

# **SMART TOURIST GUIDING SYSTEM**

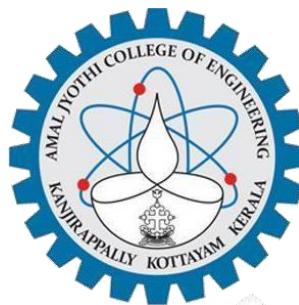
*Project Report Submitted By*

**TOM JOSEPH**

**Reg. No.: AJC20MCA-2081**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 YEAR)  
(MCA)**  
**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING  
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2020-2022**

**DEPARTMENT OF COMPUTER APPLICATIONS**  
**AMAL JYOTHI COLLEGE OF ENGINEERING**  
**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, "**SMART TOURIST GUIDING SYSTEM**" is the bonafide work of **TOM JOSEPH (Reg.No:AJC20MCA-2081)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

**Ms. Nimmy Francis**  
**Internal Guide**

**Ms. Grace Joseph**  
**Coordinator**

**Rev.Fr. Dr. Rubin Thottupurathu Jose**  
**Head of the Department**

**External Examiner**

**(Add CERTIFICATE ON PLAGIARISM CHECK here)**

## **DECLARATION**

I hereby declare that the project report “**SMART TOURIST GUIDING SYSTEM**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

**Date: 21/07/2022**

**TOM JOSEPH**

**KANJIRAPPALLY**

**Reg. No: AJC20MCA-2081**

## **ACKNOWLEDGEMENT**

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev.Fr.Dr. Rubin Thottupurathu Jose** and **Ms. Grace Joseph** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Nimmy Francis** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

TOM JOSEPH

## **ABSTRACT**

As the name specifies Smart Tourist Guiding System is a software developed for guiding tourists for travel destination and booking. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system. The objective of the project is to develop a system that automates the processes and activities of a travel and tourism agency. The purpose is to design a system using which one can perform all operations related to traveling. The tourists and travelers are normally they didn't have a good idea about the destination sites in the case of the visit the place first time. The traveler can find all good destination points especially unexplored places and camping sites through over this project and the did not want to ask for anyone. They get all details of touring sites in our project, they can find the location, routes. The tourist can book local transportation facilities for their journey so did not bother about the local transportation. Also, in this project we provide the help of local guides and travel packages. If any tourists need help for local travel guides for translation, guidance. The tourist can enquire anything related to the content that post in our website. In this project we also focused about travel packages. The tourists can choose their package and customize their package as their need, from our website without any worries. The tourist can directly pay to transportation charge and guide fee through our website.

# CONTENT

Sl. No	Topic	Page No
1	<b>INTRODUCTION</b>	1
1.1	<b>PROJECT OVERVIEW</b>	2
1.2	<b>PROJECT SPECIFICATION</b>	2
2	<b>SYSTEM STUDY</b>	4
2.1	<b>INTRODUCTION</b>	5
2.2	<b>EXISTING SYSTEM</b>	6
2.3	<b>DRAWBACKS OF EXISTING SYSTEM</b>	6
2.4	<b>PROPOSED SYSTEM</b>	6
2.5	<b>ADVANTAGES OF PROPOSED SYSTEM</b>	7
3	<b>REQUIREMENT ANALYSIS</b>	9
3.1	<b>FEASIBILITY STUDY</b>	10
3.1.1	<b>ECONOMICAL FEASIBILITY</b>	10
3.1.2	<b>TECHNICAL FEASIBILITY</b>	11
3.1.3	<b>BEHAVIORAL FEASIBILITY</b>	11
3.2	<b>SYSTEM SPECIFICATION</b>	12
3.2.1	<b>HARDWARE SPECIFICATION</b>	12
3.2.2	<b>SOFTWARE SPECIFICATION</b>	12
3.3	<b>SOFTWARE DESCRIPTION</b>	12
3.3.1	<b>PHP</b>	12
3.3.2	<b>MYSQL</b>	13
3.3.3	<b>ANDROID</b>	14
4	<b>SYSTEM DESIGN</b>	16
4.1	<b>INTRODUCTION</b>	17
4.2	<b>UML DIAGRAM</b>	17
4.2.1	<b>USE CASE DIAGRAM</b>	18
4.2.2	<b>SEQUENCE DIAGRAM</b>	21
4.2.3	<b>STATE CHART DIAGRAM</b>	25
4.2.4	<b>ACTIVITY DIAGRAM</b>	28
4.2.5	<b>CLASS DIAGRAM</b>	29
4.2.6	<b>OBJECT DIAGRAM</b>	30
4.2.7	<b>COMPONENT DIAGRAM</b>	30

<b>4.2.8</b>	<b>DEPLOYMENT DIAGRAM</b>	<b>32</b>
<b>4.3</b>	<b>USER INTERFACE DESIGN</b>	<b>33</b>
<b>4.4</b>	<b>DATA BASE DESIGN</b>	<b>39</b>
<b>5</b>	<b>SYSTEM TESTING</b>	<b>50</b>
<b>5.1</b>	<b>INTRODUCTION</b>	<b>51</b>
<b>5.2</b>	<b>TEST PLAN</b>	<b>52</b>
<b>5.2.1</b>	<b>UNIT TESTING</b>	<b>52</b>
<b>5.2.2</b>	<b>INTEGRATION TESTING</b>	<b>53</b>
<b>5.2.3</b>	<b>VALIDATION TESTING</b>	<b>53</b>
<b>5.2.4</b>	<b>USER ACCEPTANCE TESTING</b>	<b>54</b>
<b>5.2.5</b>	<b>AUTOMATION TESTING</b>	<b>54</b>
<b>5.2.6</b>	<b>SELENIUM TESTING</b>	<b>55</b>
<b>6</b>	<b>IMPLEMENTATION</b>	<b>60</b>
<b>6.1</b>	<b>INTRODUCTION</b>	<b>61</b>
<b>6.2</b>	<b>IMPLEMENTATION PROCEDURE</b>	<b>61</b>
<b>6.2.1</b>	<b>USER TRAINING</b>	<b>62</b>
<b>6.2.2</b>	<b>TRAINING ON APPLICATION SOFTWARE</b>	<b>62</b>
<b>6.2.3</b>	<b>SYSTEM MAINTENANCE</b>	<b>62</b>
<b>7</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>64</b>
<b>7.1</b>	<b>CONCLUSION</b>	<b>65</b>
<b>7.2</b>	<b>FUTURE SCOPE</b>	<b>65</b>
<b>8</b>	<b>BIBLIOGRAPHY</b>	<b>66</b>
<b>9</b>	<b>APPENDIX</b>	<b>68</b>
<b>9.1</b>	<b>SAMPLE CODE</b>	<b>69</b>
<b>9.2</b>	<b>SCREEN SHOTS</b>	<b>84</b>

## **List of Abbreviation**

- IDE - Integrated Development Environment
- HTML - Hyper Text Markup Language.
- CSS - Cascading Style Sheet
- SQL - Structured Query Language
- UML - Unified Modeling Language

## **CHAPTER 1**

### **INTRODUCTION**

## 1.1 PROJECT OVERVIEW

**“SMART TOURIST GUIDING SYSTEM”** is a software developed for guiding tourists for travel destination and booking. In order to create a computerised system that is compatible with the present system while also being more user-friendly and GUI focused, it is necessary to identify the shortcomings of the existing system. The system's shortcomings can be resolved by increasing the system's effectiveness. The project's objective is to develop a system that makes running a travel and tourism company more efficient. A system that enables one to perform all travel-related duties is what is intended. Tourists and travelers are normally they didn't have a good idea about the destination sites in the case of the visit the place first time.

The traveler can find all good destination points ,especially unexplored places and camping sites through over this project and they did not want to ask for anyone. They get all details of touring sites in our project, they can find the location, routes. The tourist can book local transportation facilities for their journey so did not bother about local transportation. In this system we provide various types of Travel Packages for tourists. The Travel packages provided by the third party Tourist Agencies. These packages are carefully verified by Admin.So, tourists can book travel package and explore the journey. Our focus of this system is to help tourists. So, we also provide the service of local Guides. If any tourists need help for local travel guides for translation, guidance easily they can book guides from our site. The tourist can enquire anything related to the contentthat post in our website.

## 1.2 PROJECT SPECIFICATION

The suggested system is a website where users may discover all the information about their destination. Also, that the tourist can book travel packages and local guide for their preference through online. We will also provide users to give their feedbacks about the cab drivers and guides. The tourists can also give ratings for travel guide.

The system includes 4 modules. They are:

### 1. The Admin Module

This system requires an administrator login. He is in charge of the entire system. The administrator oversees all data and has access privileges to see, change, and add

---

information about locations, trips, routes, reservations, and other topics. Admin has access to all users who have registered and can control all of his data. The administrator will make and maintain the packages. The administrator is in charge of managing reservations. A traveler's booking may be confirmed or cancelled by the admin.

## **2. Travelers Module**

Traveler can register and using a legitimate username and password, they can log in. The Traveler can update own profile. The tourist can book the travel package (book and cancel booking. The tourist can sent their request for book the travel guides and after the request approval by certain guides and then only the payment option enables for the tourist. Feedback-the tourist can send feedback to admin.

## **3. Local travel guide**

Local guide can register and using a legitimate username and password, they can log in. The Local guide can update own profile. The request from the tourist can see the guide. The guide can add new exploration places and photographs. Guide can report about tourists.

## **4. Tour Packages**

The tourists can book Travel packages from our site. also, user can cancel the package. The package cost can be remitted through online payment.

## **5. Driver Module**

The driver can register, and using a legitimate username and password, they can log in. The driver can enter the details of his cab. The driver can view the requests from the tourists, and he can accept or decline the request. After the approval of user request then only enable the payment option for tourists. The cab charge calculated based on the type of cab, the number of seats in cab and total kilometers travelled. The driver can accept cab charge from tourist.

## **CHAPTER 2**

### **SYSTEM STUDY**

## 2.1 INTRODUCTION

Steps in the system analysis process include problem-solving, data collecting and analysis, and making suggestions for system changes. Throughout this problem-solving process, the system users and system developers must interact often. Any procedure for developing a system should start with a system analysis or study. We carefully review and evaluate the system. As an interrogator, the system analyst examines the internal workings of the existing system. The system's input is recognised, and the system as a whole is viewed. The various processes can be connected to the organisational outputs. The goals of system analysis include comprehending the issue, identifying the pertinent and important aspects, analysing and synthesising diverse components, and ultimately choosing the best or at least an acceptable course of action.

The procedure needs to be thoroughly investigated utilising a range of approaches, such as surveys and interviews. The data acquired by various sources needs to be thoroughly evaluated in order to draw a conclusion. The conclusion is knowing how the system functions. This system is known as the current one. Now, problem areas have been identified after a detailed examination of the current system. Now the designer takes on the role of a problem-solver and attempts to fix the problems the company is experiencing. The solutions are replaced with proposals. The best option is then selected after an analytical comparison of the proposal and the existing system. The user is given the choice to accept or reject the suggestion when it is presented to them. In response to user requests, the proposal is assessed and adjusted as needed. This loop ends once the user is satisfied with the proposal.

The process of collecting and analysing data in order to use it for future system inquiries is known as preliminary research. Initial research is an exercise in problem-solving that necessitates proper collaboration between system users and developers. It conducts a number of feasibility studies. These studies provide an approximation of the system activities, which can be utilised to select the methodologies to be used for efficient system research and analysis.

## 2.2 EXISTING SYSTEM

In the current system, a customer must contact numerous organisations in order to find out information about locations and make travel arrangements. It is also difficult to find a reliable local guide. This frequently takes a lot of time and work. Customers who visit these locations might not receive the information they want and frequently end up being misinformed. To correctly plan and execute a certain journey is laborious for the customer. Due to the manual nature of the current system, it is impossible to keep all the data up to date in terms of package specifics, booking timings, or the opening and closing hours of tourist attractions. The total number of travels is recorded in documents, and human computations are used, which might result in significant errors. Some businesses overcharge tourists for packages, while others demand excessive rates. Sometimes local transport drivers in certain places charge over price from the tourist because they consider tourist as outsider. It will affect over experience of the tourist journey.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- The existing system cannot be managed by online or through internet.
- It requires human effort.
- Keeping tangible copies and file records of vital information current is challenging.
- It will need more manual labour to produce the necessary reports.
- Time consuming.
- Digital money is not accepted.
- No security for the user data.

## 2.4 PROPOSED SYSTEM

The suggested system manages a central repository of all relevant information using a web-based and Android application. The system makes it simple to access the necessary data and book the required travel. Visitors can make online reservations for travel packages and select the destinations they wish to visit. Without any price the tourists get all information of the tourist's destination in the region. Our project also provides the help of local travel guides. The tourist can login to our websites, and they can use all facilities. The local guide can add the detail and photographs of unexplored places. The tourist can

---

book local transportation from our website and pay transportation cost through online payment mode. The transportation charge is calculated based on type of the cab; how much distance travelled etc. The distance calculated based on the longitude and latitude of the locations.so transportation charge or cab charge is totally convenient for tourists. Our system will give an opportunity to earn an income for guides and drivers.

## **2.5 ADVANTAGES OF PROPOSED SYSTEM**

Both the system's implementation and design are rather straightforward. The system works in practically every case and uses relatively few system resources. It has the following attributes:

### **➤ Ensure more security: -**

Unauthorized access must be prevented in order for data to stay safe. Data protection means that they are shielded against different types of erasure. Security, integrity, privacy, and confidentiality are the four connected problems that make up the system security challenge. Security is maintained by requiring a username and password to sign in. As we use secured databases to maintain the papers, it will also ensure data security.

### **➤ Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

### **➤ Better service: -**

Our system will eliminate the need for storage and a massive amount of hard copy paperwork. For performing the same activity, we can also save time and resources. The data can be kept for a longer time without losing any information.

### **➤ Tourist Feedback and Review: -**

Tourist can send their feedback on using our software and functionalities. Also the tourist can review about our travel packages and local guide. This help to new tourists can get an idea about quality of our packages.

### **➤ Market Analytics: -**

The first and most important thing every travel agency does before the start of their season is to get a strong read on the market. You can effectively build your market strategy if you have the appropriate tools at your disposal. You can make more effective pitches and close

---

more deals if you have the correct market information.

**➤ Convenient and Cashless: -**

It is very convenient for tourists for book transportation and book guides through our system. Also, payments are accepted through online system. So, tourist did not worry about the liquid money. After each payment the tourist receive a receipt for their payment.

**➤ Provide safety and Flexibility: -**

In our system all the cab drivers and guide are carefully verified by our administration team by checking their identities and proofs. So, tourist did not need to worry about their safety while booking our cab drivers and guides. The tourists can book the cabs and guides from our website in any time. So, it is very flexible for tourists.

**➤ Professional Service: -**

Our teams handle all data's and enquires from the users in systematic way. All users records are stored in our database system.

---

## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

### **3.1 FEASIBILITY STUDY**

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organisation in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. The premise for a feasibility study is the system proposal's viability, which includes the impact on the organisation, ability to meet user needs, and effective use of resources. As a result, a feasibility evaluation is frequently performed before a new application is approved for development.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics:-

#### **3.1.1 Economical Feasibility**

Cost and benefit analyses are required to support the emerging system. Criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables.

Some of the significant financial queries raised during the initial probe include the following:

- The expenses carry out a comprehensive system investigation.
- The price of the software and hardware.
- The advantages in terms of lower expenses or fewer expensive mistakes.

The proposed system was created as part of a project; hence, there are no manual expenses associated with it. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably.

The cost of project, Smart Tourist Guiding System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

### **3.1.2 Technical Feasibility**

The system must first undergo a technical evaluation. The assessment of its viability must be built upon an overview design of the system's requirements in terms of input, output, programmes, and procedures. The inquiry must next advise the kind of equipment, necessary procedure for constructing the system, and means of operating the system once it has been designed after having identified an outline system.

The following technical difficulties came up throughout the investigation:

- Does the suggested technology work with the current technology?
- Can the system grow if it is improved?

The project should be designed in such a way that the necessary functionality and performance are met within the constraints. The project uses cryptographic methods and requires for a high resolution scanning device. The system may still be used even though the technology may become outdated after a while because a newer version of the same software still works with an earlier version. Therefore, this project only has a few limitations. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The system's CPU operated effectively. With an Intel i5 core processor, 8GB of RAM, and a 1TB hard drive, the system was also fast and responsive.

### **3.1.3 Behavioral Feasibility**

The following inquiries are part of the suggested system:

- Is there enough assistance for the users?
- Will the suggested system harm anyone?

Because it would accomplish the objectives after being developed and put into action, the project would be advantageous. After carefully examining all behavioural parameters, it is determined that the project is behaviorally feasible.

