-z])(?=.[A-Z])(?=.\d)(?=.[\W_])[A-Za-z\d\W_]{8,}$/;
    if (!strongPasswordRegex.test(Password)) {
      errors.password = "Password must be at least 8 characters long and include an uppercase letter, a lowercase letter, a number, and
a special character.";
    }
  }
```

```
  if (Object.keys(errors).length > 0) {
    return res.render("users/signup", {
      admin: false,
      layout: 'empty',
      errors,
      Fname,
      Email,
      Phone,
      Address,
      Pincode,
      District,
      Password
    });
  }

  // Proceed with signup
  userHelper.doSignup(req.body).then((response) => {
    req.session.signedIn = true;
    req.session.user = response;
    res.redirect("/");
  }).catch((err) => {
    console.error("Signup error:", err);
    res.status(500).send("An error occurred during signup.");
  });
});


router.get("/signin", function (req, res) {
  if (req.session.signedIn) {
    res.redirect("/");
  } else {
    res.render("users/signin", {
      admin: false,
      layout: 'empty',
      signInErr: req.session.signInErr,
    });
    req.session.signInErr = null;
  }
});


router.post("/signin", function (req, res) {
  const { Email, Password } = req.body;

  if (!Email || !Password) {
    req.session.signInErr = "Please fill in all fields.";
    return res.render("users/signin", {
      admin: false,
      layout: 'empty',
      signInErr: req.session.signInErr,
      email: Email,
      password: Password,

    });
  }

  userHelper.doSignin(req.body).then((response) => {
    if (response.status) {
      req.session.signedIn = true;
      req.session.user = response.user;
      res.redirect("/");
    } else {
      req.session.signInErr = "Invalid Email/Password";
      res.render("users/signin", {
        admin: false,
        layout: 'empty',
        signInErr: req.session.signInErr,
        email: Email
      });
    }
  });
});
```

```
router.get("/signout", function (req, res) {
  req.session.signedIn = false;
  req.session.user = null;
  res.redirect("/");
});
```

APP.JS

```
var createError = require("http-errors");
var express = require("express");
var path = require("path");
var cookieParser = require("cookie-parser");
var logger = require("morgan");
var hbs = require("express-handlebars");
var usersRouter = require("./routes/users");
var adminRouter = require("./routes/admin");
var driverRouter = require("./routes/driver");

var providerRouter = require("./routes/provider");
var fileUpload = require("express-fileupload");
var db = require("./config/connection");
var session = require("express-session");
var app = express();
const Handlebars = require('handlebars');


// view engine setup
app.set("views", path.join(__dirname, "views"));
app.set("view engine", "hbs");

Handlebars.registerHelper('eq', function (a, b) {
  return a === b;
});

// Set up Handlebars engine with helpers
app.engine(
  "hbs",
  hbs({
    extname: "hbs",
    defaultLayout: "layout",
    layoutsDir: __dirname + "/views/layout/",
    partialsDir: __dirname + "/views/header-partials/",
    helpers: {
      incremented: function (index) {
        index++;
        return index;
      },
      eq: function (a, b) {
        return a === b;
      },
      NanValue: function (amount) {
        return isNaN(amount) ? 0 : amount;
      }

    },
  })
);



app.use(logger("dev"));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, "public")));
app.use(fileUpload());
app.use(
  session({
```

```
    secret: "Key",
    resave: true,
    saveUninitialized: true,
    cookie: { maxAge: 24 * 60 * 60 * 1000 }, // 1 day in milliseconds
  })
);
db.connect((err) => {
  if (err) console.log("Error" + err);
  else console.log("Database Connected Successfully");
});
app.use("/", usersRouter);
app.use("/admin", adminRouter);
app.use("/provider", providerRouter);
app.use("/driver", driverRouter);
app.use("/admin/users", adminRouter);
app.use("/admin/provider", adminRouter);
app.use("/admin/driver", adminRouter);


// catch 404 and forward to error handler
app.use(function (req, res, next) {
  next(createError(404));
});

// error handler
app.use(function (err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get("env") === "development" ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render("error");
});

module.exports =
```