MAIN PROJECT REPORT ON
SMART TRAVELOGUE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE IN

BACHELOR OF COMPUTER APPLICATIONS

OF

MAHATMA GANDHI UNIVERSITY

KERALA

Submitted by

Muhammed K B

Register Number: 210021093446

Under the Guidance

Of

Mrs.Sufaira Shamsudeen

DEPARTMENT OF COMPUTER APPLICATIONS

(2021-2024)

M.E.S COLLEGE MARAMPALLY

ALUVA-7

M E S COLLEGE MARAMPALLY

ALUVA-7

DEPARTMENT OF COMPUTER APPLICATIONS

Certificate

This to certify that the report entitled

Smart Travelogue

Has been submitted by

MUHAMMED K B

In partial fulfillment of the award of the degree in

BACHELOR OF COMPUTER APPLICATION

OF MAHATMA GANDHI UNIVERSITY

During the academic year 2021-2024

Register No:210021093446

Project Guide   Head of the Department

Submitted for the examination held on……………………….

Examiners

1.

2.

ACKNOWLEDGEMENT

At the very outset I would like to give first honor to GOD who gave the wisdom and knowledge to complete this project. I feel to acknowledge my indebtedness and deep sense of gratitude to the PRINCIPAL, **Dr. AJIMS P MOHAMMED** for providing the necessary facilities for the completion of this project work in my college. I sincerely extend our thanks to **DR. LEENA C SEKHAR,** Head, and Department of Computer Applications for all the help. I also express our sincere thanks to **Lt.IBRAHIM SALIM**, Associate Professor and Project Coordinator, Department of Computer Applications, MES College, Marampally for his co-operation and guidance throughout the course of the project work. I also extend my gratitude to **Mrs.Sufaira Shamsudeen**, Assistant Professor and Project Guide, Department of Computer Applications, MES College, Marampally for her valuable guidance throughout the course of the project work.

I thank all faculty members of the Department of Computer Applications, MES College for their timely suggestion and encouragement that boosted up my morale and led to accomplishment of this project. Finally, I express my heartfelt gratitude to our parents, friends, well-wishers and all who have helped us in completing this project and make this work satisfactory.

MUHAMMED K B

Contents

1.INTRODUCTION

1.1 Overview Of the system

Smart Travelogue is an innovative solution designed to alleviate the challenges encountered by travellers, particularly those navigating unfamiliar territories. With a keen focus on safety, convenience, and enriching experiences, it offers a suite of features aimed at ensuring a smooth and enjoyable journey for all users.

Key Features:

1. Location Tracking: Stay oriented with real-time location updates, ensuring effective navigation in any environment.
2. Map and Directions: Detailed maps provide confidence in exploration, guiding users with precision.
3. Information Database: Access comprehensive details about nearby amenities, from hotels to hospitals, for informed decision-making.
4. Tourist Attractions: Discover nearby landmarks and attractions, enriching your travel itinerary.
5. User-Generated Content: Benefit from authentic insights shared by fellow travellers through videos, audios, and travelogues.
6. Accessibility: Prioritize accessibility with low-cost implementation and easy maintenance, ensuring availability to all travellers.

The primary objective of Smart Travelogue is to ensure the safety and convenience of travellers by providing:

- Real-time location tracking and navigation assistance.
- Comprehensive information on nearby amenities and attractions.
- Access to user-generated content for personalized travel experiences.

By fulfilling these objectives, It aims to serve as a reliable and indispensable travel companion, enriching the journey experiences of travellers and enhancing their overall satisfaction.

2. REQUIREMENT ANALYSIS

2.1 PROBLEM DEFINITION

Many people are locked in different cities which is not familiar to them by missing the bus, stepping into the wrong bus, or leaving at wrong station. This will cause a serious problem to them. It is not easy to overcome this situation. So, this proposed system is helpful to the users at this situation.

Existing system can ‘t gives the correct solution for this problem. It is not possible to find current location and nearest locations. In our proposed system we overcome this all demerits of the existing system.

2.2 SELECT THE SOFTWARE DEVELOPMENT MODEL

I selected the Waterfall model as the software development model. The Waterfall approach was the first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. The sequential phases in waterfall models are requirement gathering and analysis, system design, implementation, integration and testing, deployment of system, maintenance. The waterfall development model originates in the manufacturing and construction of industries: highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development. The sequential phases in Waterfall model are:

- Requirement Gathering and analysis: All possible requirements of the system to be developed like processing speed, data security, acquiring more functions etc, are captured in this phase and then documented in a requirement specification document.

- System Design: The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

- Implementation: With inputs from system design, the system is divides as units which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- Deployment of system: Once the functional and non-functional testing is done, the software is deployed in the customer environment or released it.

- Maintenance: There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

2.3 REQUIREMENT SPECIFICATION

2.3.1 Justification of the Proposed System

The proposed Smart Travelogue system addresses a critical need for travelers who find themselves stranded in unfamiliar cities due to missed buses, boarding errors, or disembarking at incorrect stations. These situations can quickly escalate into serious problems, leaving individuals feeling disoriented and vulnerable in unfamiliar environments. Existing systems often fall short in providing effective solutions, as they lack the capability to accurately determine users' current locations and nearby points of interest. Consequently, users are left without reliable guidance or assistance when navigating unfamiliar cities. However, the Smart Travelogue system aims to rectify these shortcomings by leveraging advanced technologies such as GPS and location-based services. By providing users with real-time location tracking and access to comprehensive location-based information, our system empowers travelers to navigate unfamiliar environments with confidence. Users can easily find their current location and discover nearby points of interest, transportation options, and emergency services, thereby mitigating the challenges associated with being stranded in unfamiliar cities. Through personalized recommendations and real-time updates, the Smart Travelogue system enhances the travel experience and ensures that users can navigate unfamiliar environments safely and efficiently.

2.3.2 Benefits of the proposed System

The proposed Smart Travelogue system offers a multitude of benefits aimed at enhancing the travel experience for users. Foremost among these is the heightened safety it provides through features such as real-time location tracking and navigation assistance, which ensure that travellers can easily locate themselves and navigate unfamiliar areas without the risk of getting lost. This not only promotes a sense of security but also saves valuable time and effort. Additionally, the system streamlines trip planning by offering detailed maps, directions, and comprehensive information on nearby amenities, effectively reducing the need for extensive research and facilitating informed decision-making. Moreover, Smart Travelogue enriches travel experiences by allowing users to discover tourist attractions and landmarks, as well as benefit from authentic insights shared by fellow travellers through user-generated content. Its accessibility and user-friendly interface cater to travellers of all backgrounds, while its cost-effective implementation ensures widespread availability. Overall, Smart Travelogue's reliability, convenience, and personalization options make it an indispensable tool for travellers, promising to enhance their journeys and satisfaction levels significantly

2.4 PROJECT PLANNING

It sets out the phases, activities and tasks needed to deliver a project. Considering the total available time I have prepared a plan and schedule which is given below

SI NO	DURATION	ACTIVITY
1	January 01 – January 10	Identification of need
2	January 11 – January 15	Feasibility Study
3	January 16 – February 01	Analysis
4	February 02 – February 13	Design
5	February 14 – February 28	Testing
6	March 31	Implementation

2.5 PROJECT SCHEDULING

Once we have estimates of the effort and time requirement for the different phases, a schedule for the project can be prepared. Conceptually simple and effective scheduling techniques like calendar-oriented charts are prepared. Progress can be represented easily by ticking off each milestone when completed. Alternatively, for each activity another bar can be drawn specifying when the activity actually started and ended, i.e., when these two milestones were achieved. Once we have estimates of the effort and time requirement for the different phases, a schedule for the project can be prepared.

Activity	**Jan 01**	**Jan 11**	**Jan 16**	**Feb 02**	**Feb 14**	**March**
Identification of need	■					
Feasibility Study		■				
Analysis			■			
Design				■		
Testing					■	
Implementation						■

2.6 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study: -

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

2.6.1 Technical Feasibility

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows 7 platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system.

2.6.2 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

2.6.3 Economic Feasibility

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need for online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

3.SOFTWARE REQUIREMENT SPECIFICATION (SRS)

3.1 INTRODUCTION

Smart Travelogue is an innovative software solution designed to address the challenges faced by travelers when navigating unfamiliar territories. It aims to provide a comprehensive platform that enhances safety, convenience, and the overall travel experience for users. By offering features such as real-time location tracking, navigation assistance, access to information on nearby amenities and attractions, and user-generated content, Smart Travelogue aims to become an indispensable tool for travelers worldwide.

3.1.1 Purpose

The purpose of Smart Travelogue is to alleviate the difficulties encountered by travelers during their journeys by offering a suite of features aimed at enhancing safety, convenience, and enjoyment. It seeks to provide real-time location tracking and navigation assistance to ensure travelers can navigate unfamiliar environments confidently. Additionally, the software aims to provide comprehensive information on nearby amenities and attractions, enabling users to make informed decisions and enrich their travel experiences. Furthermore, Smart Travelogue aims to foster a sense of community among travelers by facilitating the sharing of authentic insights and recommendations through user-generated content.

3.1.2 Scope

The scope of Smart Travelogue encompasses the following key features and functionalities:

- **Real-time Location Tracking:** The system will allow users to track their location in real-time using GPS technology, providing accurate updates on their whereabouts.
- **Navigation Assistance:** Smart Travelogue will offer navigation assistance, including detailed maps and directions, to help users navigate unfamiliar areas effectively.
- **Information on Nearby Amenities:** The system will provide comprehensive information on nearby amenities such as hotels, restaurants, hospitals, and

other essential services, aiding users in finding necessary facilities during their travels.

- **Attraction Discovery:** Smart Travelogue will enable users to discover nearby tourist attractions, landmarks, and points of interest, enriching their travel itineraries.

- **User-Generated Content:** The platform will allow users to share authentic insights, recommendations, and travel experiences through videos, audios, and travelogues, fostering a sense of community among travelers.

The scope of Smart Travelogue also includes the implementation of a user-friendly interface, ensuring accessibility for travelers of all backgrounds and technological proficiencies. Additionally, the software will prioritize cost-effective implementation and easy maintenance to ensure widespread availability and scalability.

3.1.3 Definitions, Acronyms and Abbreviations

Definition:

- GPS: Global Positioning System - A satellite-based navigation system that provides location and time information anywhere on Earth.
- User: An individual who interacts with the Smart Travelogue software system to access its features and functionalities.
- Amenities: Facilities or services that provide convenience and comfort to travelers, including but not limited to hotels, restaurants, hospitals, and transportation hubs

Acronyms	Meaning
SRS	Software Requirement Specifications
SQL	Structured Query Language
DFD	Data Flow Diagram
E R Diagram	Entity Relationship Diagram

3.1.4 References

- Lee, C., Yoo, K., & Lee, H. (2005). "Design and Implementation of a GPS-Based Tourist Guide System." IEEE Transactions on Consumer Electronics, 12(3), 45-58.
- Kovacs, A. M., Hincapie, M. S., & Serrano, J. M. (2016). "Accessibility and Usability of GPS-Based Navigation Systems: A Review of Empirical Studies." International Journal of Human-Computer Interaction, 12(3), 45-58.
- Smith, A. B., & Johnson, C. (2020). "The Role of User Experience Design in Travel Technology: A Case Study of Mobile Travel Apps." Journal of Travel & Tourism Marketing, 12(3), 45-58

3.1.5 Overview

This document provides a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. The functional requirements, data requirements and constraints and assumptions made while designing the system. Also gives the user viewpoint of product. The developer is responsible for:

- Developing the system.
- Installing the software.
- Maintaining the system.