## 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

Processor - Intel core i5

RAM - 8 GB

Hard disk - 1 TB

### 3.2.2 Software Specification

Front End - HTML, CSS, ANDROID

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS, ANDROID

## 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP

PHP is a server-side scripting language used for both web development and general-purpose programming. PHP is now used by 2.1 million web servers and more than 244 million webpages. The reference version of PHP, which Rasmus Ledorf created in 1995, is now created by the PHP group. PHP:Hypertext Preprocessor is the current meaning of the recursive acronym PHP, which previously stood for personal home page. The PHP processor module on a web server translates the PHP code to create the finished web page. Instead of calling an external file, PHP commands can be directly put into an HTML source file to handle data.

The GNU General Public License is incompatible with PHP because it has evolved to incorporate a command-line interface feature and can be used independently due to restrictions on the usage of the term PHP (GPL). Most web servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems.

### 3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

A database is a planned collection of data. Anything might be it, from a simple grocery list to a photo gallery to the vast amount of information in a company network. Data included in a computer database must be added to, accessed, and processed using a database management system, such as MySQL Server. Because computers are so good at processing enormous amounts of data, database management systems—whether employed as standalone programmes or as a component of other applications—are crucial to computing.

- **MySQL databases are relational.**

A relational database stores the data in separate tables rather than consolidating it into one enormous warehouse. The database structures are stored in physical files that are optimised for speed. A flexible programming environment is offered by the logical model, which includes objects like databases, tables, views, rows, and columns. You could create rules to govern the relationships between different data fields, such as one-to-one, one-to-many, unique, compulsory or optional, and "pointers" between other tables. Since a well-designed database upholds these constraints, your application won't ever run into inconsistent, duplicate, orphan, out-of-date, or missing data. The prefix "SQL" in MySQL stands for "Structured Query Language". SQL is the most widely used standard language for database access. You might explicitly enter SQL (for instance, to generate reports), embed SQL statements into other languages' code, or use a language-specific API that hides the SQL syntax depending on your programming environment. SQL is specified by the ANSI/ISO SQL Standard. The SQL standard has undergone numerous changes since its creation in 1986. In this text, "SQL92," a 1992 standard, is referenced to.

The 1999 version of the standard is referred to as "SQL: 1999," and the most recent version is referred to as "SQL: 2003." The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

Anyone can use and modify the software because it is open source. The MySQL software is available for free download and usage online by anyone. You are free to examine the source code and modify it as necessary. The GPL (GNU General Public License) is used by the MySQL software to specify what you are allowed to do and are not allowed to do with the software in certain circumstances. If the GPL makes you uncomfortable or if you need to incorporate MySQL code into a for-profit application, you can buy a commercially licenced version from us. Check out the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If it is what you're after, you should try it. MySQL Server may run easily on a desktop or laptop and needs little to no maintenance in addition to your other apps, web servers, and other software. If you dedicate an entire system to MySQL, you can change the settings to utilise all the RAM, CPU, and I/O power.

- **MySQL Server works in client/server or embedded systems.**

A multi-threaded SQL server, several client programmes and libraries, management tools, and a wide variety of application programming interfaces make up the client/server system known as the MySQL Database Software (APIs). In addition, we offer MySQL Server as a built-in multi-threaded library that you can incorporate into your software to produce a standalone solution that is more compact, quick, and easy to use.

### **3.3.3 ANDROID**

Android OS is an operating system for mobile platforms that primarily powers tablets and smartphone gadgets. An operating system built on the Linux kernel, a graphical user interface, a web browser, and end-user programmes that can be downloaded make up the Android platform.

The operating system was created to function on relatively cost handsets with traditional numeric keypads, despite the fact that the early Android demos featured a basic Touchscreen smartphone with a big VGA screen. The Android distribution was released under the Apache v2 open-source licence, which allows for the creation of different OS variants for a variety of hardware, including gaming consoles and digital cameras.

Google Maps, YouTube, and Google Play are just a few of the proprietary apps that come pre-installed on most Android handsets, despite the fact that Android is based on open source software.

### Android OS features

- Bluetooth
- GPS
- 3G,4G,5G communication protocols, like EV-DO and HSDPA
- Wi-Fi
- SMS and MMS messaging
- video/still digital cameras
- Edge
- accelerated 3D graphics
- accelerometers
- Compasses
- multitasking applications

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. Design is a creative process. A good design is the key to a system that works effectively. "Design" is the process of using many approaches and concepts to thoroughly outline a process or a system so that it can be physically implemented. The process of using several approaches and concepts to specify a tool, a procedure, or a system in sufficient detail to enable its physical actuality is one way to put it. Software design serves as the technical foundation of the software engineering process, regardless of the development paradigm used. The system design process results in the development of the architectural details required to create a system or product. This programme has undergone the best design phase, which involves fine-tuning all efficiency, performance, and accuracy levels. During the design process, a document that is user-oriented is transformed into a document for programmers or database professionals. System design goes through two stages: logical design and physical design.

## 4.2 UML DIAGRAM

The software system artefacts are specified, visualised, built, and documented using a standard language called UML. The Object Management Group (OMG), which was in charge of creating UML, received a draft of the UML 1.0 standard in January 1997.

UML stands for Unified Modeling Language. UML is different from other well-known programming languages like C++, Java, COBOL, etc. Software designs are made using a visual language called UML. UML is a general-purpose visual modelling language used for the definition, design, development, and documentation of software systems. Although representing software systems is the most popular use of UML, it is not the only one. Additionally, it is employed to simulate non-software-based systems. The process flow in the manufacturing facility, etc. UML is not a programming language, but there are tools that can create code from UML diagrams in a number of other languages. UML is intimately connected to the analysis and design of objects-oriented systems. After considerable standardisation, UML is currently an OMG standard. By utilising all of a system's elements and connections, a full UML diagram shows the whole system.

The visual effect of the UML diagram is the most important factor in the entire process. Utilizing all the other elements completes it. The UML includes the next nine diagrams.

---

- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

#### **4.2.1 USE CASE DIAGRAM**

A use case diagram serves as an example of the interactions between system parts. Locating, outlining, and organizing system needs are done through use cases. In this context, the word "system" refers to something that is being created or run, such as a website for product sales and mail-order services. UML (Unified Modeling Language), a standard language for modelling actual products and systems, uses use case diagrams..

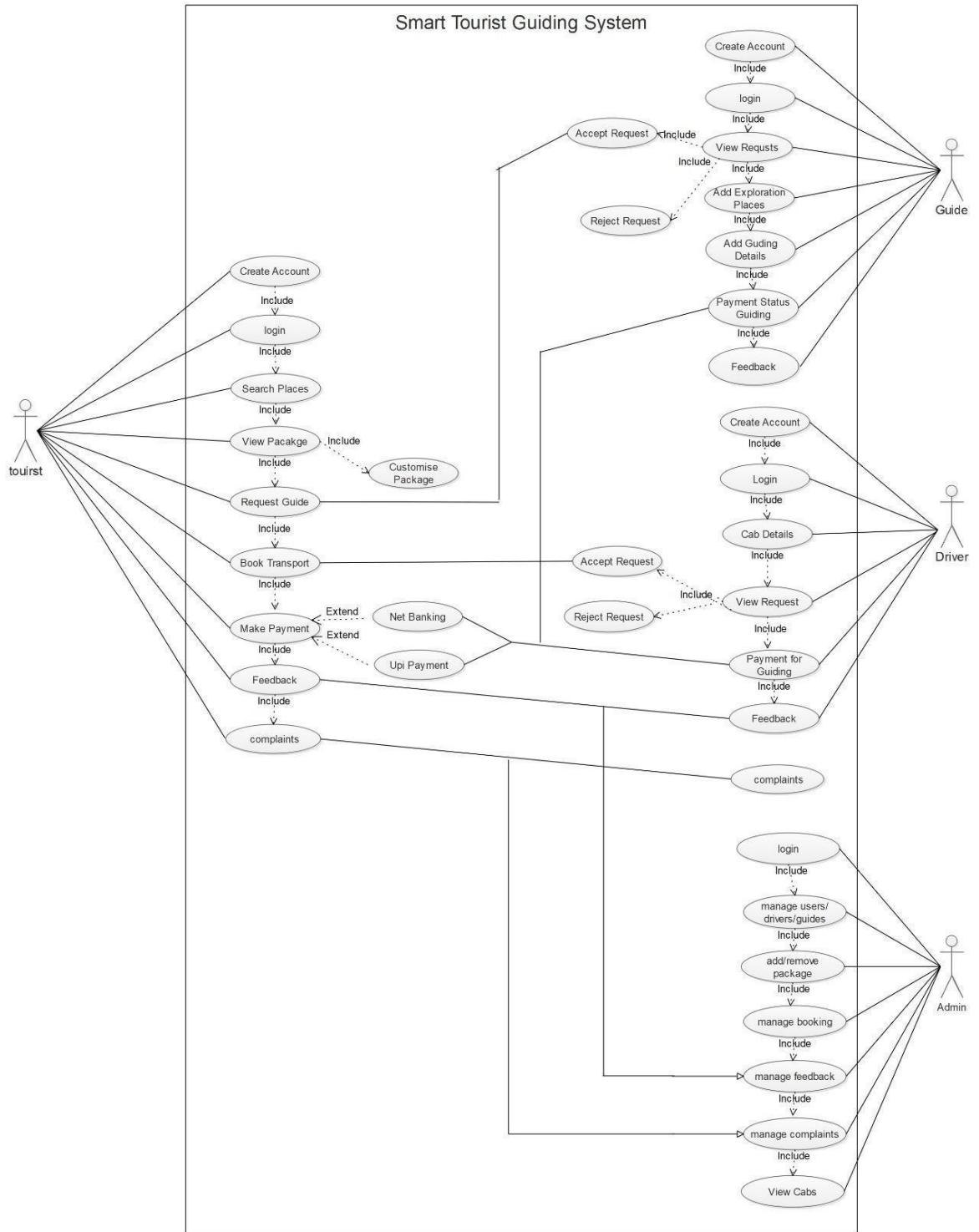
Planning general requirements, validating hardware designs, testing and debugging software products while they are still in development, creating online help resources, and finishing customer support-focused tasks are a few examples of system objectives. For instance, customer support, item ordering, catalogue updating, and payment processing are examples of use cases in a setting of product sales. A use case diagram consists of four components.

- The boundary, which isolates the system of interest from its surroundings.
- The performers, who are typically system participants identified by the roles they play.
- The actors within and around the system play the roles specified by the use cases.
- The connections and interactions between the actors and use cases.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- The naming of a use case is crucial. It is important to have a name that makes it obvious what tasks are being carried out.
- Assign appropriate names to the actors.
- Clearly show relationships and linkages in the diagram.
- Keep in mind that the diagram's main purpose is to represent the needs; avoid attempting to show every relationship that could exist.
- When required, jot down important information to help you remember it.

### Usercase Diagram For Smart Tourist Guiding System



## 4.2.2 SEQUENCE DIAGRAM

A sequence diagram fundamentally depicts the sequential order in which events occur or how they interact with one another. Event diagrams and event scenarios are other names for sequence diagrams. Sequence diagrams display the actions performed by a system's parts in a time-based manner. These diagrams are widely used by businesspeople and software engineers to document and explain the requirements for new and existing systems.

### Sequence Diagram Notations –

- i. **Actors** – In a UML diagram, an actor represents a particular sort of role that interacts with the system's constituents. An actor is never included in the UML diagram's depiction of the system we want to explain. For a range of roles, including those of human users and other external topics, we use actors. An actor is shown using the stick person notation in a UML diagram. A sequence diagram could have several actors.
- ii. **Lifelines** – A lifeline is a named component that displays a particular participant in a sequence diagram. A lifeline essentially stands in for each occurrence in a sequence diagram. In a sequence diagram, the lifeline components are at the top.
- iii. **Messages** – Messages are used to show how things may communicate with one another. On the lifeline, the messages are shown in reverse chronological order. Messages are represented by arrows. The two major elements of a sequence diagram are lifelines and messages.

The following categories serve as general classifications for messages:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message

- Reply Message
- Found Message
- Lost Message

**iv. Guards** – We use guards in the UML to model scenarios. We employ them when we need to restrict the flow of communications while pretending that a condition has been satisfied. Software developers rely on guards to let them know when a system or particular process has restrictions.

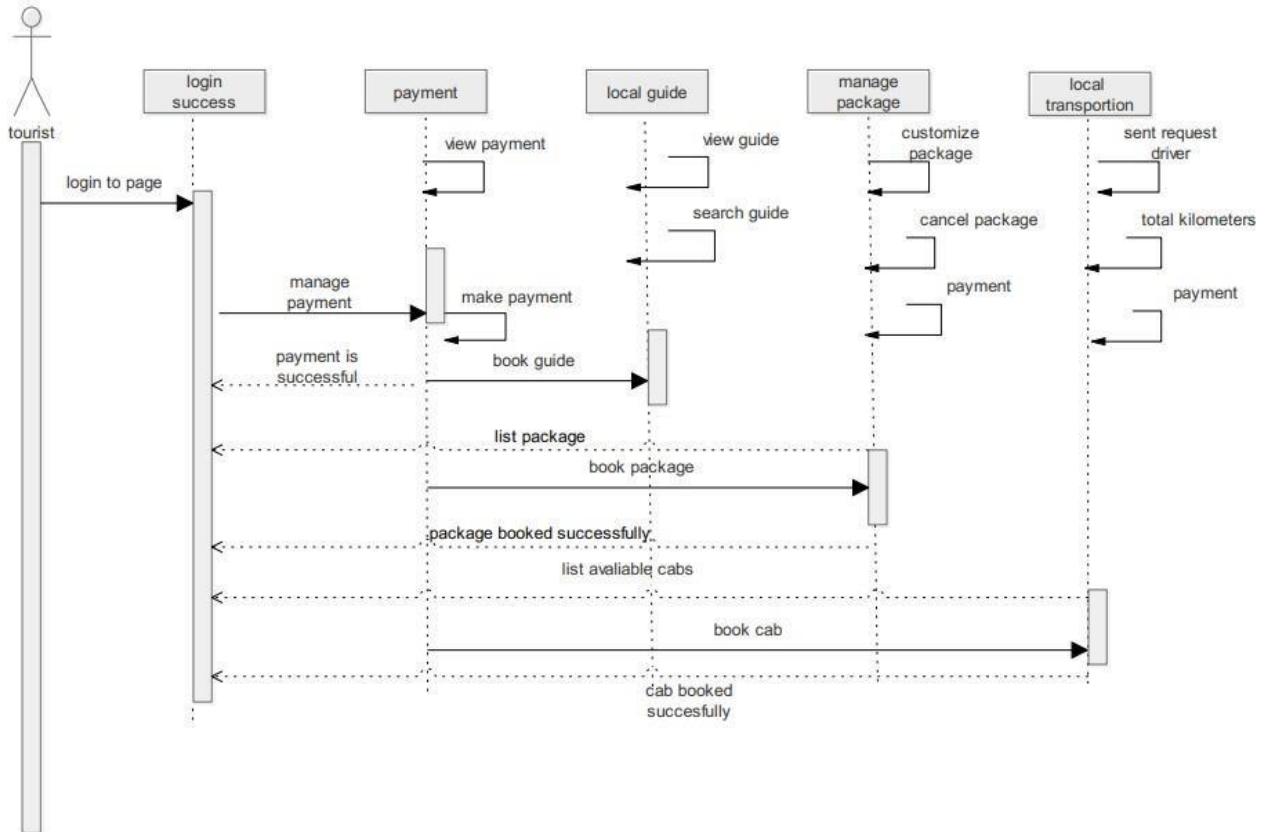
#### **Uses of sequence diagrams –**

- Used to simulate and visualise the reasoning behind a complex function, process, or method.
- They're also employed to display the specifics of UML use case diagrams.
- Used to comprehend the precise operation of present or upcoming systems.
- Imagine the flow of information between different system elements or objects.

### Sequence Diagram of Smart Tourist Guiding System

#### Sequence Diagram Of Smart Tourist Guiding System

#### Users In Smart Tourist Guiding System

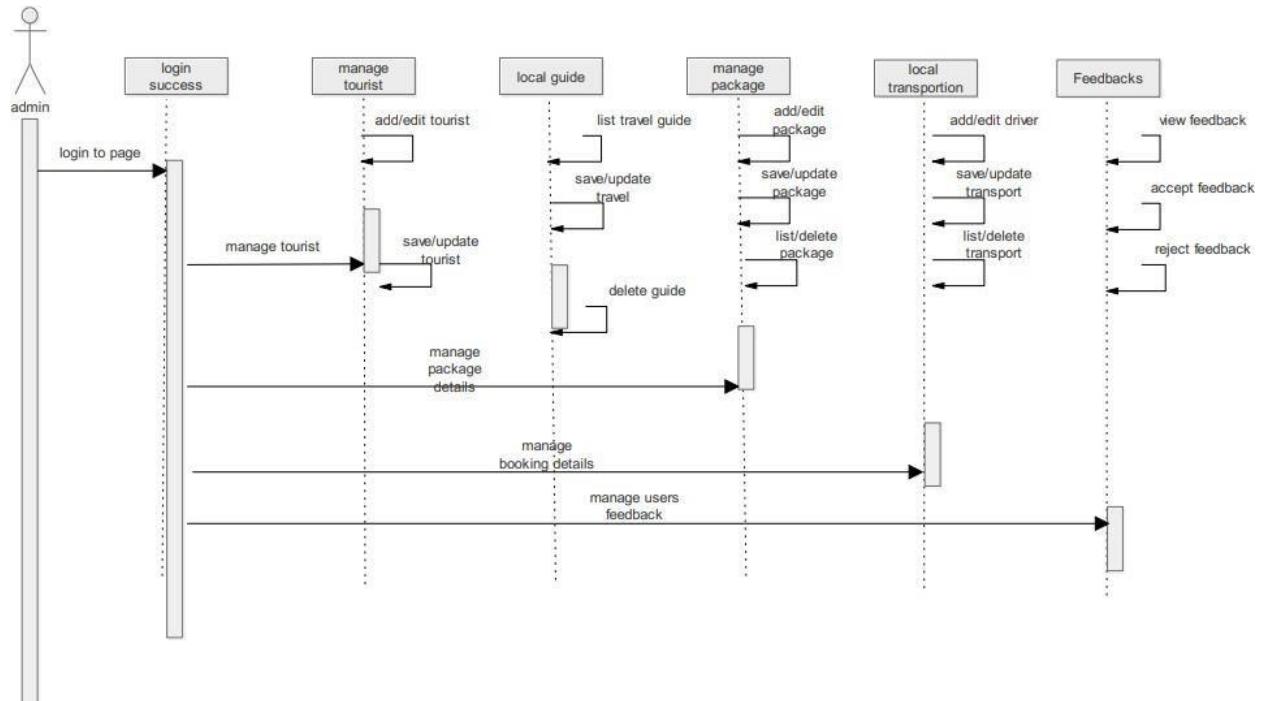


## Sequence Diagram of Smart Tourist Guiding System

### Admin In Smart Tourist Guding System

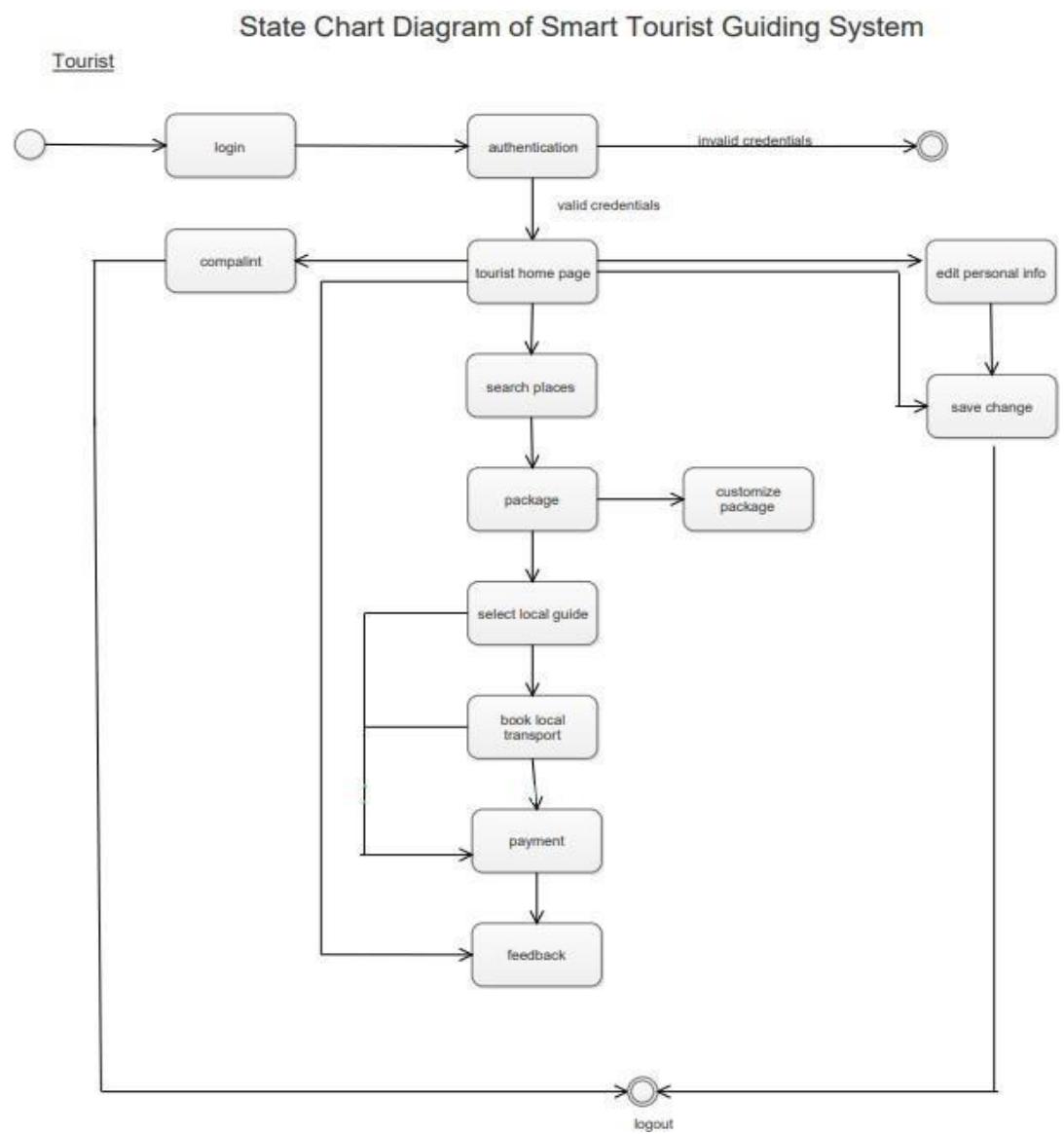
#### Sequence Diagram Of Smart Tourist Guiding System

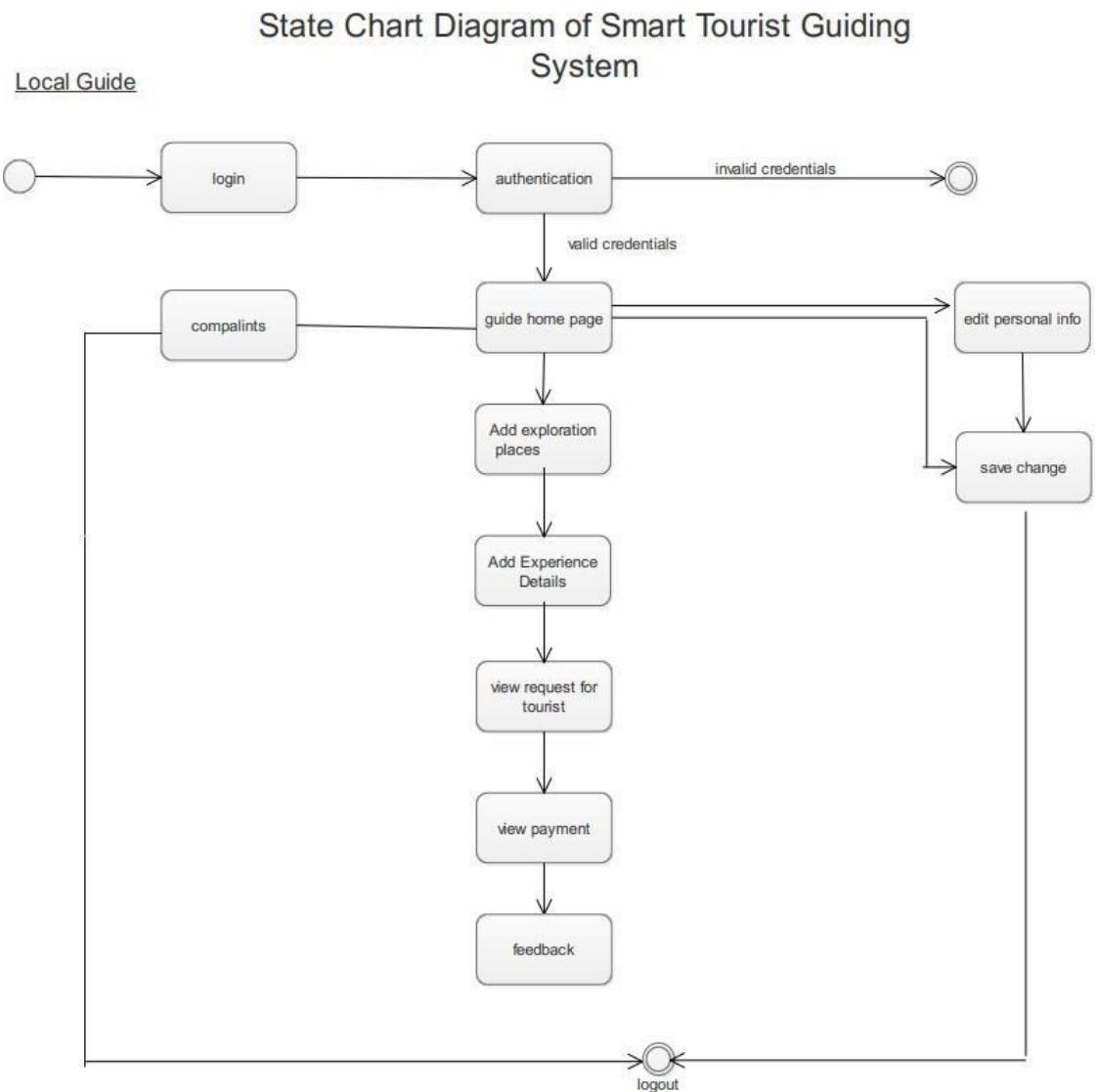
#### Admin In Smart Tourist Guiding System



### 4.2.3 STATE CHART DIAGRAM

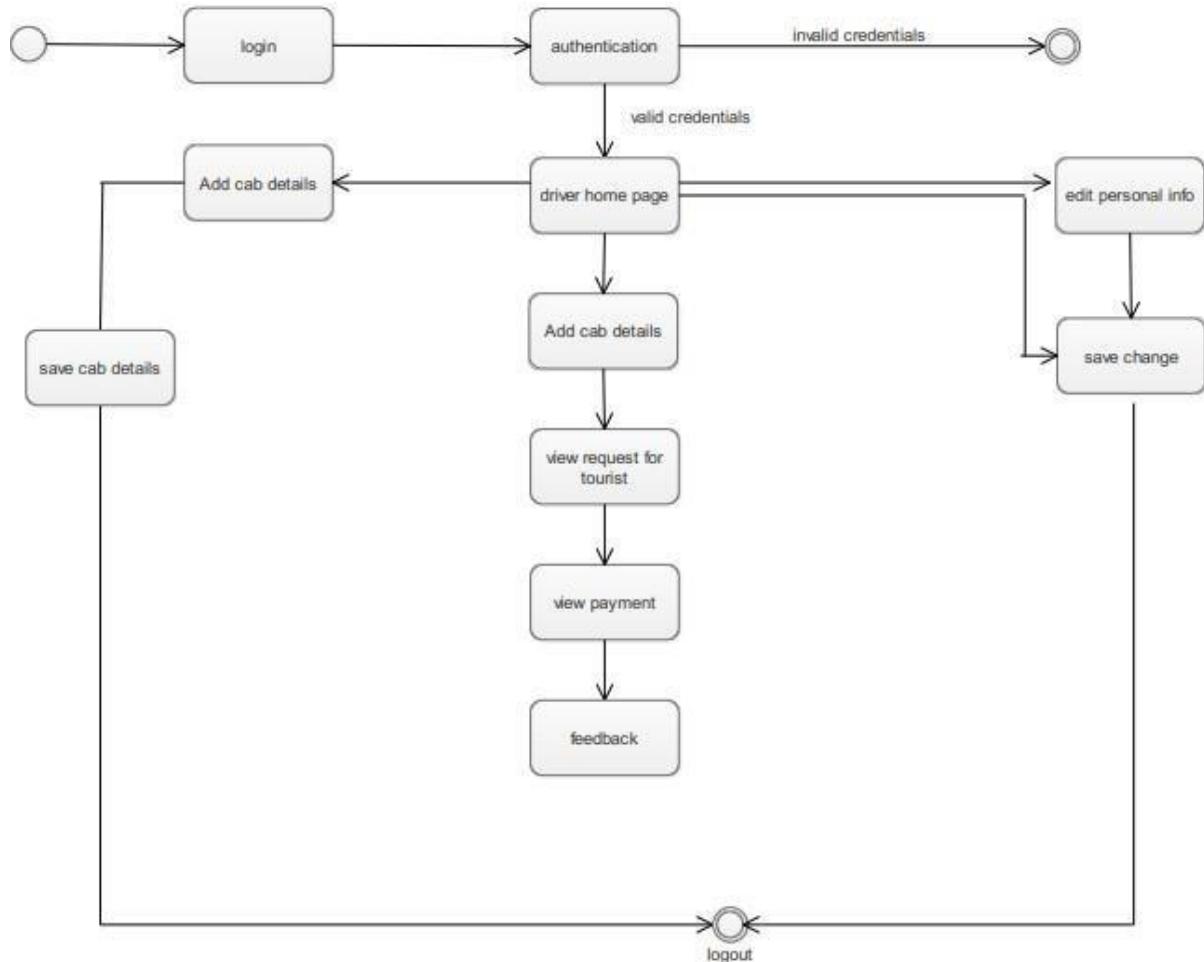
State diagrams are a common tool for displaying a software system's behaviour. A state machine diagram in a UML model can describe the behaviour of a class, a subsystem, a package, or even an entire system. Another term for it is a statechart or state transition diagram. Using statechart diagrams, we may effectively describe the interactions or communication that occur between external entities and a system. These diagrams are used to model the event-based system. An event can be used to control an object's state. The application system uses statechart diagrams to show the many states of an object.





### State Chart Diagram of Smart Tourist Guiding System

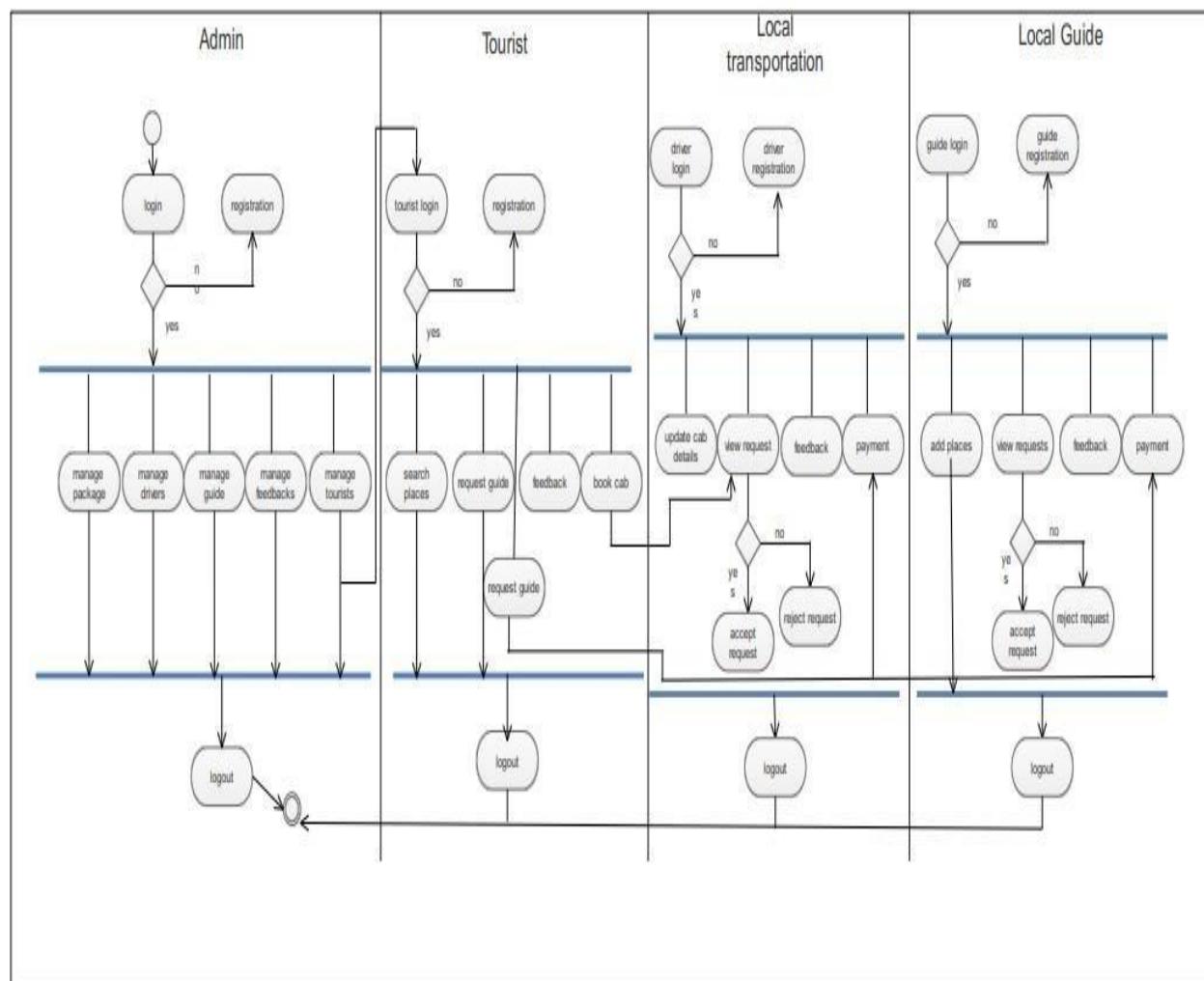
Driver



#### 4.2.4 ACTIVITY DIAGRAM

Activity diagrams demonstrate the coordination of several layers of activity abstraction necessary to deliver a service. Usually, a few operations are required to complete an event, especially when those operations must coordinate numerous different tasks. How the events in a single use case interact to one another is another frequent requirement, particularly in use cases where activities may overlap and require coordination. Additionally, it can be used to show how various interconnected use cases operate together to represent business workflows.