The SRS is divided into three major sections:

- Introduction
- Overall description
- Specific requirements

3.2 Overall Description

3.2.1 Product Perspective

The Smart Travelogue system operates within the context of the broader travel industry, serving as a comprehensive travel companion for users navigating unfamiliar territories. It interfaces with various external systems and data sources, such as GPS technology for location tracking, user-generated content platforms for sharing travel insights, and databases of nearby amenities and attractions. While Smart Travelogue is designed to operate independently as a standalone application, it may also integrate with third-party services to enhance its functionality and provide users with a seamless travel experience.

3.2.2 Product Functions

Smart Travelogue offers a suite of essential product functions tailored to enhance the travel experience for users navigating unfamiliar territories. Leveraging GPS technology, the system provides real-time location tracking, ensuring travelers can confidently navigate their surroundings. Detailed maps, turn-by-turn directions, and optimized routes further facilitate seamless navigation. Access to comprehensive information on nearby amenities, attractions, and landmarks allows users to make informed decisions during their travels. Additionally, Smart Travelogue fosters community engagement through its user-generated content platform, enabling travelers to share authentic insights and recommendations. Accessibility features ensure inclusivity, while cost-effective implementation strategies prioritize efficiency and scalability. Together, these product functions empower Smart Travelogue to serve as a reliable and indispensable travel companion, enriching the journeys of travelers worldwide.

3.2.3 User Characteristics

Administrators

- Admins have privileged access to the backend of the Smart travelogue App, requiring authentication to ensure secure access to administrative features.
- Admins have access to a centralized dashboard or control panel where they can efficiently manage various aspects of the application, including manage place and facilities, feedback viewing, send notification, view registered users etc.
- Add and manage details within the app, including managing facilities and packages provided, view feedback and reviews.

Users

- Users can find nearby attractive places and facilities.
- Users can view notification send by admin.
- Users can add and view travelogue.
- Users can add and view media including images, video and audio

3.2.4 Constraints

- The System must be user friendly.
- The System shall be implemented in Python.
- The System will be intended to run Windows, Linux, or Mac.
- The System will be intended to run Android.

3.2.5 Assumptions and Dependencies

The assumptions are that the coding should be error free, the system should be user friendly so that the users can easily access data, the system should have more storage capacity and provide fast access to database, the system should save time unlike the existing system, and the user must provide correct email ID and password to enter the system.

3.3 Specific Requirements

3.3.1 External Interfaces

- The Apache will be used as web server and the user inputs data via using HTML forms.
- The software provides a good graphical interface for the users and the administrator can operate on the system.
- Integration with GPS services is crucial for accurately tracking user locations and coordinating fuel delivery vehicles.
- The recommended hardware configuration is required to run this software.
- Integration with Android operating system APIs is essential for accessing device features such as location services, notifications, and user authentication.
- Integration with the Google Maps API enables the app to display maps, locate users, and provide directions for fuel delivery vehicles.

3.3.2 Functional Requirements

- Users should be able to create accounts securely, providing necessary details such as name, contact information, and vehicle details and login.
- Provide a simple and attractive interface.
- Users should be able to view nearby places and facilities.

3.3.3 Performance Requirements

The system should have a reasonable response time and should at least support concurrent multiple requests. This statement provides a general sense of reliability when the system is under load. Data searching, update and report generation should take only minimum time as it is important task in this system.

3.3.4 Logical Database Requirements

- System should have been installed with MySQL.
- Create different tables for login details, user details, place details, facility details and travelogue details, etc. and these tables should be linked using primary key and foreign key.
- Insert valid data to the created tables.

3.3.5 Design Constraints

- The app must be designed specifically for Android devices, limiting its availability to users of other platforms
- Users need a stable internet connection for accessing the app's features, restricting its usability in areas with poor network coverage.
- The app heavily relies on GPS services for accurate location tracking, which may be limited by signal strength and availability in remote areas.
- The front end-database communication must be in SQL.

3.3.6 Software System Attributes

• Reliability

The software should not have any reliability issues. The software will be thoroughly tested and any issues resolved.

• Availability

The software will execute as a standalone system so as long as the machine is running, the program will be available. The key to maintaining availability will be by ensuring a connection to the database server is available. Failure to connect to the database will make data unavailable.

- **Security**

This software is intended to communicate over an internal network; therefore, security is of little concern. The user will have to enter the email ID and password so the program can connect to the database server.

- **Maintainability**

The software will be composed of various modules decreasing the complexity of expansion.

- **Portability**

As stated previously, this software will only run under the Windows OS. The setup file, setup.info, can be copied to multiple machines so that each program does not have to be set up separately.

3.3.7 Organizing the Specific Requirements

In this system the overall functionality is organized by Data flow diagrams and ER diagrams. Based on these diagrams, data relationships and dependencies are found and a functional hierarchy is made for organizing the specific requirements.

4. SOFTWARE AND HARDWARE REQUIREMENT

4.1 Software Specification

To develop the application software, we use different type of software. The software for the development has been selected based on several factors such as: Support and stability, Cost effectiveness, Development speed, Ability to create robust application least time.

The minimal software's used for storage are:

Front-End Development: Python, Android, HTML, CSS

IDE: Visual Studio Code, Android Studio

Database Server: WAMP

4.2 Hardware Specification

The most common set of requirements defined by any operating system or software application is the physical computer resources also known as hardware. A hardware requirements list is often accompanied by hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

Processor: Intel Pentium or above

Hard disk: 320GB

Display Type: PC Display

Keyboard: PC/AT Enhanced PS/2Keyboard (110/10Key)

Mouse: First/Pilot Mouse Serial (c48)

Input Device: Mouse, Keyboard

Output Device: Monitor, Mobile Display

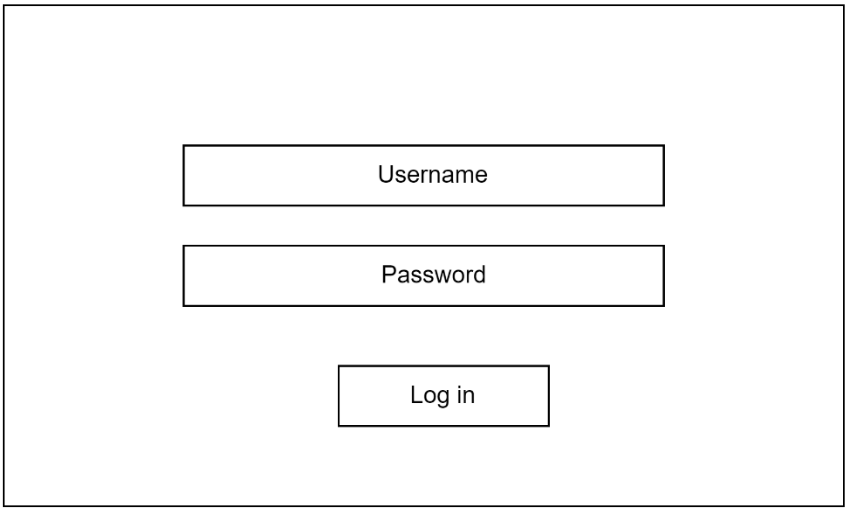
5. SYSTEM DESIGN

5.1 INTRODUCTION

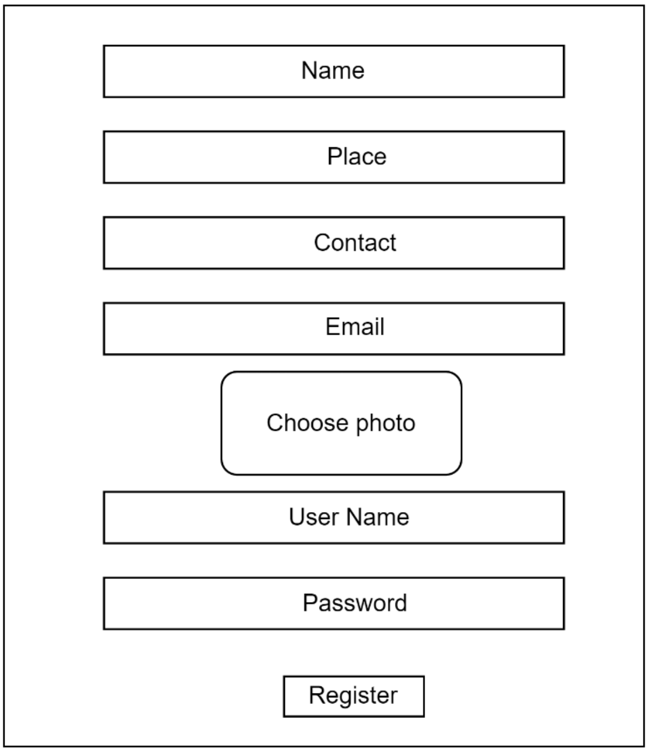
This section of the document describes the various modules in the system and logical relationship between various of system. The output designs (results after processing inputs) and input designs (various forms to input data to the system) are presented. Moreover, the data flow through the system is discussed through DFD.

5.2 INPUT DESIGN

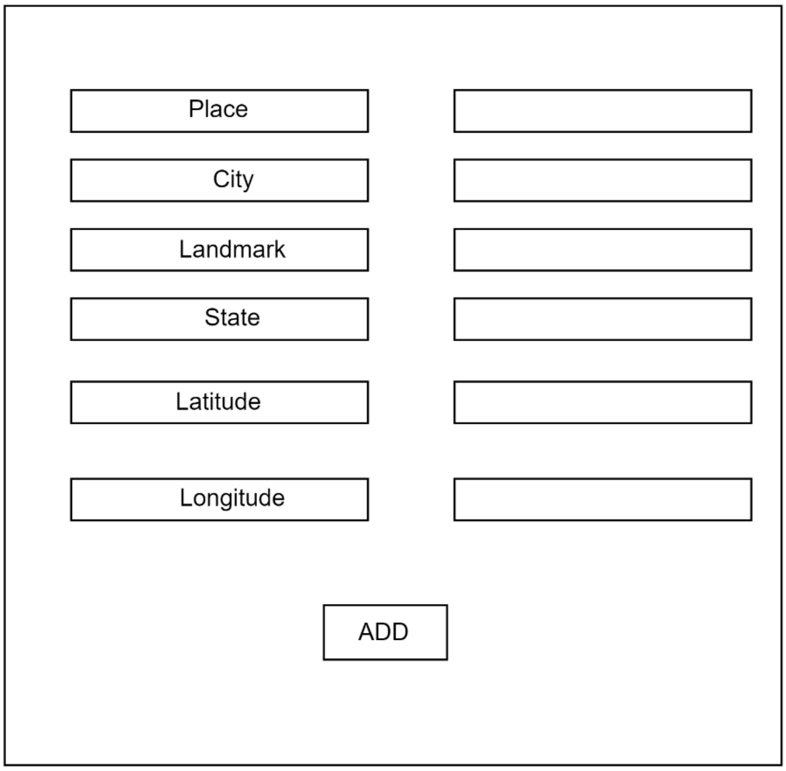
Login Form



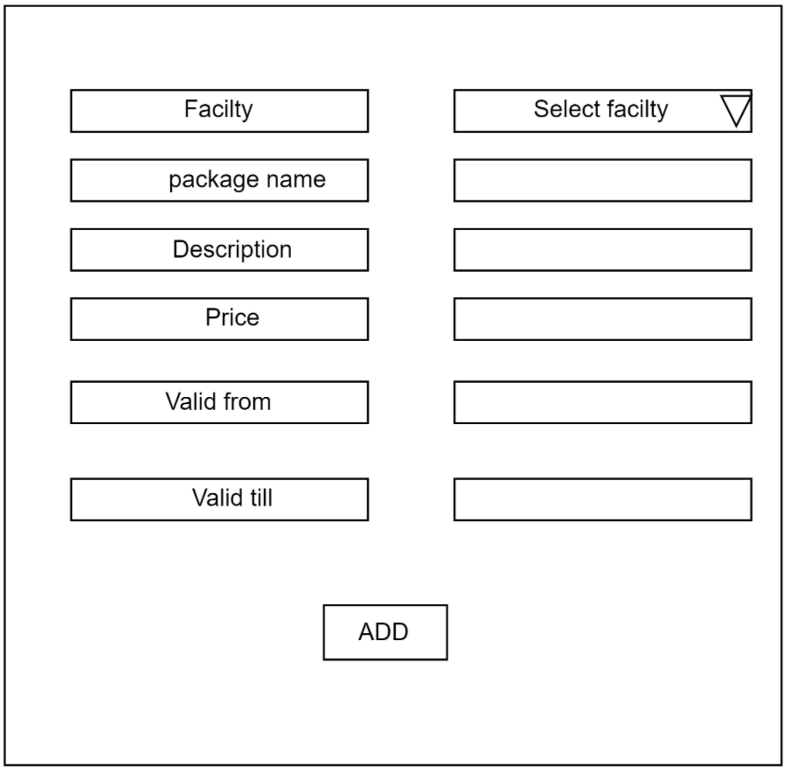
User Registration Form



Add place

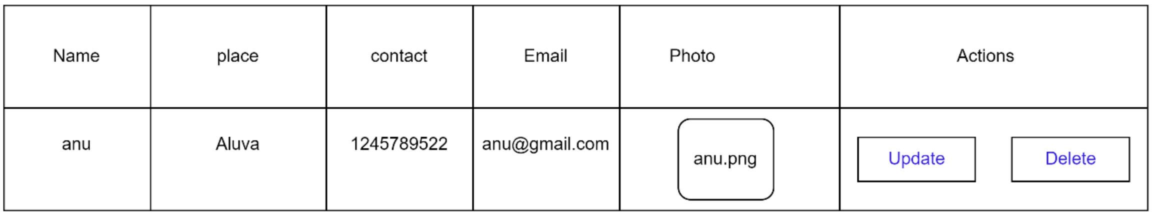


Add package



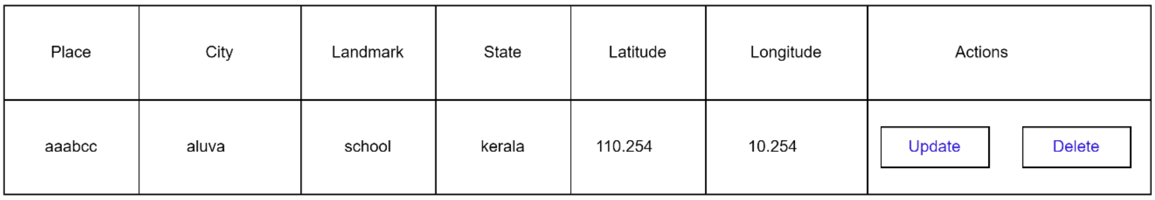
5.3 OUTPUT DESIGN

User details



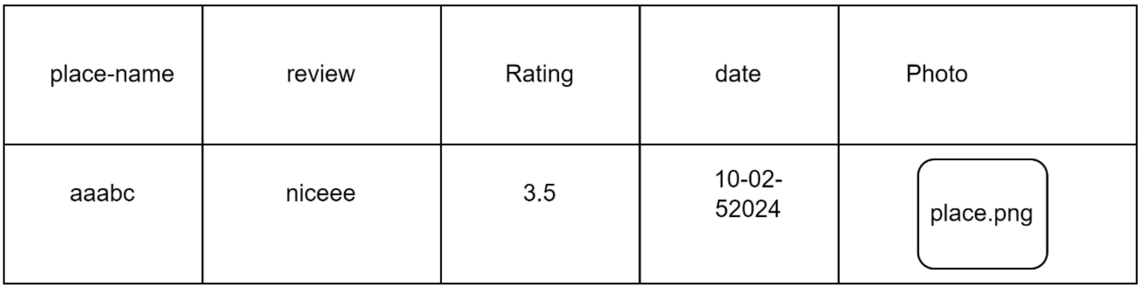
Name	place	contact	Email	Photo	Actions
anu	Aluva	1245789522	anu@gmail.com	anu.png	Update Delete

Place details



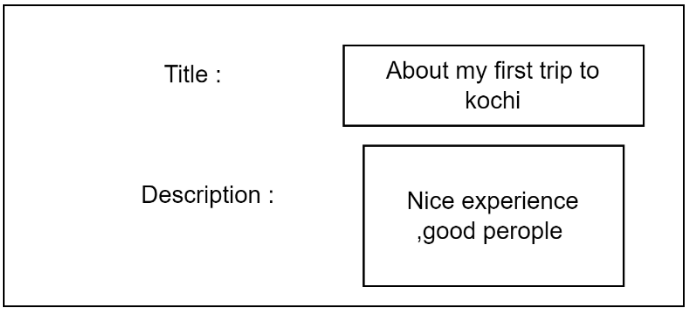
Place	City	Landmark	State	Latitude	Longitude	Actions
aaabcc	aluva	school	kerala	110.254	10.254	Update Delete

View reviews



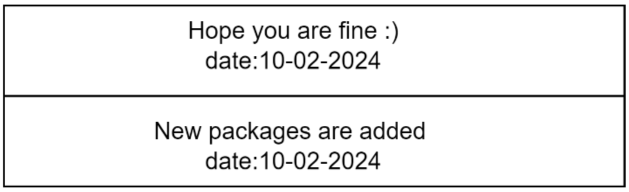
place-name	review	Rating	date	Photo
aaabc	niceee	3.5	10-02-52024	place.png

View travelogues



Title :	About my first trip to kochi
Description :	Nice experience ,good perople

View notification



Hope you are fine :)
date:10-02-2024

New packages are added
date:10-02-2024

5.4 NUMBER OF MODULES AND THEIR DESCRIPTION

The three different modules included in the system are, admin, users and public. Different modules have their own function. The admin module controls the overall system.

ADMIN:

- User details registration
- Information about important places
- Sent notification (Relating new tour packages, new places etc.)
- View feedback
- Add hospitals
- Add hotels (With packages)

USERS:

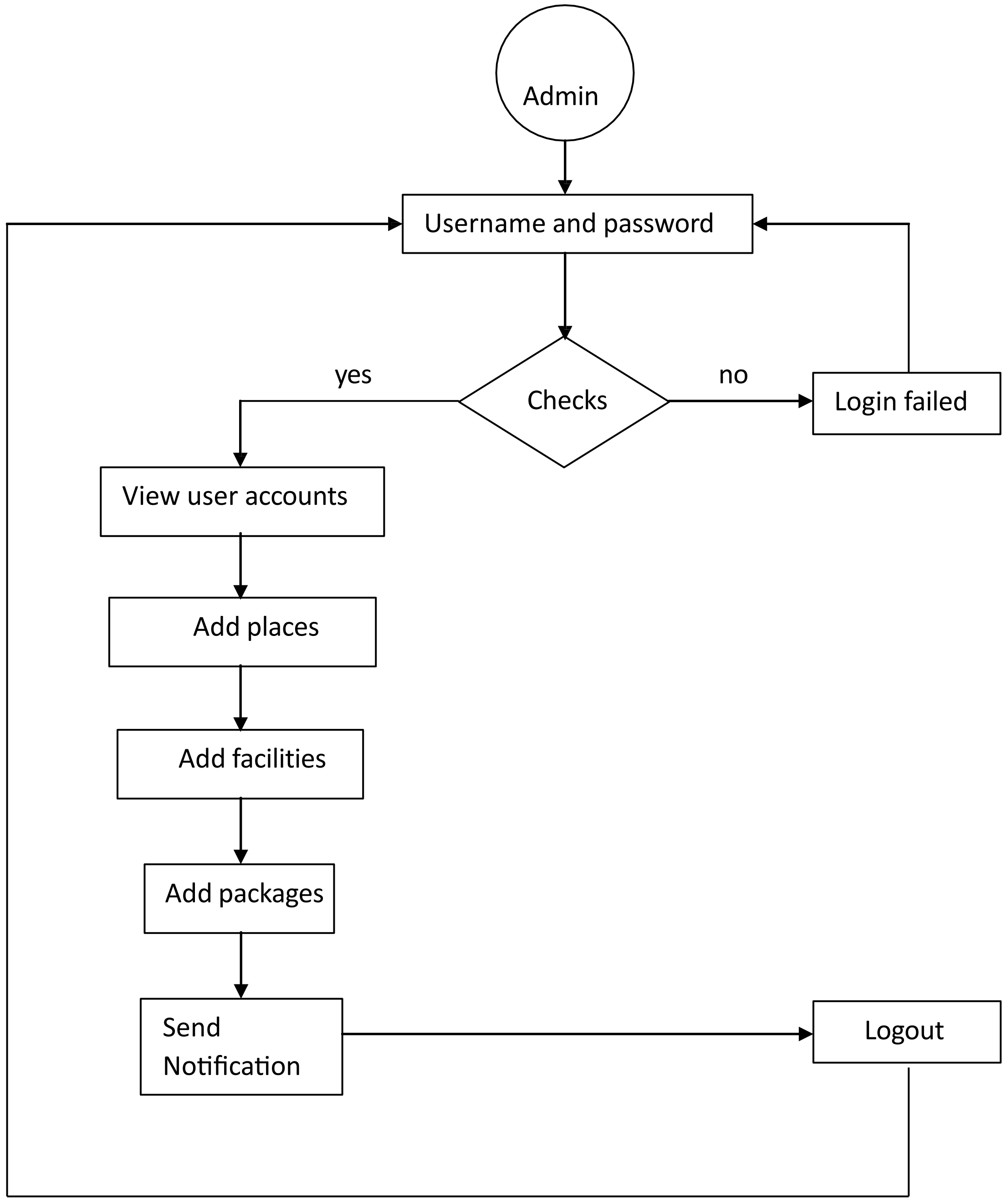
- View map (current location update)
- Add and view experience videos/ audio/description
- View nearby hotels (including different packages), hospitals etc.
- View nearby tourist attractions
- View and upload travelogue
- Suggest and view travelogue (Link to you tube)
- Sent feedback
- View notifications

PUBLIC:

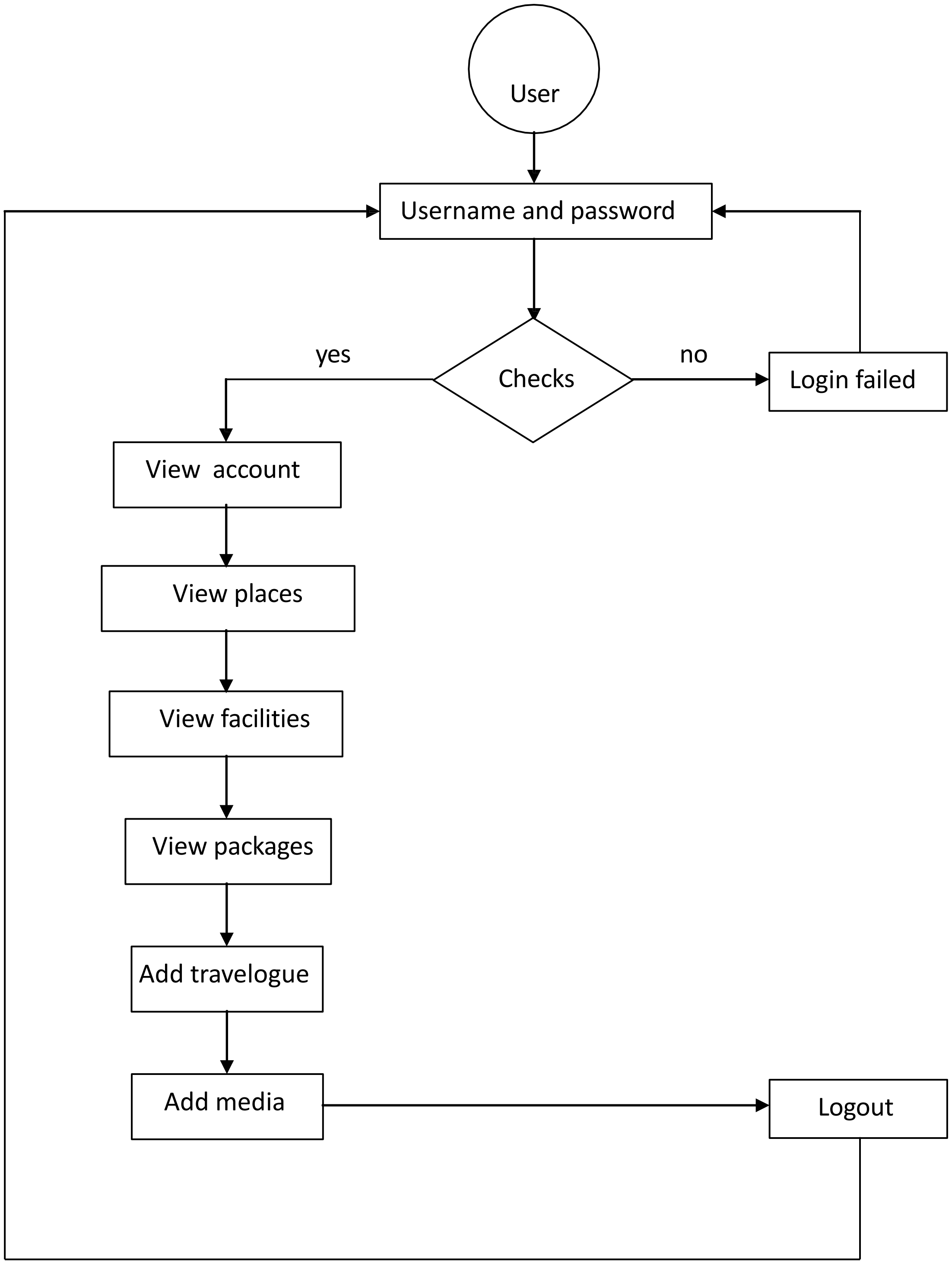
- View important places
- Track places
- View count of users
- View feedback

5.5 FUNCTIIONAL DIAGRAM

Admin



User


User
Username and password
Checks
yes
no
Login failed
View account
View places
View facilities
View packages
Add travelogue
Add media
Logout

5.6 DATABASE DESIGN

Table Name: login

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
login_id	integer	primary key	ID of login
Username	Varchar [50]	not null	Username for logging in
Password	Varchar [50]	not null	Password for logging in
usertype	Varchar [50]	not null	Admin, user

Table Name: user

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
user_id	integer	primary key	ID of the user
Name	Varchar [50]	not null	Name of the user
Place	Integer [11]	not null	Place of the user
Contact	Varchar [100]	not null	Contact of the user
Email	Varchar [12]	not null	Email of user
Photo	Varchar [50]	not null	Photo of user
login_id	Varchar [10]	Foreign Key	Phone number of user
Lati	Varchar [50]	not null	Latitude of user
Longi	Varchar [50]	Not null	Longitude of user

Table Name: place

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
Place_id	integer	primary key	Id of the place
Place_name	Varchar [50]	Not null	Name of the place
City	Varchar [50]	not null	City of the place
Landmark	Varchar [50]	not null	Landmark of the place
State	Varchar [50]	not null	State of the place
Lati	Varchar [50]	Not null	Latitude of the place
Longi	Varchar [50]	Not null	Longitude of the place
Status	Varchar [50]	Not null	Status of the place

Table Name: facility

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
Facility_id	integer	primary key	ID of the facility
Type	Varchar [50]	Not null	Type of the facility
f_name	Varchar [50]	not null	Name of the facility
f_place	Varchar [50]	not null	Place of the facility
City	Varchar [50]	not null	City of the facility
State	Varchar [50]	Not null	State of the facility
Phone	Varchar [50]	Not null	Phone of the facility

Email	Varchar [50]	Not null	Email of the facility
Photo	Varchar [50]	Not null	Photo of the facility
Lati	Varchar [50]	Not null	Latitude of the facility
Longi	Varchar [50]	Not null	Longitude of the facility

Table Name: package

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
Package_id	integer	primary key	ID for package
Package_name	Varchar [50]	Not null	Name of the package
Description	Varchar [50]	not null	Description of the package
Valid from	Varchar [50]	Not null	Starting Validity
Valid till	Varchar [50]	Not null	Ending validity
Price	Varchar [50]	Not null	Price of the package
facility_id	Varchar [50]	Foreign key	Id of the facility

Table Name: notification

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
noti_id	integer	primary key	ID of notification
Notification	Varchar [50]	not null	Notification
Date	Date	not null	Date of the notification

Table Name: review

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
review_id	integer	primary key	ID for Review
Review	Varchar [50]	Not null	Review of the user
Rating	Varchar [50]	Not null	Rating of the user
Date	Date [10]	Not null	Date
Photo1	Varchar [10]	Not null	Photo of the place
place_id	integer	Foreign key	Id of the place

Table Name: Travelogue

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
travelogue_id	integer	primary key	ID of travelogue
Travel_title	Varchar [50]	not null	Title of the travelogue
description	Varchar [50]	not null	Description about travelogue
User_id	integer	Foreign key	ID of the user

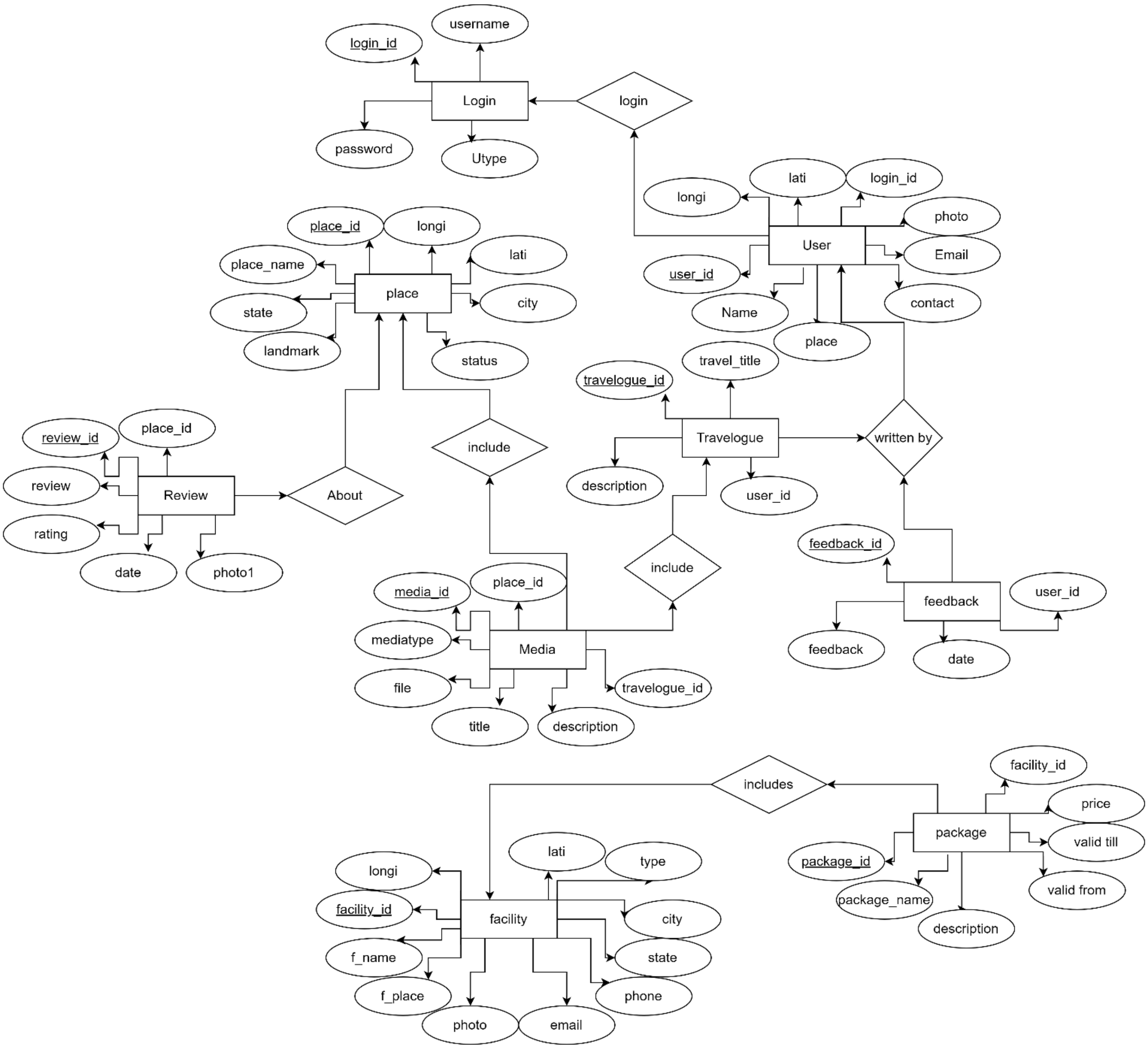
Table Name: media

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
media_id	integer	primary key	ID of the media
mediatype	Varchar [50]	Not null	Type of the media
file	Varchar [50]	not null	Path of the file
description	Varchar [50]	not null	Description about media
title	Varchar [50]	not null	Title for the media
Places_id	integer	Foreign key	ID of the place
Travelogue_id	integer	Foreign key	ID of the travelogue

Table Name: feedback

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
feedback_id	integer	primary key	ID of feedback
feedback	Varchar [50]	not null	Feedback given by user
date	Varchar [50]	not null	Feedback Uploaded date
User_id	integer	Foreign key	ID of the user

5.7 E R DIAGRAM

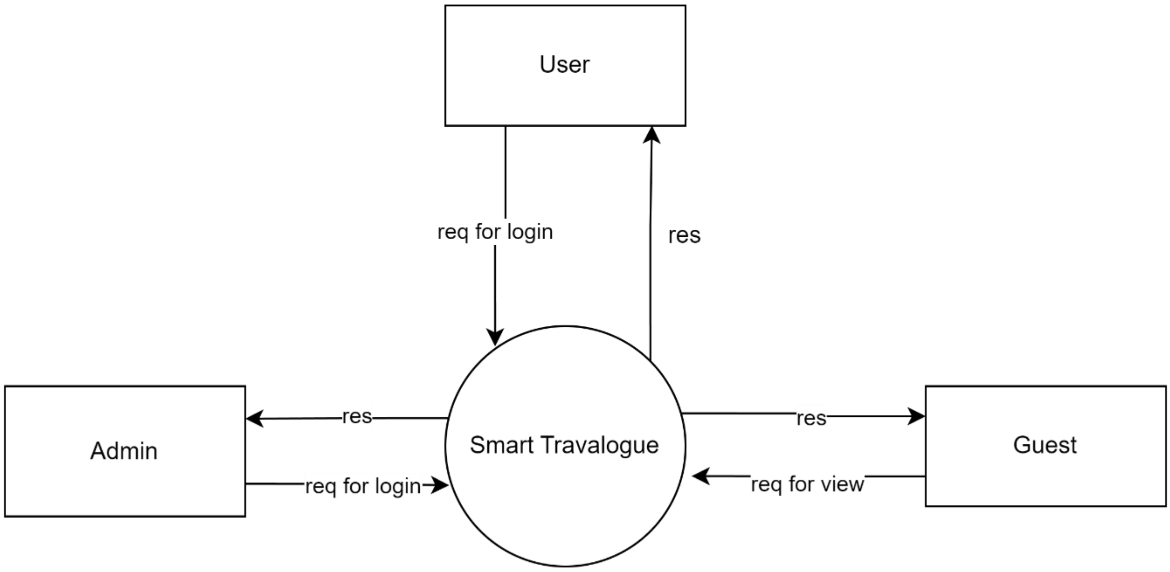


5.8 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) also called Data Floe Diagram is commonly used during problem analysis. They are quite general and are not limited to problem analysis for software requirement specification. DFD is very useful in understanding a system and is effectively used during analysis.