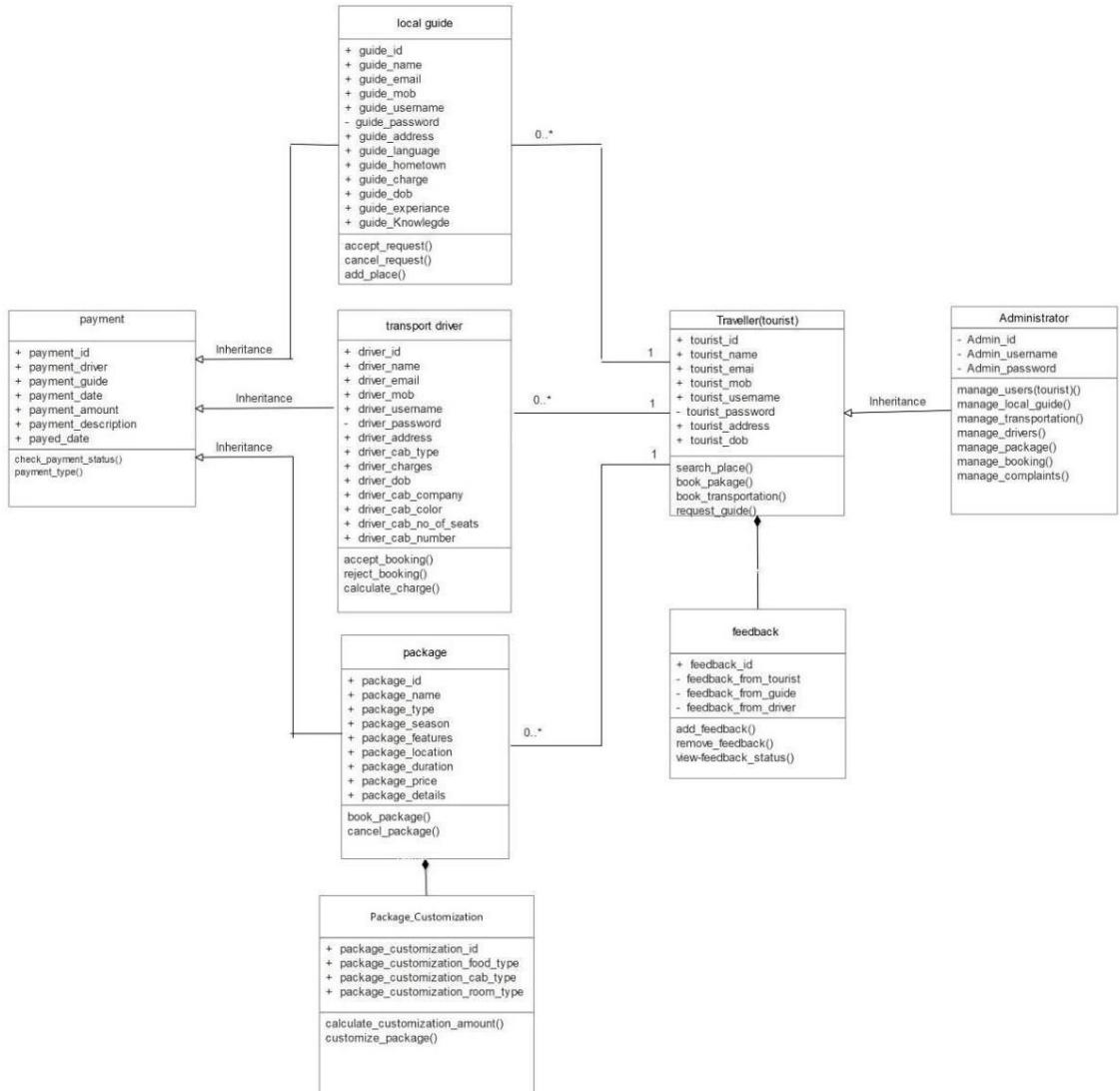
Activity Diagram Of Smart Tourist Guiding System



### 4.2.5 CLASS DIAGRAM

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are helpful for developing executable code for software programmes as well as for visualising, explaining, and documenting various system components. The attributes and actions of a class are described in a class diagram together with the restrictions placed on the system. Since class diagrams are the only UML diagrams that can be immediately translated using object-oriented languages, they are often employed in the modelling of object-oriented systems. An assortment of classes, interfaces, affiliations, collaborations, and limitations are displayed in a class diagram. Alternatively called a "structural diagram".

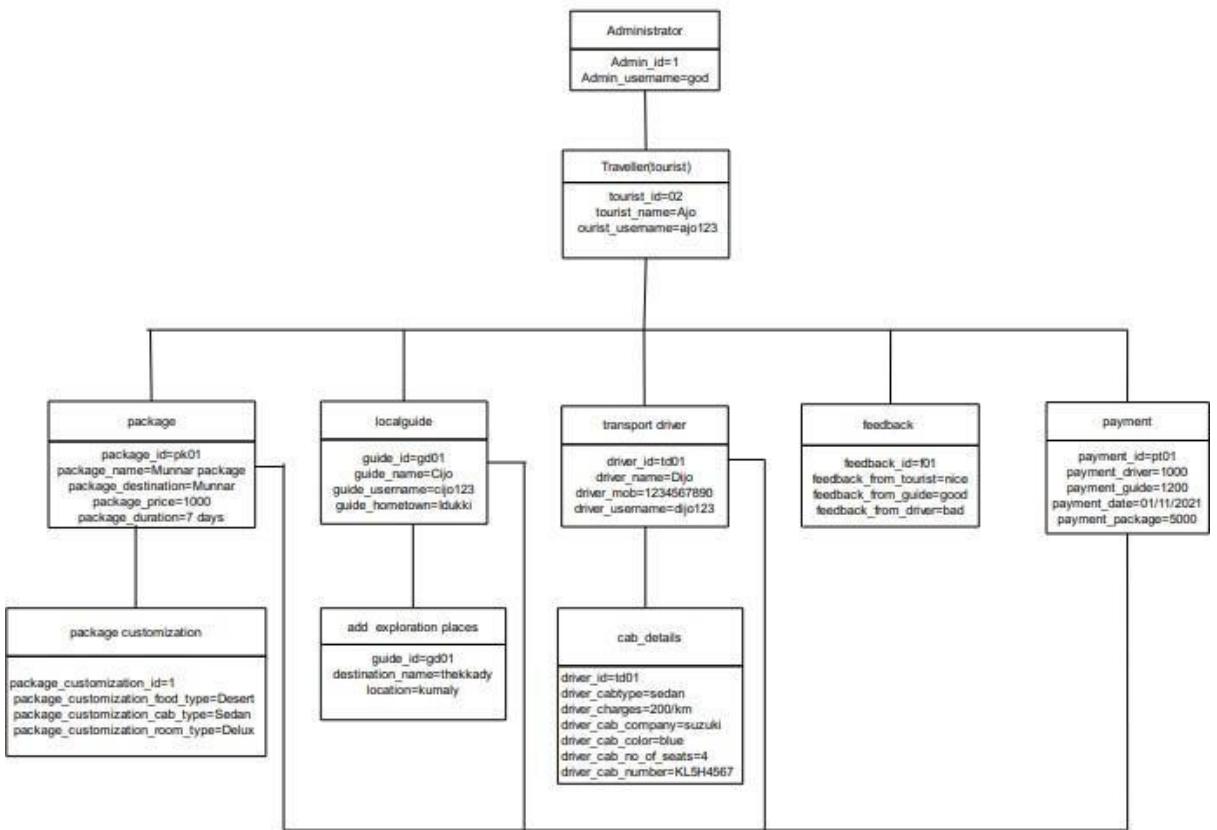
Class Diagram for Smart Tourist Guiding System



#### 4.2.6 OBJECT DIAGRAM

Class diagrams serve as the foundation for object diagrams, hence class diagrams are required. An object diagram is used to illustrate a class diagram instance. The core concepts used in class and object diagrams are similar. Object diagrams can be used to describe the static view of a system, which is a snapshot of the system at a certain point in time. Object diagrams are used to illustrate a set of items and their connections.

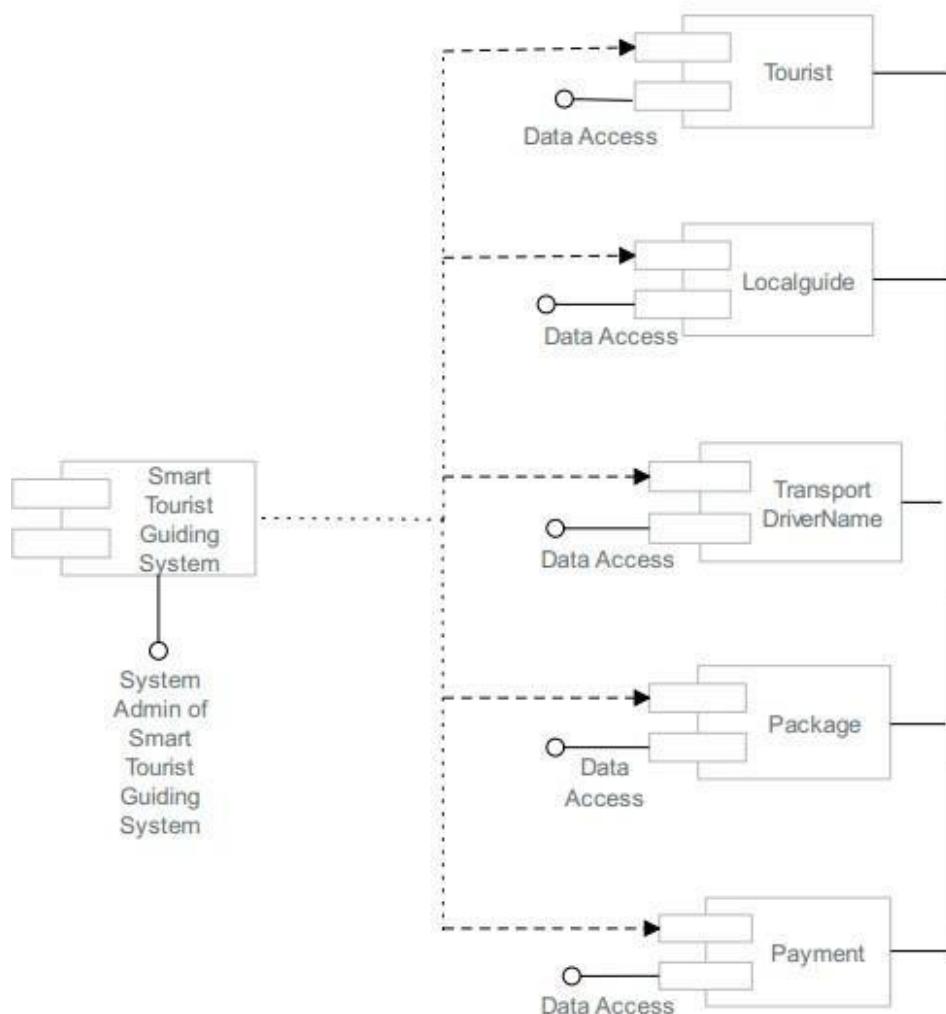
Object Diagram For Smart Tourist Guiding System



#### 4.2.7 COMPONENT DIAGRAM

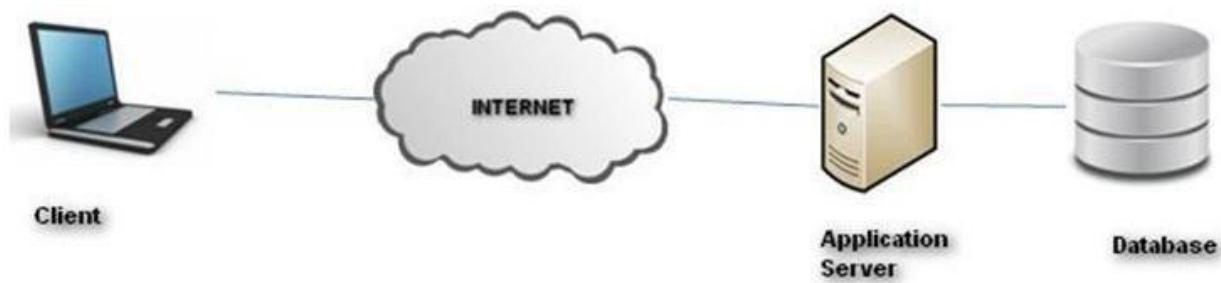
Component diagrams have different behaviours and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams can also be used to construct systems that can be run.

### Component Diagram of Smart Tourist Guiding System



#### 4.2.8 DEPLOYMENT DIAGRAM

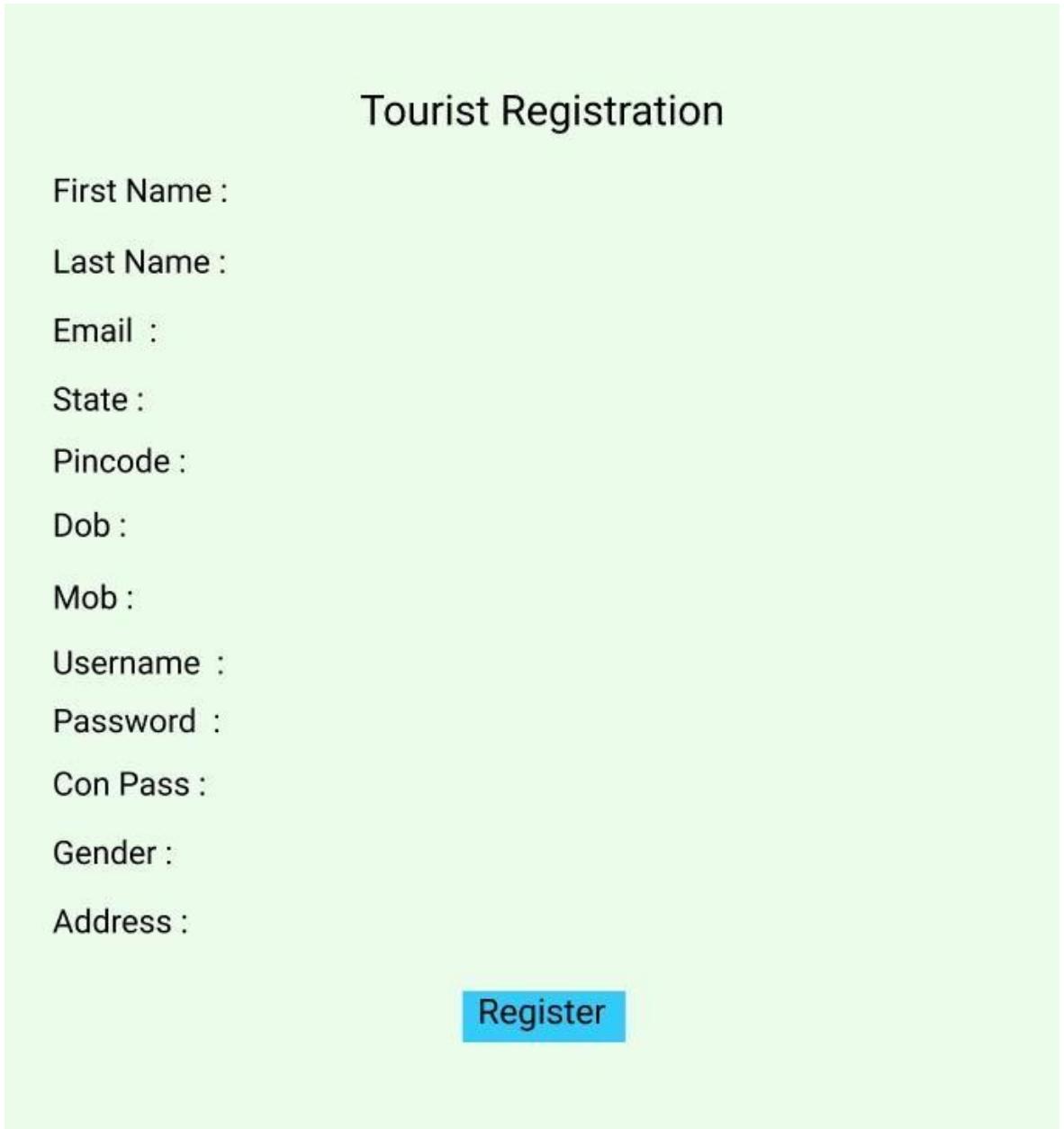
Deployment diagrams are used to depict the topology of a system's physical components, as well as the locations of software components. Deployment diagrams are used to describe a system's static deployment view. Nodes and their relationships are shown in deployment diagrams.