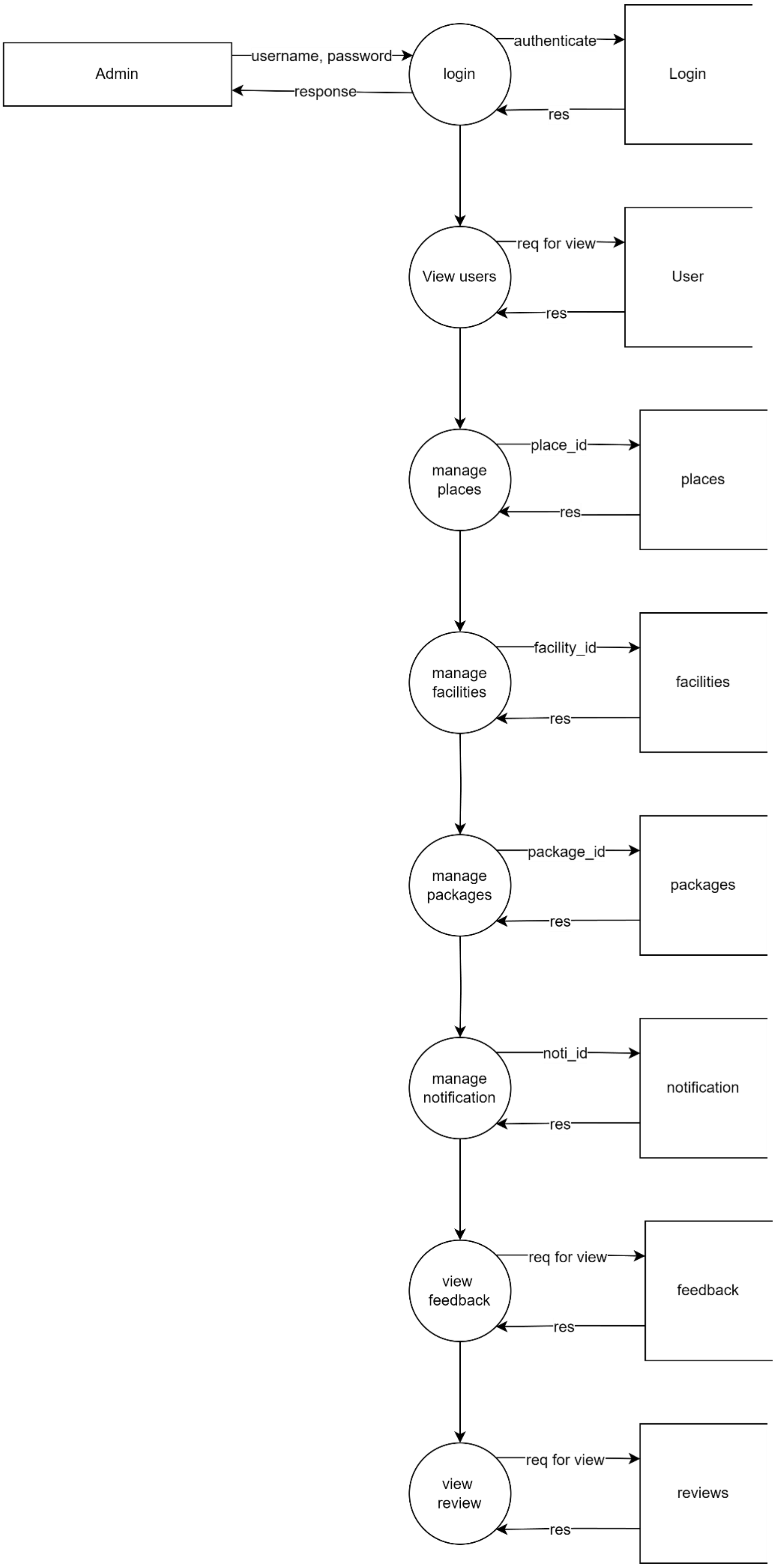
A DFD shows the flow of data through a system. It views the system as a function that transforms the input into desired output. Any complex system will not perform this transformation in a single step and data will typically undergo a series of transformations before it becomes the output. The DFD aims to capture this transformation that takes place in the system. The agent that performs this transformation of data from one stage to another is called a process or a bubble. So, a DFD shows the movement of data through different transformations in the system.