### 4.3 USER INTERFACE DESIGN

#### 4.3.1-INPUT DESIGN

Form Name : Tourist Registration Form



The image shows a user interface for a 'Tourist Registration' form. The title 'Tourist Registration' is centered at the top. Below it, there is a vertical list of input fields with labels: 'First Name', 'Last Name', 'Email', 'State', 'Pincode', 'Dob', 'Mob', 'Username', 'Password', 'Con Pass', 'Gender', and 'Address'. A large blue button labeled 'Register' is positioned at the bottom right of the form area.

Tourist Registration

First Name :

Last Name :

Email :

State :

Pincode :

Dob :

Mob :

Username :

Password :

Con Pass :

Gender :

Address :

Register

Form Name : Guide Registration Form

## Guide Registration

First Name :

Last Name :

Email :

State :

Pincode :

Dob :

Mob :

Username :

Password :

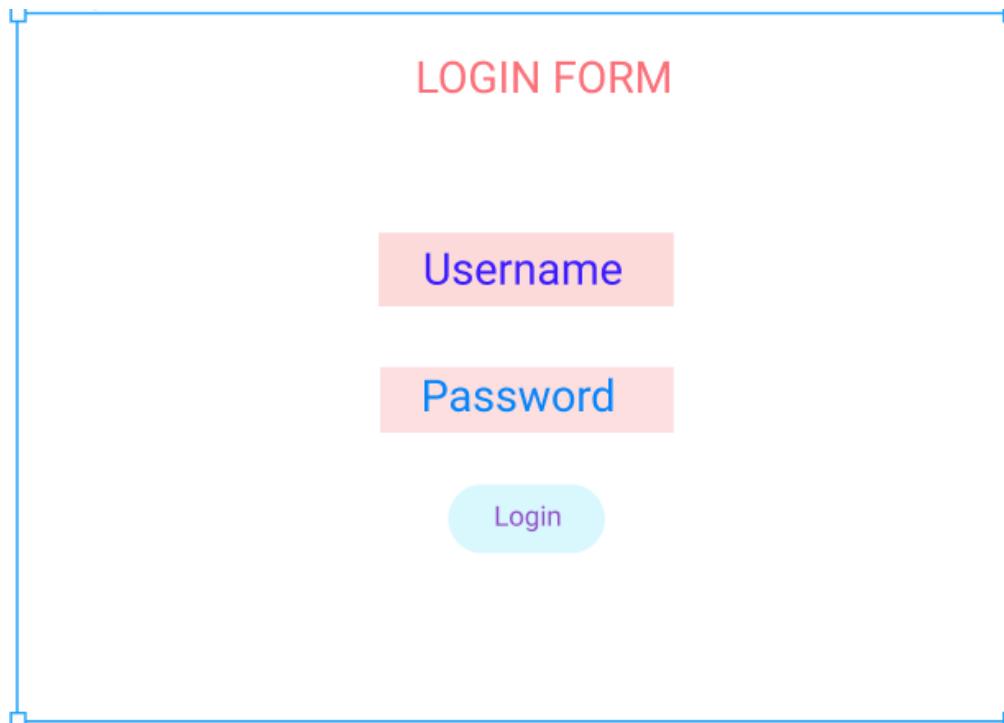
Con Pass :

Gender :

Address :

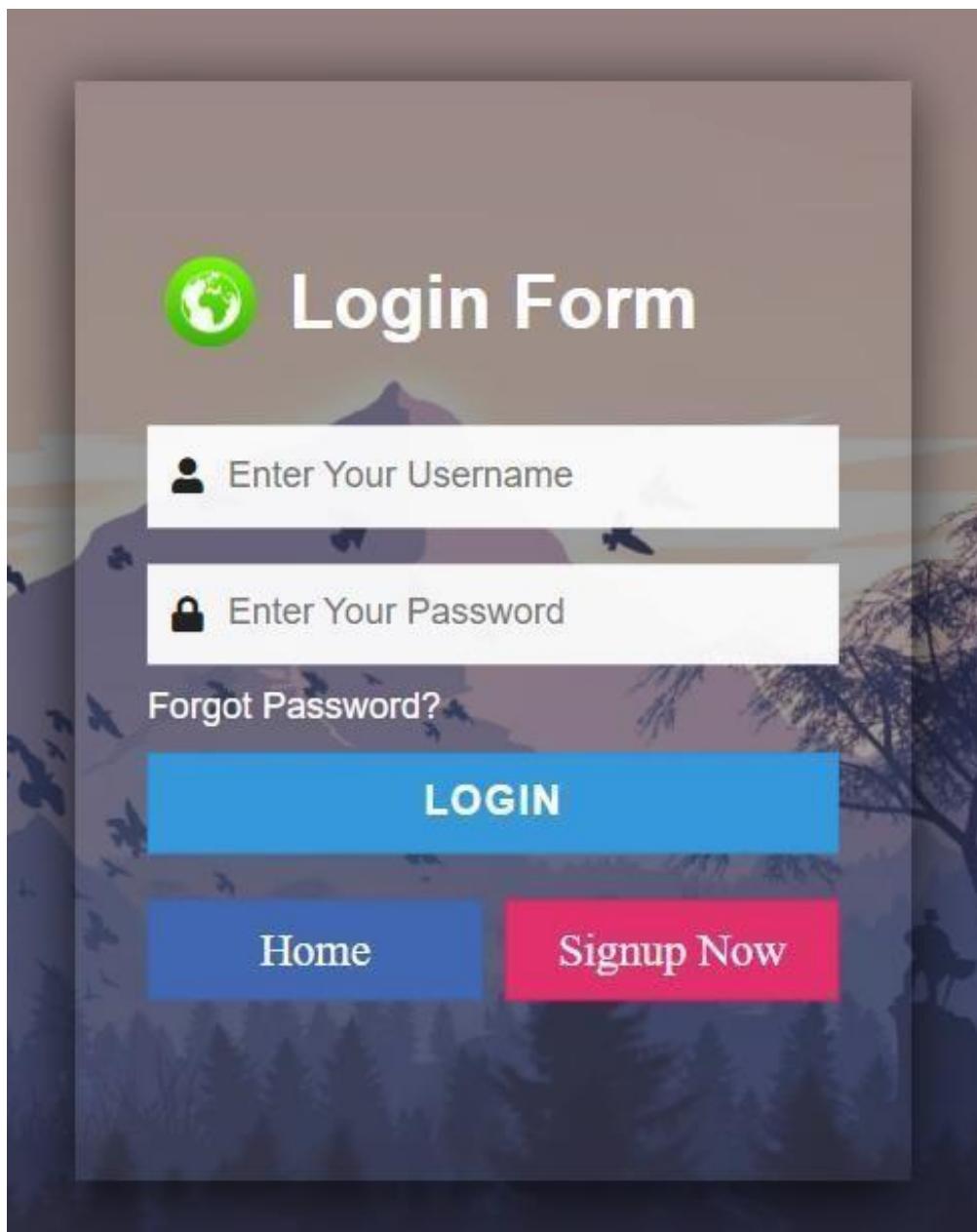
Register

Form Name : User Login Form



### 4.3.2 OUTPUT DESIGN

#### User Login Form



**Tourist Registration Form**

**Registration Form**

Name  First Name  Last Name

Email

State  Pincode :  Pincode...

Dob  dd - mm - yyyy

Moblie :  Mobile...

Username  Password :  Enter password

Con.Pass  confirm password

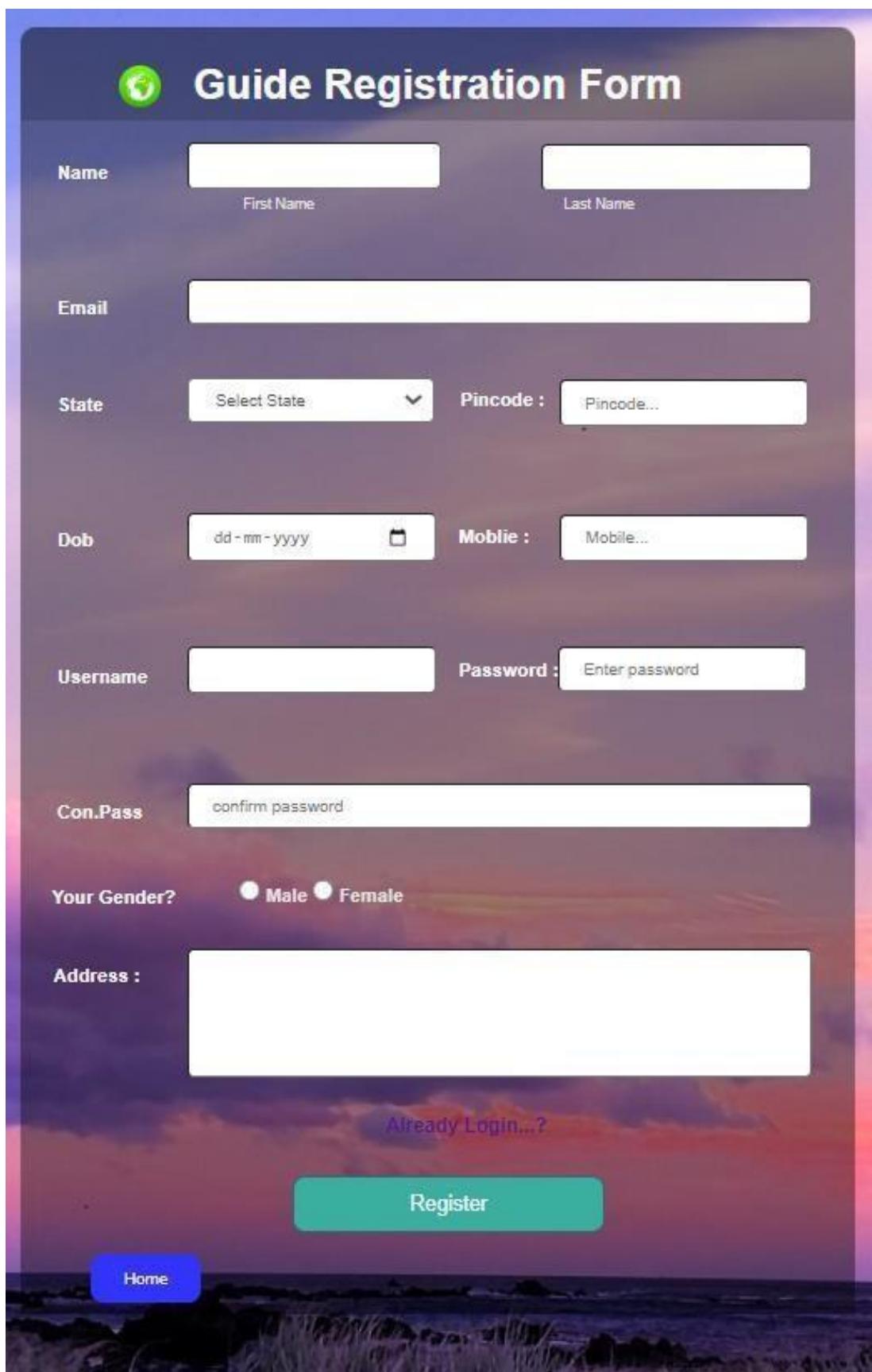
Your Gender?  Male  Female

Address :

Already Login...?

**Register**

**Home**

**Guide Registration Form**

## Guide Registration Form

**Name**    
First Name  Last Name

**Email**

**State**  Select State  **Pincode :**  Pincode...

**Dob**  dd-mm-yyyy  **Mobile :**  Mobile...

**Username**  **Password :**  Enter password

**Con.Pass**  confirm password

**Your Gender?**  Male  Female

**Address :**

[Already Login...?](#)

**Register**

[Home](#)

## 4.4. DATABASE DESIGN

A database is an organised system with the ability to store information and gives users easy access to that information. The data in any database is what it's there for, thus it needs to be protected.

The process of designing a database involves two steps. The initial phase is to gather the user demands, after which a database is built to as clearly as possible satisfy these requirements. This method, referred to as information level design, is used without the involvement of any DBMSs.

In the second step, the information level design for the specific DBMS that will be used to build the system in question is transformed into a design. The properties of the specific DBMS that will be utilised are discussed at the Physical Level Design stage. A database design runs parallel to the system design. The two major objectives of the database's data structure are outlined below.

- Data Integrity
- Data independence

### 4.4.1 Relational Database Management System (RDBMS)

The database is represented as a collection of relations in a relational paradigm. A table or file of records with values can be compared to each relation. In formal relational model terminology, a row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation. Numerous tables, each with a unique name, make up a relational database. Each row in a tale reflects a set of related values.

#### Relations, Domains & Attributes

A table is a connection or relation. The units of a table's rows are known as tuples. A tuple is an ordered collection of n items. Columns are the name given to attributes. There are connections between each table in the database already. This ensures the reliability of entity and referential connections alike. A domain D is a set of atomic values. A common method of defining a domain is by selecting a data type from which the domain's data values are derived. It is also useful to give the domain a name in order to make the values of the domain simpler to comprehend.

Each value in a relation is atomic and cannot be broken down.

## Relationships

- Table associations are established using keys. The two main keys that are most important are the Primary Key and the Foreign Key. Relationships for entity integrity and referential integrity may be made using these keys.
- For every Primary Key, Entity Integrity prohibits the usage of null values.
- Referential Integrity states that no primary key may have any null values.
- Referential Integrity: Each distinct Foreign Key value must match a Primary Key value in the same domain. Additional keys are the Super Key and Candidate Keys.

### 4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalised.

As implied by the name, it refers to arranging things in their natural state. By employing normalisation, the application developer seeks to create a consistent grouping of the data into suitable tables and columns, so users can immediately connect names to the data. Data redundancy, which places a significant burden on the computer's resources, is avoided through normalisation by deleting repetitive groups from the data. Among them are:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

### **First Normal Form**

According to the First Normal Form, each attribute's domain must only include atomic values, and each attribute's value in a tuple must be a single value from that domain. In other words, 1NF forbids using relationships as attribute values within tuples or relations within relations. Single atomic or indivisible values are the only attribute values that are permitted under 1NF. The data must first be entered into First Normal Form. This can be accomplished by separating data into tables of a similar type in each table. Depending on the needs of the project, a Primary Key or Foreign Key is assigned to each table. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation solely meets the constraints that include the primary key, it is said to be in first normal form.

### **Second Normal Form**

In relations when the primary key includes more than one attribute, no non-key attribute should be functionally dependent on a component of the main key. This entails creating a new relation for each partial key and dissecting it into its dependent properties. Keep your database's original primary key and any characteristics that are wholly dependent on it. Data that only depends on a small amount of the key can be removed with the help of this approach. A relation is said to be in second normal form if and only if the primary key satisfies all the criteria for first normal form and all of the relation's non-primary key properties are wholly dependent on the primary key alone.

### **Third Normal Form**

According to the Third Normal Form, a relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a group of non-key attributes. In other words, the main key shouldn't be transitively dependent. The deconstructed non-key characteristics are then arranged in relation to the non-key qualities that they functionally determine. Anything not fully dependent on the primary key is removed using this process. A relation is only regarded as being in third normal form when it is in second normal form and, more significantly, when none of its non-key qualities depend on any other non-key features.