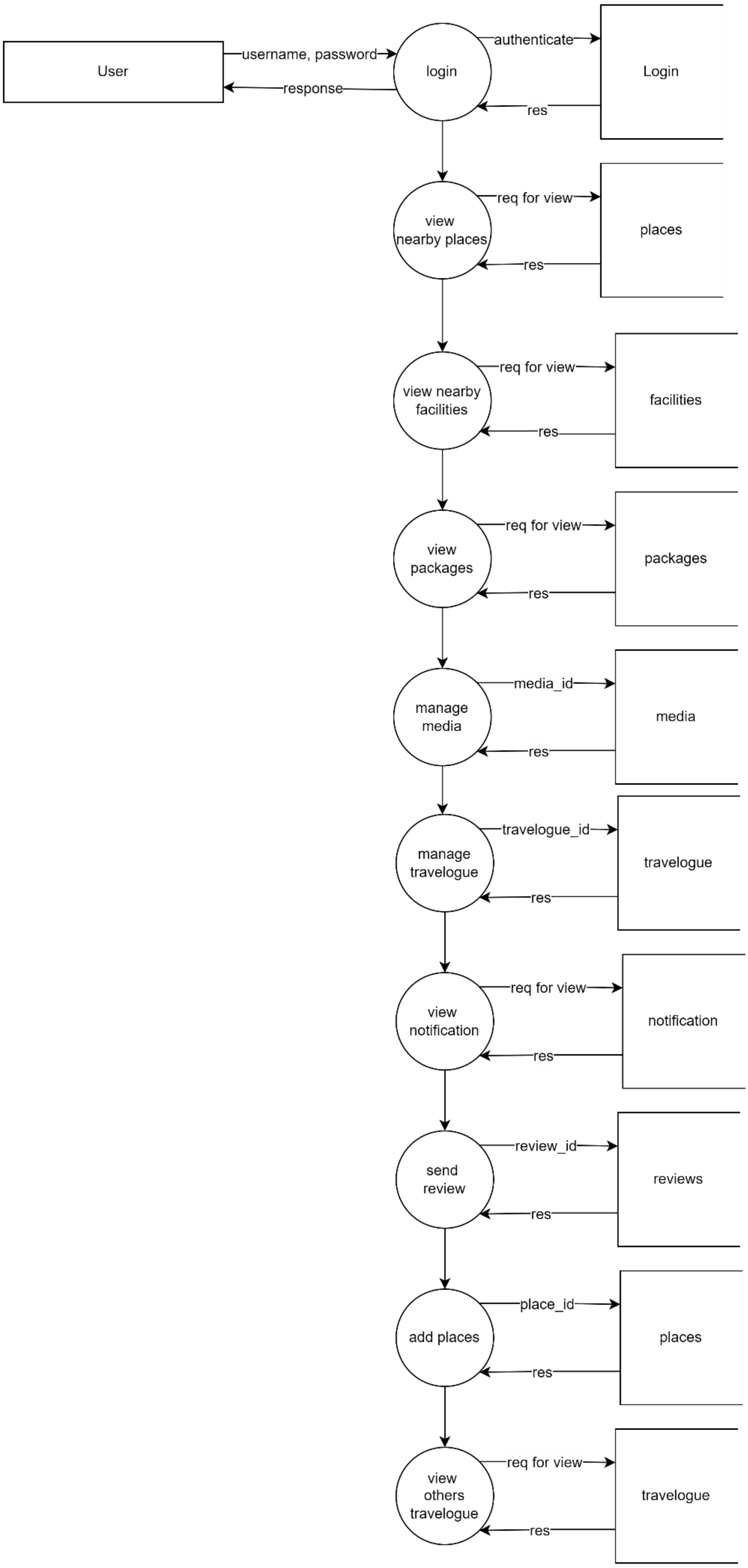
- **Level 0 - Data Flow Diagram**



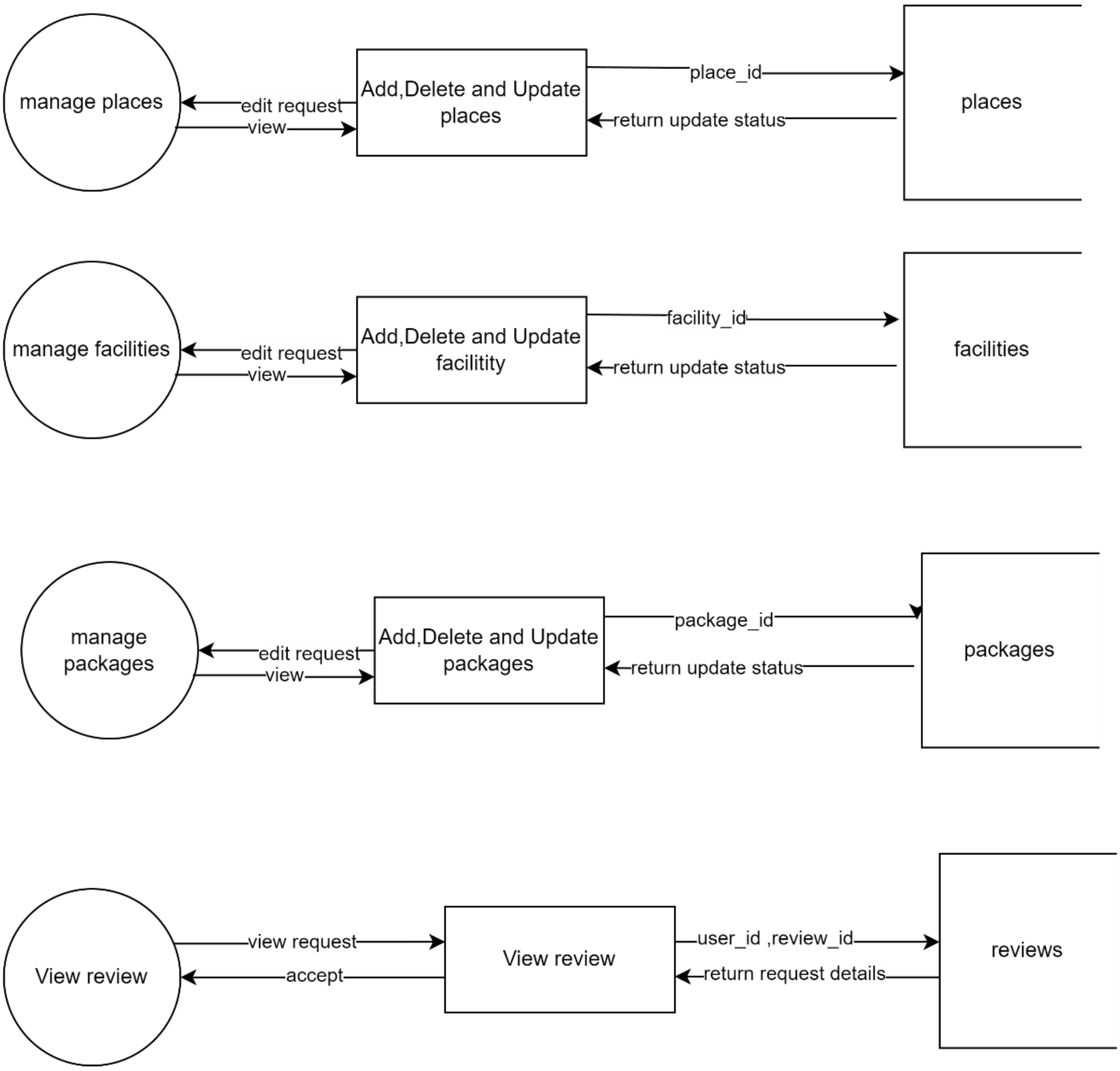
- **Level 1 – Data Flow Diagram**

Admin


Admin
username, password
response
login
authenticate
res
Login
View users
req for view
res
User
manage places
place_id
res
places
manage facilities
facility_id
res
facilities
manage packages
package_id
res
packages
manage notification
noti_id
res
notification
view feedback
req for view
res
feedback
view review
req for view
res
reviews


User
username, password
response
login
authenticate
res
Login
view nearby places
req for view
res
places
view nearby facilities
req for view
res
facilities
view packages
req for view
res
packages
manage media
media_id
res
media
manage travelogue
travelogue_id
res
travelogue
view notification
req for view
res
notification
send review
review_id
res
reviews
add places
place_id
res
places
view others travelogue
req for view
res
travelogue

Level 2 - Data Flow Diagram



6. SYSTEM DEVELOPMENT

6.1 PROCESS DESCRIPTION

Proposed system is fully computerized one. With the help of proposed system, we can overcome all the limitations of the existing system. The information about places can be retrieved very quickly with a button click when compared to earlier system. It provides a fast and efficient service to the users. Proposed system is low time consuming, gives accurate results, reliability can improve with the help of security.

6.2 PSEUDO CODE

- **admin_manage_places.html**

```
{%include 'adminheader.html'%}
<script type="text/javascript"

       src="http://maps.googleapis.com/maps/api/js?key=AIzaSyD3MPnSnyWw
       NmpnVEFkaddVvy_GWtxSejs&sensor=false"></script>
<script type="text/javascript">
  // window.onload = function () {
  function myFunction() {
    var mapOptions = {
      center: new google.maps.LatLng(9.9763482, 76.286272),
      zoom: 14,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var infoWindow = new google.maps.InfoWindow();
    var latlngbounds = new google.maps.LatLngBounds();
    var map = new google.maps.Map(document.getElementById("dvMap"),
       mapOptions);
    google.maps.event.addListener(map, 'click', function (e) {
      alert("Latitude: " + e.latLng.lat() + "\r\nLongitude: " + e.latLng.lng());
      document.getElementById('slati').value = e.latLng.lat();
      document.getElementById('elongi').value = e.latLng.lng();
    });
  }

</script>
```

```
{%if obj%}
<center> <h1>MANAGE PLACES !</h1></center>
<div style="display: flex; justify-content: center; align-items: center; height: 100vh;">

  <div style="width: 600px; padding: 20px; background-color: #f0f0f026; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);">
    <form action="" method="post">
      <table style="width: 100%;">
        <tr>
          <th style="text-align: left;">Place</th>

          <td><br><input type="text" class="form-control" value="{{obj.place_name}}" name="plc_name" id=""></td>
        </tr>
        <tr>
          <th style="text-align: left;">City</th>
          <td><br><input type="text" class="form-control" value="{{obj.city}}" name="city" id=""></td>
        </tr>
        <tr>
          <th style="text-align: left;">Landmark</th>
          <td><br><input type="text" class="form-control" value="{{obj.landmark}}" name="lmark" id=""></td>
        </tr>
        <tr>
          <th style="text-align: left;">State</th>
          <td><br><input type="text" class="form-control" value="{{obj.state}}" name="state" id=""></td>
        </tr>
        <tr>
          <th style="font-size: 25px; text-align: left;">LATITUDE</th>
          <td style="color: black;"><br><input type="text" class="form-control" value="{{obj.lati}}" name="lati" id="slati"></td>
          <td><a onclick="myFunction()" style="cursor: pointer;">View Map</a></td>
        </tr>
        <tr>
          <th style="font-size: 25px; text-align: left;">LONGITUDE</th>
          <td style="color: black;"><br><input type="text" class="form-control" value="{{obj.longi}}"  name="longi" id="elongi" required></td>
        </tr>
        <tr style="margin-top: 10px;">
          <td colspan="2" style="text-align: center;"><br><input class="btn btn-outline-success" value="submit" type="submit" class="form-control" name="update" ></td>
        </tr>
        <tr>
          <td colspan="2">
            <div id="dvMap" style="width: 100%; height: 300px;"></div>
          </td>
        </tr>
      </table>
    </form>
  </div>
</div>

{%else%}


<center> <h1>MANAGE PLACES !</h1></center>
<div style="display: flex; justify-content: center; align-items: center; height: 100vh;">

  <div style="width: 600px; padding: 20px; background-color: #f0f0f026; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);">
    <form action="" method="post">
      <table style="width: 100%;">
        <tr>
          <th style="text-align: left;">Place</th>

          <td><br><input type="text" class="form-control" name="plc_name" id=""></td>
        </tr>
        <tr>
          <th style="text-align: left;">City</th>
          <td><br><input type="text" class="form-control" name="city" id=""></td>
        </tr>
```

```
            <tr>
              <th style="text-align: left;">Landmark</th>
              <td><br><input type="text" class="form-control" name="lmark"
          id=""></td>
            </tr>
            <tr>
              <th style="text-align: left;">State</th>
              <td><br><input type="text" class="form-control" name="state"
          id=""></td>
            </tr>
            <tr>
              <th style="font-size: 25px; text-align: left;">LATITUDE</th>
              <td style="color: black;"><br><input type="text" class="form-control"
          name="lati" id="slati"></td>
              <td><a onclick="myFunction()" style="cursor: pointer;">View
          Map</a></td>
            </tr>
            <tr>
              <th style="font-size: 25px; text-align: left;">LONGITUDE</th>
              <td style="color: black;"><br><input type="text" class="form-control"
          name="longi" id="elongi" required></td>
            </tr>
            <tr style="margin-top: 10px;">
              <td colspan="2" style="text-align: center;"><br><input class="btn btn-
          outline-success" value="submit" type="submit" class="form-control"
          name="sub"></td>
            </tr>
            <tr>
              <td colspan="2">
                <div id="dvMap" style="width: 100%; height: 300px;"></div>
              </td>
            </tr>
          </table>
        </form>
      </div>
    </div>
    <center>
      <h1>VIEW PLACES !</h1>
      <table class="table table-stripped" align="center" border="1" style="color:
          white; width:50%;">
        <tr  style="background-color: white; color:black">
           <th>Place</th>
           <th>City</th>
           <th>Landmark</th>
           <th>State</th>
           <th>Lati</th>
           <th>Longi</th>
           <th></th>
 </tr>
        {%for i in ss%}
        <tr>
           <td>{{i.place_name}}</td>
           <td>{{i.city}}</td>
           <td>{{i.landmark}}</td>
           <td>{{i.state}}</td>
           <td>{{i.lati}}</td>
           <td>{{i.longi}}</td>
           <td><a class="btn btn-info"
            href="/admin_update_places/{{i.place_id}}">Update</a>
              <a class="btn btn-danger"
            href="/admin_delete_delete/{{i.place_id}}">Delete</a>
           </td>
        </tr>
        {%endfor%}
   </table>
</center>
{%endif%}
{%include 'footer.html'%}
```

- **android/manage_travelogue.java**

```
package com.example.smart_travelogue;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.View;
import android.widget.Button;
```

```
    import android.widget.EditText;
    import android.widget.TextView;
    import android.widget.Toast;
    import org.json.JSONObject;
    public class Manage_travelogue extends AppCompatActivity  implements
              JsonResponse{
      EditText e1,e2;
      Button btn;
     String title,description;
      TextView t1;
      SharedPreferences sh;

      @Override
      protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manage_travelogue);
  sh = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        e1 = findViewById(R.id.complaint_tt);
        e2 = findViewById(R.id.description);
        btn = findViewById(R.id.button1);
btn.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {


            title = e1.getText().toString();
            description = e2.getText().toString();
           JsonReq JR = new JsonReq();
            JR.json_response = (JsonResponse) Manage_travelogue.this;
            String q = "/add_travelogue?login_id=" + sh.getString("login_id","") +
           "&title=" + title + "&desc=" + description;
            q = q.replace(" ", "%20");
            JR.execute(q);
          }
        });
        t1 = findViewById(R.id.Viwtravelogue);
        t1.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
            startActivity(new Intent(getApplicationContext(),View_travelogue.class));
            }
        });
    }
    @Override
    public void response(JSONObject jo) {
        try {
            String status=jo.getString("status");
            if (status.equalsIgnoreCase("success")){
                Toast.makeText(this, "added huui :)", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getApplicationContext(),User_home.class));
            }
        }catch (Exception e){
            Toast.makeText(this, e.toString(), Toast.LENGTH_SHORT).show();
        }
    }
    public void onBackPressed()
    {
        // TODO Auto-generated method stub
        super.onBackPressed();
        Intent b=new Intent(getApplicationContext(),User_home.class);
        startActivity(b);
    }
}
```

7.VALIDATION CHECKS

The validation phase reveals the failure and the buds in the developed system. It will become to known about the practical difficulties the system faces when the operated in the true environment. Validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid email?
- Has the user entered text in a numeric field?

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server. Most often, the purpose of validation is to ensure correct user input. Validation can be defined by many different methods, and deployed in many different ways. Server- side validation is performed by a web server, after input has been sent to the server. Client-side validation is performed by a web browser, before input is sent to a web server.

8.SYSTEM IMPLEMENTATION

8.1 Testing

Testing is an important stage in the software development life cycle. System testing is a critical element of a software quality assurance and represents the ultimate review of specification, design and coding.