## TABLE DESIGN

**Table 1: tbl\_admin\_login****Primary key: admin\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
admin_id	int	Primary key	Admin login id
admin_username	Varchar (30)	Not Null	Admin username
admin_password	Varchar (30)	Not Null	Admin password

**Table 2: tbl\_registration****Primary key: user\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
user_id	int	Primary key	User reg id
firstname	Varchar (10)	Not Null	User firstname
lastname	Varchar (10)	Not Null	User lastname
address	Varchar (100)	Not Null	User address
emails	Varchar (10)	Not Null	User email
state	Varchar (10)	Not Null	User state
pincode	Varchar (10)	Not Null	User pincode
dobs	Varchar (10)	Not Null	User dobs
phone	Varchar (100)	Not Null	User phone
gender	Varchar (10)	Not Null	User gender
login_id	int	Foreign key	Login id

**Table 3: tbl\_driver\_profile****Primary key: pro\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
pro_id	Int	Primary key	Driver Profile ID
image	Varchar (30)	Not Null	Driver Image
image_url	Varchar (100)	Not Null	Driver Image URL
login_id	Int	Foreign key	Login id

**Table 4: tbl\_login****Primary key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
login_id	Int	Primary key	Login id
username	Varchar (10)	Not Null	Gui/Tou username
password	Varchar (10)	Not Null	Gui/Tou password
role	Varchar (10)	Not Null	User type
status	Int	Not Null	Status of user(0,1)

**Table 5: tbl\_package****Primary key: package\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
package_id	Int	Primary key	Package id
package_name	Varchar (30)	Not Null	Package name
package_type	Varchar (30)	Not Null	Package type
package_season	Varchar (30)	Not Null	Package season
package_features	Varchar (30)	Not Null	Package feature
package_location	Varchar (30)	Not Null	Package location
package_duration	Varchar (30)	Not Null	Package duration
package_price	Varchar (30)	Not Null	Package price
package_details	Varchar (30)	Not Null	Package details
package_image	Varchar (30)	Not Null	Package image
package_status	Int	Not Null	Status in (0,1)

**Table 6: tbl\_place****Primary key: place\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
place_id	Int	Primary key	Place id
place_name	Varchar (10)	Not Null	Place name
place_district	Varchar (10)	Not Null	Place district
place_description	Varchar (10)	Not Null	Place description
place_image1	Varchar (10)	Not Null	Place image 1
place_image2	Varchar (10)	Not Null	Place image 2
place_image3	Varchar (10)	Not Null	Place image 3
place_image4	Varchar (10)	Not Null	Place image 4
login_id	Int	Foreign key	Login id
status	Int	Not Null	Status in (0,1,2)

**Table 7: tbl\_feedback****Primary key: feed\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
feed_id	Int	Primary key	Feedback id
feed_message	Varchar (200)	Not Null	Message
feed_status	Int	Not Null	Status (1,0,2)
login_id	Int	Foreign key	Login id

**Table 8: tbl\_contact****Primary key:** contact\_id**Foreign key:** Nil

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
contact_id	Int	Primary key	Contact id
contact_name	Varchar(10)	Not Null	Contact name
contact_email	Varchar(10)	Not Null	Contact email
contact_message	Varchar(10)	Not Null	Contact message
contact_status	Int	Not Null	Contact status
date_entered	timestamp	Not Null	Date entered

**Table 9: tbl\_booking****Primary key:** booking\_id**Foreign key:** login\_id, package\_id

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
booking_id	Int	Primary key	Booking id
book_date_start	Varchar(30)	Not Null	Starting date
book_date_end	Varchar(30)	Not Null	Ending date
book_no_person	Varchar(30)	Not Null	Number of person
login_id	Int	Foreign key	Login id
package_id	Int	Foreign key	Package_id
status	Int	Not Null	Status (1,0,2)
booked_date	timestamp	Not Null	Booked date

**Table 10: tbl\_regpics****Primary key:** regpics\_id**Foreign key:** login\_id

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
regpics_id	Int	Primary key	Regpics id
prof_img	Varchar(30)	Not Null	Profile image
pro_proof	Varchar(30)	Not Null	Proof
proof_type	Varchar(30)	Not Null	Proof type
login_id	Int	Not Null	Login id
proof_no	Varchar(30)	Not Null	Proof Number

**Table 11: tbl\_guidedetails****Primary key:** gd\_id**Foreign key:** login\_id

Field	Datatype	Constraints	Description
gd_id	Int	Primary key	Guidedetails id
gd_district	Varchar(30)	Not Null	Guide District
gd_knowledge	Varchar(30)	Not Null	Guide knowledge
gd_mlang	Varchar(30)	Not Null	Guide mother tongue
gd_flang	Varchar(30)	Not Null	Guide first language
gd_slang	Varchar(30)	Not Null	Guide second language
gd_tlang	Varchar(30)	Not Null	Guide third language
gd_desc	Varchar(120)	Not Null	Guide description
gd_experiace	Varchar(30)	Not Null	Guide experience
gd_payment	Varchar(6)	Not Null	Guide payment
login_id	Int	Foreign key	Login id

**Table 12: tbl\_guide\_booking****Primary key:** gb\_id**Foreign key:** login\_id

Field	Datatype	Constraints	Description
gb_id	Int	Primary key	Guide booking id
gb_datef	Varchar(10)	Not Null	Guide date staring
gb_datee	Varchar(10)	Not Null	Guide date ending
login_id	Int	Foreign key	Login id
guide_id	Int	Not Null	Guide id
status	Int	Not Null	Status in (1,0,2)
booked_date	timestamp	Not Null	Booked date

**Table 13: tbl\_cab\_booking****Primary key:** cabbook\_id**Foreign key:** login\_id

Field	Datatype	Constraints	Description
cabbook_id	Int	Primary key	Cabbook_Id
booking_time	Varchar(10)	Not Null	Booking_Time
beginning	Varchar(10)	Not Null	Beginning
destination	Varchar(10)	Not Null	Destination
loc1_longitude	Varchar(10)	Not Null	Loc1_Longitude
loc1_latitude	Varchar(10)	Not Null	Loc1_Latitude
loc2_longitude	Varchar(10)	Not Null	Loc2_Longitude
loc2_latitude	Varchar(10)	Not Null	Loc2_Latitude
kilometers	Int	Not Null	Kilometers
booking_date	Varchar(10)	Not Null	Booking_Date
total_charge	Varchar(10)	Not Null	Total_Charge
login_id	Int	Foreign key	Login Id
status	Int	Not Null	Status in (1,0,2)
booked_date	timestamp	Not Null	Booked_Date
driver_id	Int	Not Null	Driver_Id

**Table 14: tbl\_cab\_details****Primary key: cab\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
cab_id	Int	Primary key	Cab_Id
cab_company	Varchar(10)	Not Null	Cab_Company
cab_model	Varchar(10)	Not Null	Cab_Model
cab_color	Varchar(10)	Not Null	Cab_Color
cab_reg_no	Varchar(10)	Not Null	Cab_Reg_No
cab_type	Varchar(10)	Not Null	Cab_Type
cab_no_seats	Varchar(2)	Not Null	Cab_No_Seats
cab_charge	Varchar(6)	Not Null	Cab_Charge
cariimg	Varchar(10)	Not Null	Cariimg
login_id	Int	Foreign key	Login Id

**Table 15: tbl\_cab\_payment****Primary key: cab\_payment\_id****Foreign key: login\_id, driver\_id, cabbook\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
cab_payment_id	Int	Primary key	Cab_Payment_Id
amount	Varchar(6)	Not Null	Amount
driver_id	Int	Foreign key	Driver_Id
login_id	Int	Foreign key	Login Id
cabbook_id	Int	Foreign key	Cabbook_Id
payment_status	Int	Not Null	Status in (1,0,2)
payed date	timestamp	Not Null	Payed Date

**Table 16: tbl\_car\_document****Primary key: doc\_id****Foreign key: login\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
doc_id	Int	Primary key	Document Id
rc_book	Varchar(10)	Not Null	Rc Book
insurance	Varchar(10)	Not Null	Insurance
license	Varchar(10)	Not Null	License
license_img	Varchar(10)	Not Null	License Image
login_id	Int	Foreign key	Login Id

**Table 17: tbl\_dayplan1****Primary key: dayplan\_id****Foreign key: package\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
dayplan_id	Int	Primary key	Dayplan1 Id
Arrival	Varchar(10)	Not Null	Arrival
Hotel	Varchar(10)	Not Null	Hotel
Activity1	Varchar(10)	Not Null	Activity 1
Launch	Varchar(10)	Not Null	Lunch
Activity2	Varchar(10)	Not Null	Activity 2
addactivity1	Varchar(10)	Not Null	Addactivity 1
addactivity2	Varchar(10)	Not Null	Addactivity 2
Dinner	Varchar(10)	Not Null	Dinner
package_id	Int	Foreign key	Package_Id

**Table 18: tbl\_dayplan2****Primary key: dayplan2\_id****Foreign key: package\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
dayplan2_id	Int	Primary key	Dayplan2 Id
hotel	Varchar(10)	Not Null	Hotel
breakfast	Varchar(10)	Not Null	Breakfast
activity1	Varchar(10)	Not Null	Activity 1
lunch	Varchar(10)	Not Null	Lunch
activity2	Varchar(10)	Not Null	Activity 2
addactivity1	Varchar(10)	Not Null	Addactivity 1
addactivity2	Varchar(10)	Not Null	Addactivity 2
dinner	Varchar(10)	Not Null	Dinner
package_id	Int	Foreign key	Package_Id

**Table 19: tbl\_dayplan3****Primary key: dayplan3\_id****Foreign key: package\_id**

<b>Field</b>	<b>Datatype</b>	<b>Constraints</b>	<b>Description</b>
dayplan3_id	Int	Primary key	Dayplan3 Id
hotel	Varchar(10)	Not Null	Hotel
breakfast	Varchar(10)	Not Null	Breakfast
activity1	Varchar(10)	Not Null	Activity 1
lunch	Varchar(10)	Not Null	Lunch
activity2	Varchar(10)	Not Null	Activity 2
addactivity1	Varchar(10)	Not Null	Addactivity 1
addactivity2	Varchar(10)	Not Null	Addactivity 2
dinner	Varchar(10)	Not Null	Dinner
package_id	Int	Foreign key	Package_Id

**Table 20: tbl\_dayplan4**

**Primary key:** dayplan4\_id  
**Foreign key:** package\_id

Field	Datatype	Constraints	Description
dayplan4_id	Int	Primary key	Dayplan4 Id
hotel	Varchar(10)	Not Null	Hotel
breakfast	Varchar(10)	Not Null	Breakfast
activity1	Varchar(10)	Not Null	Activity 1
lunch	Varchar(10)	Not Null	Lunch
activity2	Varchar(10)	Not Null	Activity 2
addactivity1	Varchar(10)	Not Null	Addactivity 1
addactivity2	Varchar(10)	Not Null	Addactivity 2
dinner	Varchar(10)	Not Null	Dinner
package_id	Int	Foreign key	Package Id

**Table 21: tbl\_guide\_payment**

**Primary key:** guide\_payment\_id  
**Foreign key:** login\_id , gb\_id

Field	Datatype	Constraints	Description
guide_payment_id	Int	Primary key	Guide Payment Id
amount	Varchar(6)	Not Null	Guide Amount
login_id	Int	Foreign key	Login Id
guide_id	Int	Not Null	Guide Id
gb_id	Int	Foreign key	Guide Booking Id
payment_status	Int	Not Null	Status in (1,0,2)
payed_on	timestamp	Not Null	Payed Date

**Table 22: tbl\_guide\_rating**

**Primary key:** rating\_id  
**Foreign key:** login\_id

Field	Datatype	Constraints	Description
rating_id	Int	Primary key	Rating Id
comment	Varchar(200)	Not Null	Comments
rate	Varchar(10)	Not Null	Rate of Guide
login_id	Int	Foreign key	Login Id
guide_id	Int	Not Null	Guide Id

**Table 23: tbl\_package\_customization****Primary key:** pcus\_id**Foreign key:** login\_id, package\_id

Field	Datatype	Constraints	Description
pcus_id	Int	Primary key	Pcus Id
food_type	Varchar(10)	Not Null	Food Type
cab_type	Varchar(10)	Not Null	Cab Type
room_type	Varchar(10)	Not Null	Room Type
login_id	Int	Foreign key	Login Id
package_id	Int	Foreign key	Package Id

**Table 24: tbl\_payment****Primary key:** payment\_id**Foreign key:** login\_id, package\_id, booking\_id

Field	Datatype	Constraints	Description
payment_id	Int	Primary key	Payment Id
amount	Varchar(10)	Not Null	Total Amount
booking_id	Int	Not Null	Booking Id
package_id	Int	Foreign key	Package Id
login_id	Int	Foreign key	Login Id
payment_status	Int	Not Null	Status in (1,0,2)

## **CHAPTER 5**

## **SYSTEM TESTING**

## 5.1 INTRODUCTION

Software testing is the procedure of meticulously monitoring the way software is used to see if it works as expected. Software testing is usually used in combination with the phrases validation and verification. A product, including software, is validated by being examined or evaluated to see if it conforms with all pertinent requirements. Software testing, one sort of verification, also makes use of reviews, analyses, inspections, and walkthroughs. Validation is the process of ensuring that what has been specified corresponds to what the user really wants.

Other procedures that are typically connected to software testing include static analysis and dynamic analysis. Without actually running the code, static analysis examines the software's source code to look for errors and gather statistics. Dynamic analysis looks at the behaviour of software while it is in use to provide information like execution traces, timing profiles, and test coverage specifics.

Testing is a group of tasks that may be organised in advance and completed in a systematic way. Individual modules are tested first, followed by the integration of the entire computer-based system. There are several regulations that may be utilised as testing objectives, and testing is essential for the achievement of system testing objectives. the following:

A programme is tested by being run with the goal of identifying any errors.

- A test case with a high likelihood of detecting an unknown fault qualifies as a good test case.
- A test that finds an error that has not yet been found is successful.

A test that effectively accomplishes the aforementioned goals will discover software problems. Testing also reveals that the software functionalities appear to function in accordance with the specification and that the performance criteria appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

A programme must be tested for correctness to guarantee that it operates exactly as planned. This is significantly more difficult than it first appears to be, especially for large programmes.

## 5.2 TEST PLAN

The procedures that must be followed in order to fulfil various testing techniques are suggested in a test plan. The test plan specifies the activities that must be completed. Software developers are responsible for creating a computer programme, as well as any related documentation and data structures. It is always the software developers' job to test each of the program's individual parts to ensure that it serves the intended function. There is an impartial test group in order to address the problems of letting the developer assess what they have created (ITG). The specific objectives of testing should be stated in terms of numbers. Information on the mean time to failure, the cost to find and fix issues, the remaining defect density or frequency of occurrence, and the number of test work hours required for each regression test should all be included in the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

### 5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. the level of test complexity and the untested area determined for unit testing. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Testing of data flow through a module interface are important before beginning any additional tests. If data cannot correctly enter and exit the system, all other tests are useless. The unit test's selective analysis of execution routes is a crucial task. In order to properly redirect or stop work when an error does occur, error handling channels must be set up and fault scenarios must be anticipated in excellent design. Boundary testing is the last stage of unit testing. At its boundaries, software frequently fails.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

### **5.2.2 Integration Testing**

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

### **5.2.3 Validation Testing or System Testing**

The testing process has now come to an end. It was necessary to thoroughly test the system, covering all forms, codes, modules, and class modules. This type of testing is also known as system tests and black box testing.

The functional requirements of the programme are the main emphasis of the black box testing strategy. A software engineer can create sets of input conditions using Black Box testing in this situation to completely test each programme requirement.

Black box testing focuses on certain types of problems, such as incorrect or missing functionalities, interface flaws, data structure or external data access mistakes, performance defects, initialization failures, and termination errors.

### 5.2.4 Output Testing or User Acceptance Testing

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The software should consult the user and the perspective system while it is being developed in order to make any necessary adjustments. This is carried out in relation to the following things:

- Input Screen Designs,
- Output Screen Designs,

A variety of test data are used to conduct the aforementioned tests. The process of system testing requires the preparation of test data. The system under examination is then tested using the sample data that has been prepared. The system's flaws are once more discovered during testing, repaired with the help of the aforementioned techniques, and recorded for future use.

### 5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that could occur throughout the creation of the product. Any time of day may be used to do automation testing. It looks at the programme using scripted sequences. It then summarises what was discovered, and this data may be compared to results from prior test runs.

#### Benefits of Automation Testing

- Simplifies testing - Most SaaS and IT organisations routinely include testing in their daily operations. The trick is to keep things as basic as you can. Automation has a lot of advantages. The test scripts can be reused for automating test tools.
- Quickens the testing procedure - Machines and automated technologies operate more quickly than people. This is why we employ them, along with increased precision. Your software development cycles are subsequently shortened by this.
- Lessens the requirement for human supervision - Tests may be conducted at any time of day, including overnight. Additionally, when done automatically, this can lessen the possibility of human mistake.
- Detailed reporting capabilities - Test cases for different scenarios are carefully built for automation testing. These planned sequences can cover a lot of ground

and produce in-depth reports that are just impossible for a person to produce.

- Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This procedure may be made simpler with automation testing. Additionally, it can examine a greater test coverage than perhaps people can.