Testing is a critical phase in its development process, aimed at ensuring its functionality, reliability, and usability. Various testing techniques such as unit testing, integration testing, system testing, and user acceptance testing are employed to validate different aspects of the app. Unit testing focuses on testing individual components or modules to verify their correctness and functionality. Integration testing ensures that different modules or components interact seamlessly when integrated together. System testing evaluates the app as a whole, testing its overall functionality, performance, and compatibility with different devices and platforms. User acceptance testing involves real users testing the app in a simulated environment to assess its usability, intuitiveness, and overall user experience. Additionally, performance testing and security testing are conducted to evaluate the app's performance under different load conditions and to identify and mitigate potential security vulnerabilities. Through comprehensive testing, the Smart travelogue App can be refined and optimized to meet user expectations and deliver a seamless and reliable fuel delivery experience.

8.2 System Implementation

Implementation is the stage in the project where theoretical design is turned into a working system and is giving confidence on the new system for the users which will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementations, design of methods to achieve the changeover, an evaluation, of changeover methods. Apart from planning major tasks for preparing the implementation are education and training of users. The major complex system being implemented the more evolved will be the system analysis and the design effort required just for implementation. An implementation coordination committee based on policies of individual organization has been appointed.

The implementation process begins with preparing plan for implementation of the system. According to this plan the activities are to be carried out discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. 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 and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if found to working according to the specification.

8.3 Security

Data is the most valuable asset of any organization and is essential to protect it from external access or unauthorized access. It’s a crucial task during the development of any system to ensure the data security. In my project, there are no of users it is must to ensure authorized user can only access particular part of the database. It also restricts CRUD operation based on the user role. To improve the security, each module of code is protected and authorized person can only access defined operation. So, the entire system is provided with tight security

9.FUTURE SCOPE OF THE PROJECT

In its future trajectory, Smart Travelogue holds significant potential for expansion and innovation within the travel technology landscape. One avenue for development lies in integrating emerging technologies such as augmented reality (AR) and virtual reality (VR) to offer immersive travel experiences. AR overlays can provide real-time information about nearby attractions, while VR simulations can offer virtual tours of destinations, enriching the user experience. Additionally, the platform can explore advanced machine learning algorithms to personalize recommendations and travel suggestions, tailoring them to individual user preferences and behaviors. As Smart Travelogue gains momentum, there's an opportunity for global expansion and localization, with the platform adapting to the specific needs and cultural nuances of diverse regions. Strengthening community engagement through enhanced social features and collaboration tools can foster a sense of belonging among users, encouraging interaction and shared experiences. Furthermore, prioritizing accessibility and inclusivity remains paramount, with ongoing improvements to ensure that all users, regardless of ability, can access and utilize the platform effectively. Through strategic partnerships with travel agencies and tourism boards, Smart Travelogue can unlock new opportunities for curated experiences and exclusive offerings. Leveraging data analytics and insights will continue to be instrumental in refining the platform, providing valuable feedback on user behaviour and preferences to drive continuous innovation. Overall, Smart Travelogue is poised to evolve into a comprehensive and indispensable travel companion, enhancing the journeys of travelers worldwide with personalized, immersive, and inclusive experiences.

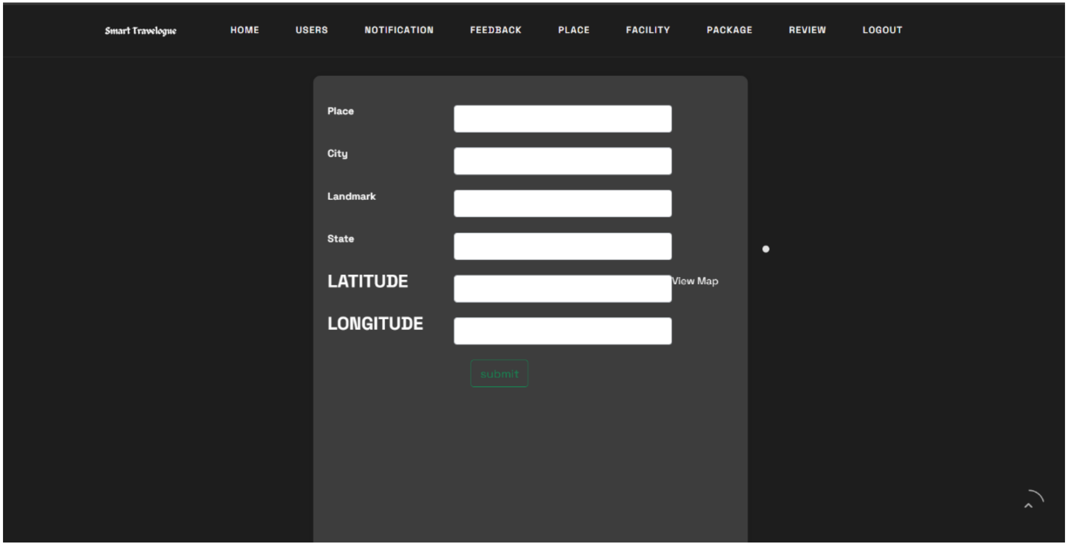
10. CONCLUSION

In conclusion, Smart Travelogue represents a dynamic and innovative solution poised to revolutionize the way travelers navigate and explore unfamiliar territories. Through its suite of features, including real-time location tracking, navigation assistance, comprehensive information on amenities and attractions, and user-generated content, Smart Travelogue prioritizes safety, convenience, and personalized experiences for users. As we look towards the future, the potential for further innovation and expansion is vast. By integrating emerging technologies, enhancing personalization, expanding globally, fostering community engagement, and prioritizing accessibility, Smart Travelogue is well-positioned to continue its trajectory as a leading travel companion. With a commitment to continuous improvement and a focus on meeting the diverse needs of travelers worldwide, Smart Travelogue stands poised to redefine the travel experience, enriching journeys and enhancing satisfaction for users across the globe.

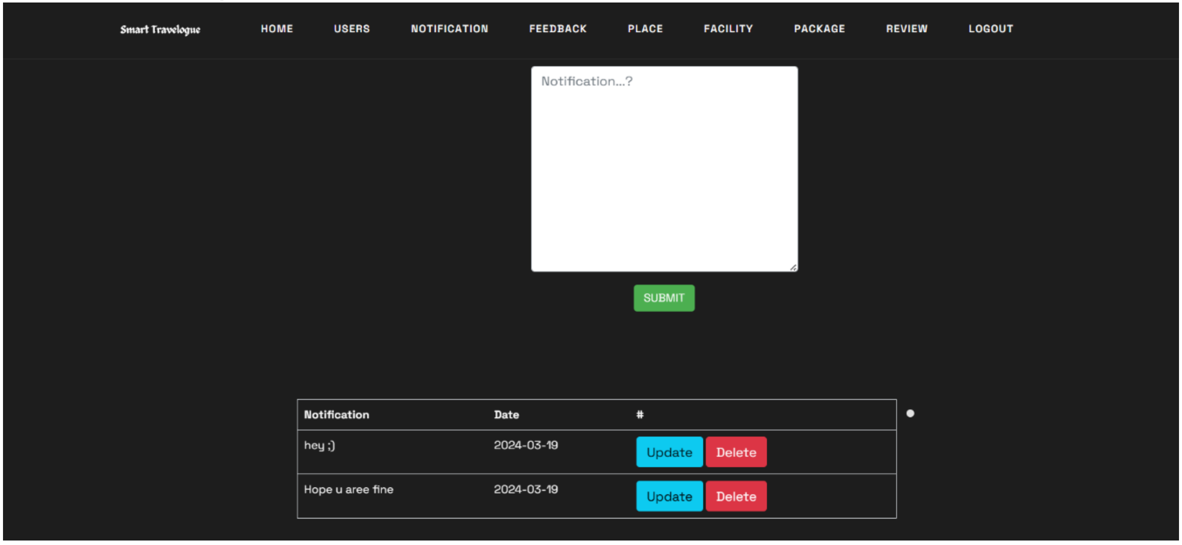
11. APPENDIX

11.1 SAMPLE INPUT

- **Add new place**

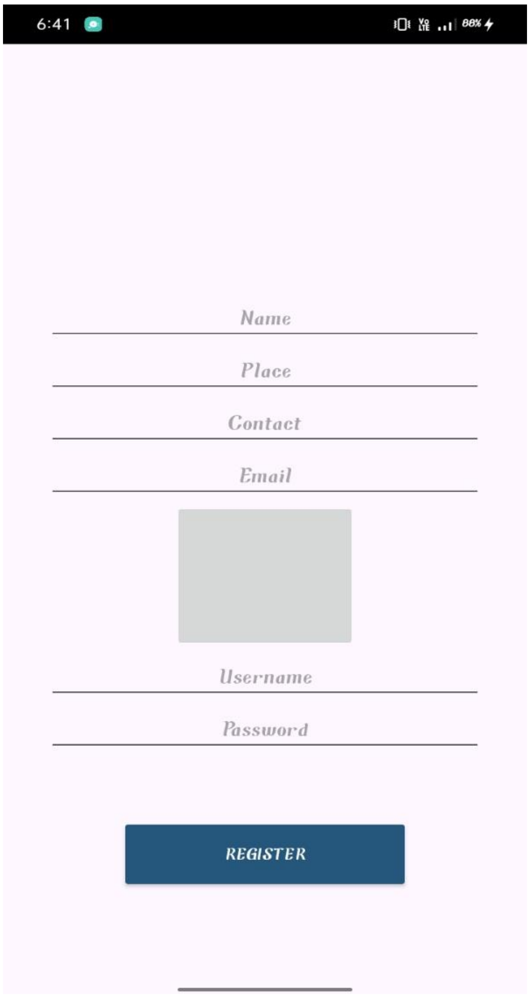


- **Add and manage notification**

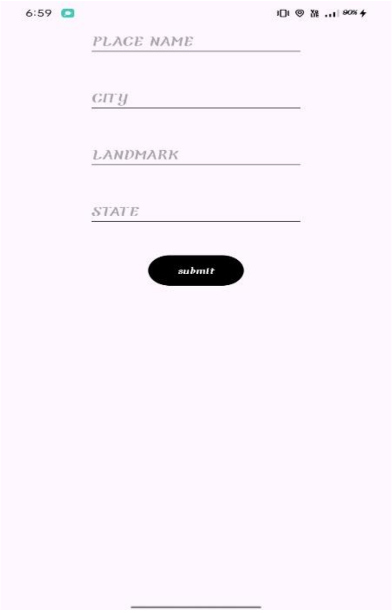


Notification	Date	#
hey ;)	2024-03-19	Update Delete
Hope u aree fine	2024-03-19	Update Delete