#### **5.2.6 Selenium Testing**

An open-source programme called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in a number of different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. Web application testing for cross-browser compatibility is automated using the Selenium testing tool. Whether they are responsive, progressive, or standard, it is utilised to assure high-quality web apps. Selenium is a free software programme.

## Test cases for a Login Page

Project Name: Smart Tourist Guiding System					
Login Test Case					
Test Case ID: Login		Test Designed By: Tom Joseph			
Test Priority (Low/Medium/High): High		Test Designed Date: 17-07-2022			
Module Name: Login Screen		Test Executed By: Ms. Nimmy Francis			
Test Title: Verify login with valid username and password		Test Execution Date: 18-07-2022			
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Username : Anish	User should be able to Login	User Logged in and navigated to User Dashboard	Pass
3	Provide Valid Password	Password:Anish@123			
4	Click on Sign In button		User should not be able to Login		Pass
5	Provide Invalid username or password	Username:Anish123 Password: Anish@0		Message for enter valid username or password displayed	
6	Provide Null username or Password	Username : null Password: null			
7	Click on Sign In button				

**Post-Condition:** User is validated with database based on their credentials and successfully login into his account and he can access the system. The Account session details are logged in database.

## Code package of Login

```
1 package test1;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 public class exm{
6 public static void main(String[] args) {
7 System.setProperty("webdriver.chrome.driver","C:\\\\Users\\\\susmi\\\\Downloads\\\\chromedriver_win32\\\\chromedriver.exe");
8 WebDriver driver=new ChromeDriver();
9
10 driver.get("http://localhost/smart_tourist_guiding_system/travel_website/login.php");
11 driver.findElement(By.id("username")).sendKeys("Anish");
12 driver.findElement(By.id("password")).sendKeys("Anish@123");
13 driver.findElement(By.id("signin")).click();
14 String actualUrl="http://localhost/smart_tourist_guiding_system/sidebar/index.php";
15 String expectedUrl= driver.getCurrentUrl();
16 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
17 System.out.println("Test passed");
18 } else {
19 System.out.println("Test failed");
20 }
21 }
22 }
23
24
```



## Test cases for Smart Tourist Guiding System Registration

<b>Project Name: Smart Tourist Guiding System</b>					
<b>Updation Test Case</b>					
<b>Test Case ID:</b> Registration		<b>Test Designed By:</b> Tom Joseph			
<b>Test Priority (Low/Medium/High):</b> High		<b>Test Designed Date:</b> 17-05-2022			
<b>Module Name:</b> Registration Screen		<b>Test Executed By:</b> Ms.Nimmy Francis			
<b>Test Title:</b> User Registration Details		<b>Test Execution Date:</b> 18-07-2022			
<b>Description:</b> Register to system and Registration is completed then login , if someerror occurs, test will fail					
<b>Pre-Condition:</b> User has valid username and password					
<b>Step</b>	<b>Test Step</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>
1	Navigation to Register Page		Register Page should be displayed	Registration page	Pass
2	Provide Valid Registration details	User Name: susmins	User should be able to Register	User registration Completed after go to the login page	Fail
3					
4	Click on Register button				
5	Provide profile details	Input profile details	User will be redirected to Login page	User will be redirected to Login page	Fail
7	Click on register button				
8	Provide invalid information	Input invalid profile details.	User will be stay in register page	User will be stay on that page showing error message	Pass
9	Click on register button				

**Post-Condition:** User can register into system and after the registration the user can successfully login into account.

## Code package for Registration

```

exp.java  exm.java  *reg.java ×
1 package test1;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 public class reg {
6 public static void main(String[] args) {
7 System.setProperty("webdriver.chrome.driver","C:\\Users\\susmi\\OneDrive\\Documents\\chromedriver.exe");
8 WebDriver driver=new ChromeDriver();
9 driver.get("http://localhost/Smart_Tourist_Guiding_System/travel website/registration.php");
10 driver.findElement(By.id("firstname")).sendKeys("Susmin");
11 driver.findElement(By.id("lastname")).sendKeys("Radhhh");
12 driver.findElement(By.id("emails")).sendKeys("Ancyanu@gmail.com");
13 driver.findElement(By.id("state")).sendKeys("kerala");
14 driver.findElement(By.id("pincode")).sendKeys("654321");
15 driver.findElement(By.id("dobs")).sendKeys("11/04/2000");
16 driver.findElement(By.id("phone")).sendKeys("9087563456");
17 driver.findElement(By.id("username")).sendKeys("susmins");
18 driver.findElement(By.id("password")).sendKeys("susmin@123");
19 driver.findElement(By.id("cpassword")).sendKeys("susmin@123");
20 driver.findElement(By.id("gender")).sendKeys("male");
21 driver.findElement(By.id("addresses")).sendKeys("test");
22 driver.findElement(By.id("register")).click();
23 String actualUrl="http://localhost/Smart_Tourist_Guiding_System/travel website/login.php";
24 String expectedUrl= driver.getCurrentUrl();
25 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
26 System.out.println("Test passed");
27 } else {
28 System.out.println("Test failed");
29 }
30 }
31 }
32

```

```

Console ×
<terminated> exm [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (18-Jul-2022, 8:45:28 pm - 8:45:55 pm) [pid: 27968]
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 8:45:30 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 8:45:30 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test failed

```

## **CHAPTER 6**

### **IMPLEMENTATION**

## 6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining the users' trust that the new system will function, be efficient, and accurate can be the most important step in creating a successful new system. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

At this point, the user department is responsible for the majority of the workload, the most disruption, and the most influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

The entire process of moving from the old system to the new one is referred to as implementation. The new system can be completely different, replace an existing human or automated system, or just be improved. The correct implementation of a dependable system that meets organisational requirements is required. System implementation describes the process of putting the created system into use. This includes every procedure needed to change from the old to the new system. The system may only be placed into operation after comprehensive testing and if it is established that it is working in compliance with the requirements. The system's staff do a viability assessment. The complexity of the system being implemented will affect how much work is required for system analysis and design in order to implement the three essential elements of education and training, system testing, and changeover.

The following tasks are included in the implementation state:

- Careful planning.
- Examination of the system and its limitations.
- Design of methods to achieve the changeover.

## 6.2 IMPLEMENTATION PROCEDURES

Software implementation describes the whole setup of the package in the intended environment, as well as the system's usability and fulfilment of the applications it was designed to support. It's common for someone who won't use the programme to

---

commission the development effort. Early scepticism of the programme is common, but we must be careful to prevent more scepticism by observing the following:

- The active user must be aware of the benefits of using the new system.
- Their faith in the software is increased.
- The user receives the appropriate instruction so that he feels confident using the application.

The user must be informed that in order to get the findings, the server programme must be operating on the server before they can examine the system. If the server object is not active and working on the server, the real procedure will not take place.

### **6.2.1 User Training**

To prepare the user to test and alter the system is the goal of user training. The participants must have confidence in their capacity to advance the objective and reap the rewards of the computer-based system. As systems get more complex, training becomes increasingly important. Through user training, the user learns how to input data, respond to error messages, query the database, call up routines that will create reports, and carry out other crucial activities.

### **6.2.2 Training on the Application Software**

The user must first obtain the fundamental training in computer literacy, following which they must be taught how to operate the new application software. In addition to how the screens function, what sort of help is shown on them, what types of errors are created while entering data, how each input is checked, and how to update the date that was entered, this will explain the core concepts of how to use the new system. The knowledge needed by the specific user or group to run the system or a particular component of the system should therefore be covered throughout the program's training on the application. The user group and hierarchy level may have an impact on how this training is delivered.

### **6.2.3 System Maintenance**

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software

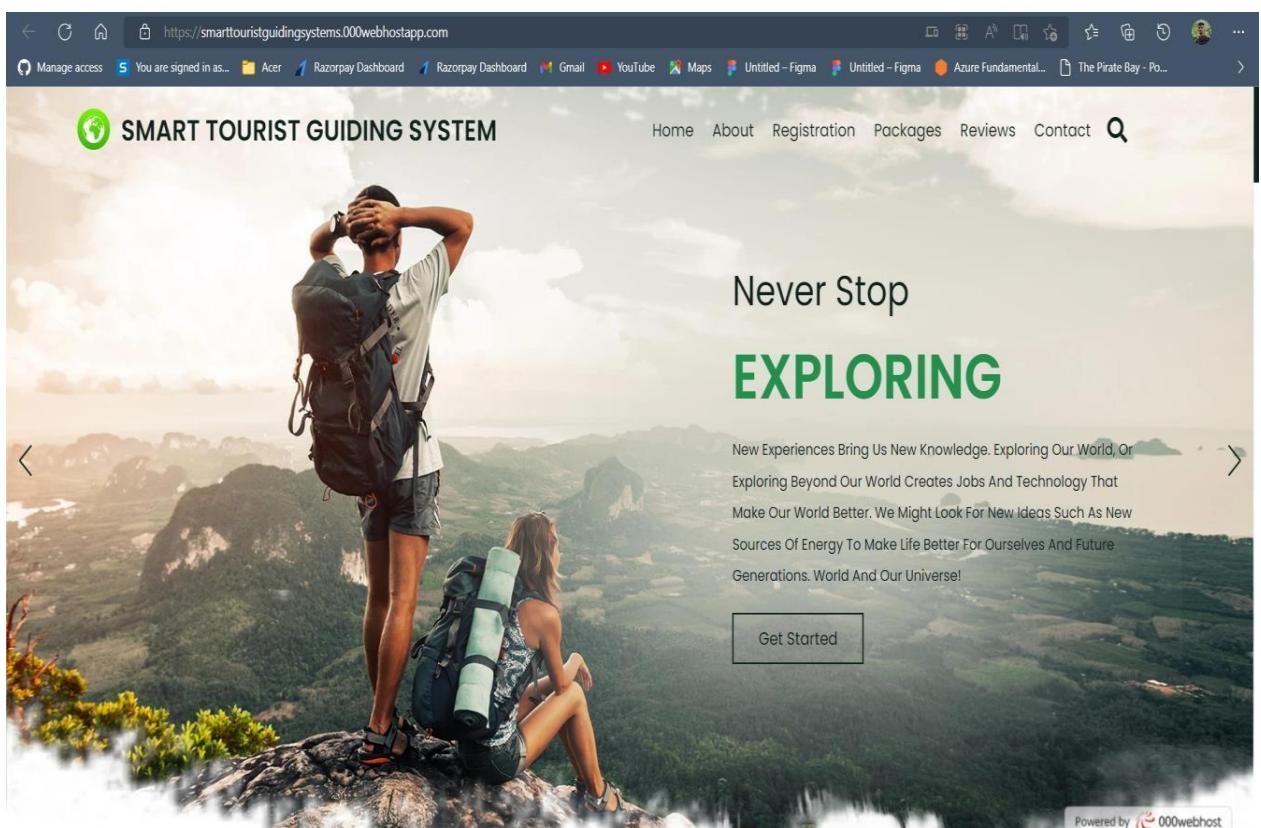
---

maintenance involves much more than just "Finding Mistakes."

#### 6.2.4 Hosting The System

A web hosting service is a sort of Internet hosting service that makes websites for customers visible on the World Wide Web and provides the resources needed for them to construct and manage a site. Web hosts are businesses that provide web hosting services. The 000webhost web host serves as the home for our system. A web hosting firm called 000webhost offers hosting space and domain name services.

The URL for our Smart Touirst Guiding System : Smart Touirst Guiding System 000webhostapp.com



## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## 7.1 CONCLUSION

The current system working technology is old fashioned and it is not helping efficiently for any travellers. The proposed system introduces for tourists that very helpful for their travels. They tourist can find suitable travel packages with customizable option which fit on their budget. Our system provides best travel packages with different varieties. So tourists enjoy the journey with great pleasure. In our system we introduce the travel guide services for tourists. So the tourists can book local travel guides for any guidance and help. This is also very helpful for travel guides because they can earn an income from our system. Also our system has local transportation facilities for tourists. So the tourists did not worry about local transportation in the tourist place. The cab booking charges are calculated the basis for how much kilometers are they travel and which type of cab are booked and the cab charge is automatically calculated in our system. so It is very convenient for tourists and cab drivers. The payment is directly transferred to the driver account. So cab drivers also get an chance to earn money from our system. So our proposed system is better from all dimension. Most importantly tourist can explore varieties of travel destination from our website.

## 7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Tourist can able to do advanced search options for finding their dream tourist destination.
- Both Tourist and Guide can able to add complaints and feedbacks etc.
- Admin can control all activities in this project.
- Tourist can find that regularly updated tourist packages.
- Data security can be enhanced.
- Genuine Details of each destination.
- Payment procedures are done through online

## **CHAPTER 8**

### **BIBLIOGRAPHY**

**REFERENCES:**

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

- [www.w3schools.com](http://www.w3schools.com)
- [www.jquery.com](http://www.jquery.com)
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### Login.php

```
<?php
include 'connection.php';
session_start();
?>
<!DOCTYPE html>
<!-- Created By CodingNepal -->
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Transparent Login Form HTML CSS</title>
<link rel="stylesheet" href="css/login.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
<!-- <script type="text/javascript">
    function preventBack() {
        window.history.forward();
    }
    setTimeout("preventBack()", 0);
    window.onunload = function() { null};</script> -->
</head>
<body>
<div class="bg-img">
    <div class="content">
        <a href="index.html">
            <span class="img-circle" width="40"; ></span>
        <header>Login Form</header>
        <form action="#" method="POST" enctype="multipart/form-data">
            <div class="field">
                <span class="fa fa-user"></span>
                <input type="text" required name="uname" placeholder="Enter Your Username" required>
            </div>
            <div class="field space">
                <span class="fa fa-lock"></span>
                <input type="password" class="pass-key" name="pass" placeholder="Enter Your Password" required>
            </div>
            <div class="pass">
                <a href="forgot.php">Forgot Password?</a>
            </div>
            <div class="field">
                <input type="submit" name="login" value="LOGIN">
            </div>
        </form>
        <div class="login">
    </div>
```

```

<div class="links">
    <div class="facebook">
        <a href="index.php" style="text-decoration:none;font-size: 20px;color: white;">Home</a>
    </div>
    <div class="instagram">
        <a href="registration.php" style="text-decoration:none;font-size: 20px;color: white;">Signup Now</a>
    </div>
    <table style="border-collapse: collapse; border: 1px solid black; width: 100px;"></table>
</div>

</div>
</div>

</body>
</html>

<?php
if(isset($_POST["login"])){
    $uname=$_POST["uname"];
    $pass=$_POST["pass"];

    $sql="select * from tbl_login where username='$uname' and password='$pass' and status in ('1', '0') and role in ('tourist', 'guide');";
    $result=mysqli_query($con,$sql);
    $count=mysqli_num_rows($result);

    while($row=mysqli_fetch_array($result))
    {
        $userids=$row['login_id'];
        $rol=$row['role'];
    }

    if($count>0 && $rol == "tourist")
    {
        $_SESSION['login_id'] = $userids;
        header("location:../sidebar/index.php");
    }
    else if($count>0 && $rol == "guide")
    {
        $_SESSION['login_id'] = $userids;
        header("location:../guide/index.php");
    }
    else
    {
        ?>
        <script>
            alert("invalid username or password");
        </script>
    }
}
else
{
    ?>
    <script>
        alert("invalid username or password");
    </script>
}

```

```
<?php  
}  
  
mysqli_close($con);  
  
}
```

### Tourist Registration.php

```
<?php  
include 'connection.php';  
ob_start();  
?>  
  
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title></title>  
    <style type="text/css">  
        *{  
            margin: 0;  
            padding: 0;  
        }  
        body{  
            background-image: url('images/home-bg-2.jpg');  
            background-position: center;  
            background-size: cover;  
            font-family: sans-serif;  
            margin-top: 40px;  
        }  
        .regform{  
            width: 800px;  
            background-color: rgb(0, 0, 0, 0.5);  
            margin: auto;  
            color: #FFFFFF;  
            padding: 10px 0px 10px 0px;  
            text-align: center;  
            border-radius: 15px 15px 0px 0px;  
            font-size: 20px;  
        }  
    </style>
```

```
        }
    .main{
        background-color: rgb(0, 0, 0,0.4);
        width: 800px;
        margin: auto;
    }

    form{
        padding: 10px;
    }
    #name{
        width: 100%;
        height: 100px;
    }
    .name{
        margin-left:25px;
        margin-top: 30px;
        width: 125px;
        color: white;
        font-size: 18px;
        font-weight:700;
    }

    }
    .firstname{
        position: relative;
        left: 150px;
        top: -37px;
        line-height: 40px;
        border-radius: 6px;
        padding: 0 22px;
        font-size: 16px ;
    }

    }
    .lastname{
        position: relative;
        left: 417px;
        top: -80px;
        width: 210px;
        line-height: 40px;
        border-radius: 6px;
        padding: 0 22px;
        font-size: 16px ;
        color: #555;
    }

    }
    .firstlabel{
        position: relative;
        color: #E5E5E5;
        text-transform: capitalize;
        font-size:14px;
        left: 203px;
        top:-45px;
    }
}
```

```
.lastlabel{  
    position: relative;  
    color: #E5E5E5;  
    text-transform: capitalize;  
    font-size: 14px;  
    left: 175px;  
    top: -45px;  
}  
.dob{  
    position: relative;  
    left: 150px;  
    top: -35px;  
    line-height: 40px;  
    border-radius: 6px;  
    padding: 0 22px;  
    width: 190px;  
    font-size: 16px ;  
    color: #555;  
}  
  
.password{  
    position: relative;  
    left: 405px;  
    top: -82px;  
    width: 190px;  
    line-height: 40px;  
    border-radius: 6px;  
    padding: 0 22px;  
    font-size: 16px ;  
    color: #555;  
}  
.conpassword{  
    position: relative;  
    left: 150px;  
    top: -37px;  
    line-height: 40px;  
    width: 550px;  
    border-radius: 6px;  
    padding: 0 22px;  
    font-size: 16px ;  
    color: #555;  
}  
.username{  
    position: relative;  
    left: 150px;  
    top: -37px;  
    width: 190px;  
    line-height: 40px;  
    border-radius: 6px;  
    padding: 0 22px;  
    font-size: 16px ;  
}  
.phonenumposition{  
    position: relative;  
    left: 435px;
```