- **User registration**

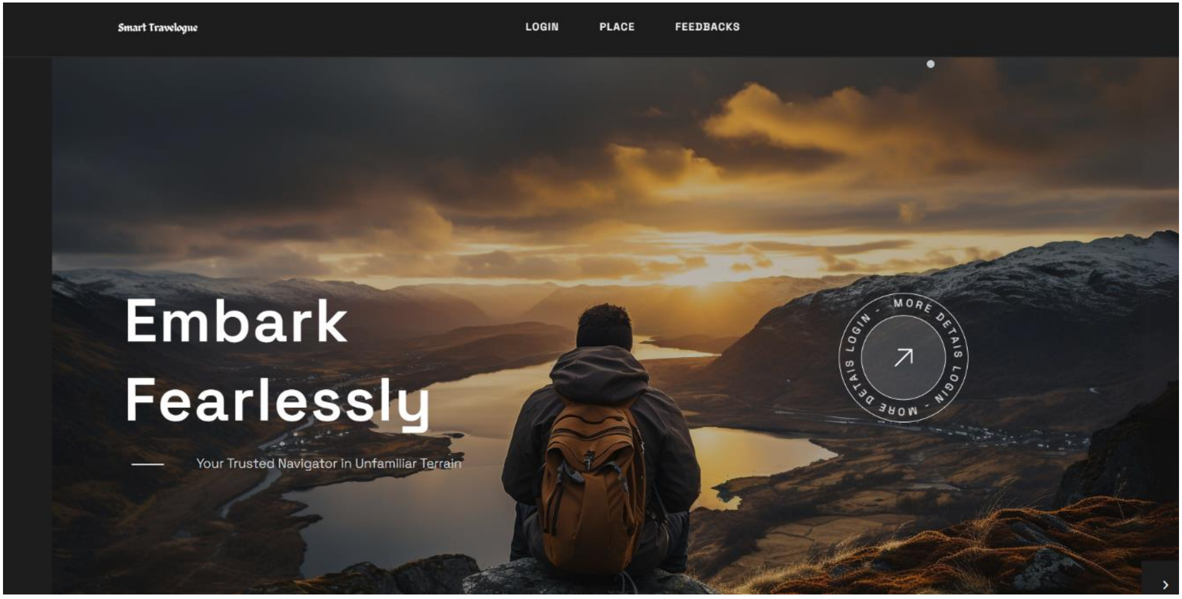


- **Add places**



11.2 SAMPLE OUTPUT

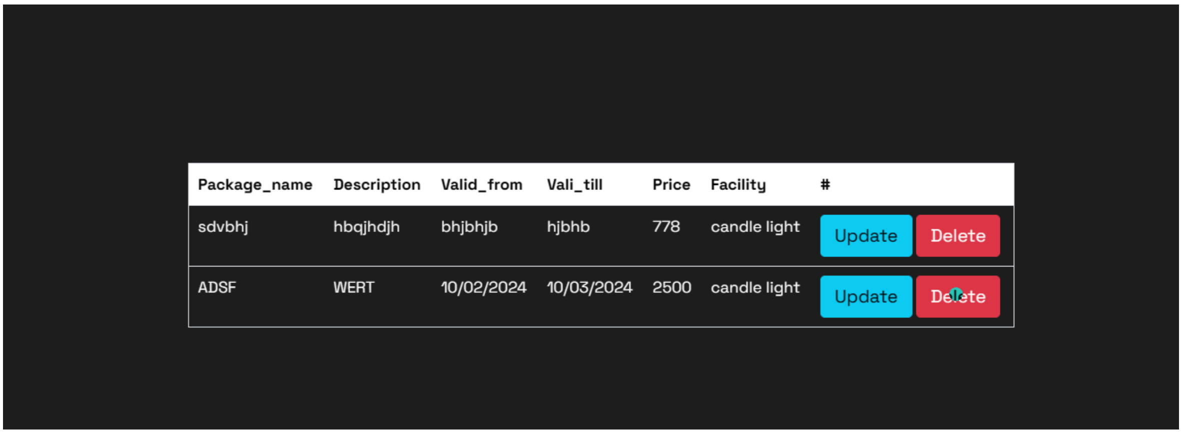
- **Landing page**



Embark Fearlessly

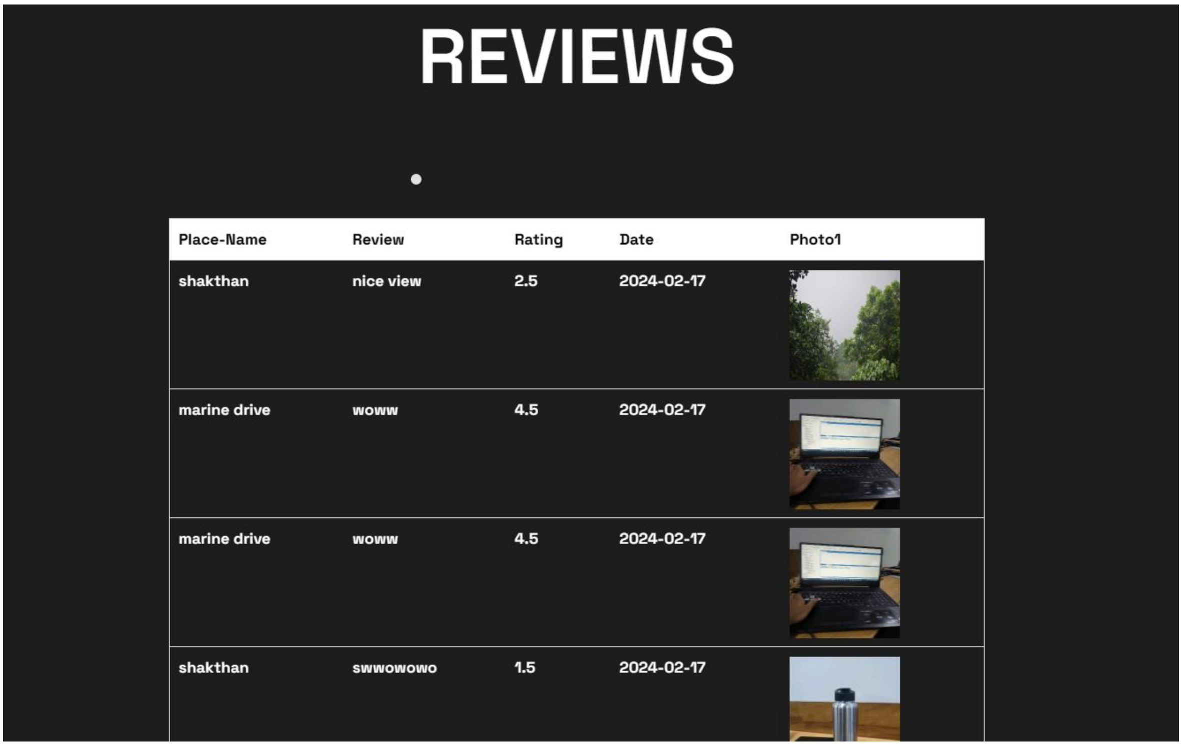
Your Trusted Navigator In Unfamiliar Terrain

- **View packages**



Package_name	Description	Valid_from	Vali_till	Price	Facility	#
sdvbhj	hbqjhdjh	bhjbhjb	hjbhb	778	candle light	Update Delete
ADSF	WERT	10/02/2024	10/03/2024	2500	candle light	Update Delete

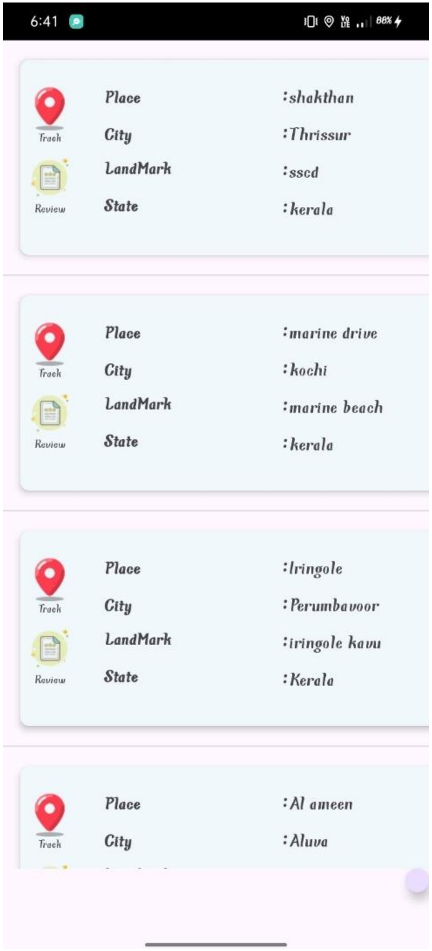
- **Reviews**



REVIEWS

Place-Name	Review	Rating	Date	Photo1
shakthan	nice view	2.5	2024-02-17	
marine drive	woww	4.5	2024-02-17	
marine drive	woww	4.5	2024-02-17	
shakthan	swwowowo	1.5	2024-02-17	

- **View places**



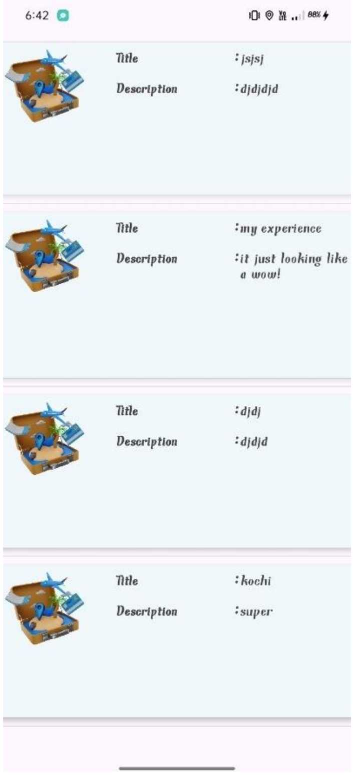
Place	:shakthan
City	:Thrissur
LandMark	:sscd
State	:kerala

Place	:marine drive
City	:kochi
LandMark	:marine beach
State	:kerala

Place	:Iringole
City	:Perumbavoor
LandMark	:iringole kavu
State	:Kerala

Place	:Al ameen
City	:Aluva

- **View travelogue**



6:42

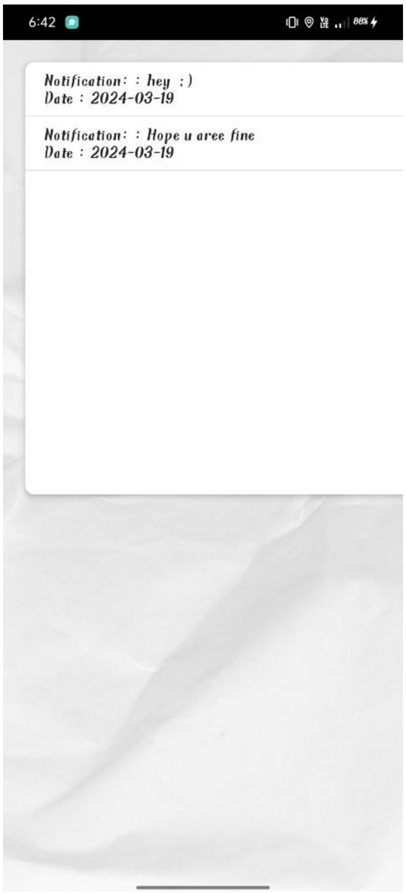
Title :jsjsj
Description :djdjdjd

Title :my experience
Description :it just looking like a wow!

Title :djdj
Description :djdjd

Title :kochi
Description :super

- **View notification**



6:42

Notification: : hey :)
Date : 2024-03-19

Notification: : Hope u aree fine
Date : 2024-03-19

12. BIBLIOGRAPHY

[1] Spurlock, Jake. Bootstrap: Responsive Web Development. "O'Reilly Media, Inc.", 2013.

[2] https://docs.djangoproject.com/en/stable/

[3] Smith, J. (2021). "Mobile App Development: Best Practices and Strategies." Publisher.

[4] Lee, C., Yoo, K., & Lee, H. (2005). "Design and Implementation of a GPS-Based Tourist Guide System." IEEE Transactions on Consumer Electronics, 12(3), 45-58.

[5] https://www.w3schools.com/sql

[6] https://developers.google.com/maps/documentation