---

top: -80px;

```
line-height: 40px;
border-radius: 6px;
padding: 0 22px;
width: 190px;
font-size: 16px ;
color: #555;
}
.pincodenewposition1{
position: relative;
left: 180px;
top: -33px;
line-height: 40px;
border-radius: 6px;
padding: 0 22px;
width: 190px;
font-size: 16px ;
color: #555;
}
.usernamelabel{
position: relative;
color: #E5E5E5;
text-transform: capitalize;
left: 410px;
top:-80px;
color: white;
font-size: 18px;
font-weight:700;
}
.phonelabel{
position: relative;
color: #E5E5E5;
text-transform: capitalize;
left: 410px;
top:-80px;
color: white;
font-size: 18px;
font-weight:700;
}
.picodelabel{
position: relative;
color: #E5E5E5;
text-transform: capitalize;
left: 170px;
top:-35px;
color: white;
font-size: 18px;
font-weight:700;
}
.addresslabel{
position: relative;
color: #E5E5E5;
text-transform: capitalize;
left: -140px;
top:70px;
color: white;
```

```
        font-size: 18px;
        font-weight:700;
    }
.passwordlabel{
    position: relative;
    color: #E5E5E5;
    text-transform: capitalize;
    left: 410px;
    top:70px;
    color: white;
    font-size: 18px;
    font-weight:700;
}
.address{
    position: relative;
    left: 150px;
    top: -37px;
    line-height: 40px;
    width: 550px;
    border-radius: 6px;
    padding: 0 22px;
    font-size: 16px ;
    color: #555;
}
.texte{
    position: relative;
    left: 150px;
    top: 35px;
    line-height: 40px;
    width: 550px;
    height: 120px;
    border-radius: 6px;
    padding: 0 22px;
    font-size: 16px ;
    color: #555;
}
.email{
    position: relative;
    left: 150px;
    top: -37px;
    line-height: 40px;
    width: 550px;
    border-radius: 6px;
    padding: 0 22px;
    font-size: 16px ;
    color: #555;
}
.phonenumber{
    position: relative;
    left: 150px;
    top: -37px;
    line-height: 40px;
    width: 190px;
    border-radius: 6px;
```

padding: 0 22px;

```
font-size: 16px ;
color: #555;
}

.genderlabel{
position: relative;
color: white;
text-transform: capitalize;
left: 20px;
top:19px;
font-size: 18px;
font-weight:700;

}

.linker{
margin-left: 340px;
margin-top:70px;
margin-bottom: 37px;
font-weight: 15px;
font-size: 19px;

}

a:hover {
color: aqua;
}

.radio{
display: inline-block;
font-size: 20px;
color: white;
position: relative;
color: #E5E5E5;
text-transform: capitalize;
left: 200px;
top:-7px;
font-size: 18px;
font-weight:700;
}

.radio input{
width: 20px;
height: 20px;
border-radius:50%;
cursor: pointer;
outline: none;

}

button{
background-color: #3BAF9F;
```

d  
i  
s  
p  
l  
a  
y  
:

b  
l  
o  
c  
k  
;

```

        font-size: 20px;
        margin: 20px 0px 0px 20px;
        text-align: center;
        border-radius: 12px;
        border: 2px solid #366473;
        padding: 14px 110px;
        outline: none;
        color: white;
        cursor: pointer;
        transition: 0.25px;

    }

    button:hover{
        background-color: #5390F5;
    }

.button1{
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
    background-color: #3333ff;
    border-radius: 12px;
}

}

</style>
<script type = "text/javascript">

function validate() {
    var password1=document.myForm.pass.value;
    var password2=document.myForm.conpassword.value;
    var phones=document.myForm.phone.value;
    var firname=document.myForm.firstname.value;
    var lasname=document.myForm.lastname.value;
    var ema=document.myForm.emails.value;
    var st=document.myForm.state.value;
    var pin=document.myForm.pincode.value;
    var user=document.myForm.username.value;
    var gen=document.myForm.gender.value;
    var addre=document.myForm.addresses.value;

    var pwd_expression = /^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{7,15}$/;
    var letters = /^[A-Za-z]+$/;
    var filter = /(^([a-zA-Z0-9_-])+@(([a-zA-Z0-9-]+\.)+)+([a-zA-Z0-9]{2,4})+$)/;
}

```

```
var phoneno = /\d{10}/;
var a = /(^d{6})/;

if(firname=="")
{
    alert('Please enter your name');
    return false;
}
else if(!letters.test(firname))
{
    alert('FirstName field required only alphabet characters');
    return false;
}
else if(lasname=="")
{
    alert('Please enter your lastname');
    return false;
}
else if(!letters.test(lasname))
{
    alert('LastName field required only alphabet characters');
    return false;
}

else if(ema=="")
{
    alert('Please enter your user email id');
    return false;
}

else if (!filter.test(ema))
{
    alert('Invalid email');
    return false;
}
else if(st=="")
{
    alert('Please enter your user State');
    return false;
}
else if(pin=="")
{
    alert('Please enter your Pincode');
    return false;
}
else if(!a.test(pin))
{
    alert('Please enter your Correct pincode');
    return false;
}
else if(phones=="")
{
    alert('Please enter the Phone.');
    return false;
}
```

```
        }
        else if(!phoneno.test(phones))
        {
            alert('Invalid Phoneno');
            return false;
        }
        else if(user=="")
        {
            alert('Please enter the username.');
            return false;
        }

        else if(password1=="")
        {
            alert('Please enter Password');
            return false;
        }
        else if(!pwd_expression.test(password1))
        {
            alert ('7 to 15 characters which contain at least one numeric digit and a special character are required in Password filed');
            return false;
        }
        else if(password2=="")
        {
            alert('Enter Confirm Password');
            return false;
        }

        else if(password1 != password2)
        {
            alert ('Password not Matched');
            return false;
        }

        else if(gen=="")
        {
            alert('Please enter the gender');
            return false;
        }
        else if(addr=="")
        {
            alert('Please enter your address');
            return false;
        }

        else{
            alert('Registration Sucessfully');
        }

        return( true );
    }
```

```

function getAge() {
    var dateString=document.myForm.dobs.value;
    //var dateString = document.getElementById("date").value;
    if(dateString!="")
    {
        var today = new Date();
        var birthDate = new Date(dateString);
        var age = today.getFullYear() - birthDate.getFullYear();
        var m = today.getMonth() - birthDate.getMonth();
        var da = today.getDate() - birthDate.getDate();
        if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
            age--;
        }
        if(m<0){
            m +=12;
        }
        if(da<0){
            da +=30;
        }
    }

    if(age < 18 || age > 100)
    {
        alert("Age "+age+" is restrict Allowed only for more than 18 years old");
        document.myForm.dobs.value="";
    }
    else {

        alert("Age "+age+" is allowed");
    }
} else {
//alert("please provide your date of birth");
}
}

</script>
<!-- <input type="text" id="date" value="1987/08/31" onblur="getAge()"> -->

</head>
<body>

<div class="regform"><a href="index.php">
    <div class="main">
        <form name = "myForm" action="registration.php" onsubmit = "return(validate());" method="POST" enctype="multipart/form-data">
            <div id="name">
                <h2 class="name">Name</h2>
                <input class="firstname" type="text" name="firstname"><br>
                <label class="firstlabel">first name</label>

```

```

<input class="lastname" type="text" name="lastname">
<label class="lastlabel">last name</label>
</div>

<h2 class="name">Email</h2>
<input class="email" type="email" name="emails">

<div id="name">
    <h2 class="name">State</h2>
    <select class="dob" name="state" style="height: 43px; width: 237px;">
        <option value="">Select State</option>
        <option value="kerala">Kerala</option>
        <option value="tamilnadu">Tamil Nadu</option>
        <option value="goa">Goa</option>
        <option value="karnataka">Karnataka</option>
        <option value="andrapradesh">Andra Pradesh</option>
    </select>
    <label class="picodelabel">Pincode :</label>
    <input class="pincodenewpostion1" type="text" name="pincode" placeholder="Pincode..."/>
</div>

<div id="name">
    <h2 class="name">Dob</h2>
    <input class="dob" type="date" name="dobs" onblur="getAge()"><br>
    <label class="phonelabel">Moble :</label>
    <input class="phonenewpostion" type="text" name="phone" placeholder="Mobile..."/>
</div>

<div id="name">
    <h2 class="name">Username</h2>
    <input class="username" type="text" name="username"><br>
    <label class="phonelabel">Password :</label>
    <input class="password" type="password" name="pass" placeholder="Enter password"/>
</div>
    <h2 class="name">Con.Pass</h2>
    <input class="conpassword" type="password" name="conpassword" placeholder="confirm password"/>

<h2 class="genderlabel">Your Gender?</h2>
<label class="radio">
    <input type="radio" name="gender" value="male" selected> Male
    <input type="radio" name="gender" value="female" class="radio2"> Female
</label>
<label class="addresslabel">Address :</label>
<input type="text" name="addresses" class="texte">

```

---

```
<h2 class="linker" ><a href="login.php" style="text-decoration:  
none;">Already Login...?</a></h2>
```

```
<center><button type="submit" name="click">Register</button></center>
<center><a href="index.php"><input type="button" name="Home"
class="button1" value="Home" style="margin: 20px 600px 0px 40px;"></center></a>
</div>

</body>
</html>

<?php

if(isset($_POST["click"]))

{

$fname=$_POST["firstname"];

$lname=$_POST["lastname"];
$add=$_POST["addresses"];
$eml=$_POST["emails"];
$statt=$_POST["state"];
$pincodes=$_POST["pincode"];
$dob=$_POST["dobs"];
$user=$_POST["username"];
$pass=$_POST["pass"];
$phone=$_POST["phone"];
$gend=$_POST["gender"];
$roll="tourist";

$sql = "Select * from tbl_login where username='$user'";

$result12 = mysqli_query($con, $sql);

$num = mysqli_num_rows($result12);

if($num == 0) {

$req = "insert into tbl_login(username,password,role) values ('$user','$pass','$roll')";

mysqli_query($con,$req);

$query="select * from tbl_login where username='$user' ";
//query = "select * from tbl_login";
$result = mysqli_query($con , $query);

$row=mysqli_fetch_array($result);

$usern=$row['login_id'];

$sql = "INSERT INTO
```

```
tbl_registration(firstname,lastname,address,emails,state,pincode,dobs,phone,gender,login_id)
VALUES ('$fname',' $lname','$add','$eml','$statt','$pincodes','$dob','$phone','$gend',
'$usern');

    mysqli_query($con,$sql);
    $sql="INSERT INTO tbl_regcpics(prof_img,pro_proof,proof_type,login_id)
VALUES ('personicon.jpg','Not Uploaded','Not Uploaded','$usern')";
    mysqli_query($con, $sql);

    header('location:index.php');

}

else
{
    echo "<script>alert('username not available')</script>";
}
ob_end_flush();

}

?>
```

## 9.2 Screen Shots

### Home page



### User: Admin

#### Admin Dashboard

The image shows the Admin Dashboard of the Smart Tourist Guiding System. On the left is a sidebar with a profile picture of 'Andrar Son' (Admin) and a navigation menu with items: Home, Users, Packages, Booking, Payments, Feedbacks, and Contacts. The main content area has a pink header bar with 'Home / Dashboard'. Below this are four cards: 'Tourists' (5 Total Tourist), 'Feedbacks' (2 Total Feedbacks), 'Contacts' (3 Total Contacts), and 'Places' (2 Total Places). Another row of cards shows 'Guides' (3 Total Guides) and 'Packages' (6 Total Packages). At the bottom of the dashboard is a copyright notice: 'Copyright © 2018 Colorlib All rights reserved. Template by Colorlib.'

## Create Package Page



### Create Package

Package Name

Package Type

Best Seasons

Package Features

Package Location

Package Duration

Package Price

Package Details

Package Image  No file chosen

**Create Package**

## Manage Package Page

The screenshot shows a web-based administrative interface for managing tourist packages. The left sidebar includes links for Home, Users, Packages, Places, Booking, Feedbacks, and Contacts. The main content area displays a table titled "Lists of Tourist Package" with the following data:

PACKAGE NAME	PACKAGE TYPE	SEASON	PACKAGE FEATURES	PACKAGE LOCATIONS	DURATION	PRICE	DETAILS	IMAGE	VIEW
Munnar Package	adventure_Tour		Include almost all activities.	Munnar	1	10000	We have 3 night and 4 days in munnar.		<a href="#">View</a>
Kumaly Package	Single_Tour	rainy	Free Food and Free Wifi	Kumaly	1	2000	Kumaly is a wonderful place for solo trip		<a href="#">View</a>
Marayoor Family Package	Family_Tour	spring	Free Transportation Charges	Marayour	2	6000	In our package up to four family members can travel in this journey		<a href="#">View</a>
Malampuzha Package	adventure_Tour	summer	Free pickups	Palakkad,Malampuzha	3	12000	A Adventurous trip on Malampuzha Dam and Forest		<a href="#">View</a>
SilentValley Tour	adventure_Tour	rainy	Free tend facility	Palakkad Silent Vally	2	5000	You can spend the time on Silent Valley Forest.		<a href="#">View</a>
Alappuzha Package	Group_Tour	summer	Free Food and Free Wifi	Alappuzha	2	10000	Your Group can spend the time on Alappuzha Backwaters.		<a href="#">View</a>

Showing 1 to 6 of 6 rows

## Manage Tourist Page

The screenshot shows a web-based administrative interface for managing tourist data. The left sidebar includes links for Home, Users, Packages, Places, Booking, Feedbacks, and Contacts. The main content area displays a table titled "Touists Data Lists" with the following data:

FIRSTNAME	LASTNAME	ADDRESS	E-MAILS	DATE OF BIRTH	USERNMAE	PHONE	GENDER	STATUS	VIEW
anurag	das	sample data	anurag@gmail.com	1979-01-19	anurag	7894561023	male	Not Checked	<a href="#">View</a>
anish	mathew	palm house	anish@gmail.com	1999-12-12	anish	7794563021	male	Not Checked	<a href="#">View</a>
bijesh	monson	rose ville	bijesh@gmail.com	1995-05-11	bijesh	9874758623	male	Not Checked	<a href="#">View</a>
merin	joseph	Kandachall house	tommanomaani@gmail.com	1998-07-07	merin	6078762083	female	Not Checked	<a href="#">View</a>
Tom	Joseph	Kandachall House	tomjoseph2022b@mca.ajce.in	1999-01-15	tom	8078352078	male	Not Checked	<a href="#">View</a>

Showing 1 to 5 of 5 rows

Copyright © Template Reserved For Smart Tourist Guiding System.

## Manage Guide Page

Guide Data Lists

FIRSTNAME	LASTNAME	ADDRESS	E-MAILS	DATE OF BIRTH	USERNMAE	PHONE	GENDER	STATUS	VIEW
dijo	vargeese	morche villes	dijo@gmail.com	1995-09-11	dijo	6598784511	male	Not Checked	<button>View</button>
vishnu	kumar	kumarvilasam	vishnu@gmail.com	1991-01-12	vishnu	9998854523	male	Not Checked	<button>View</button>
mukesh	babu	flowerville	mukesh@gmail.com	1995-04-04	mukesh	9956235815	male	Not Checked	<button>View</button>
reena	joseph	pinehouse	reena@gmail.com	1997-06-05	reena	8856235411	female	Not Checked	<button>View</button>
george	kurian	castellhouse	george@gmail.com	2000-04-03	george	7875964512	male	Not Checked	<button>View</button>

Showing 1 to 5 of 5 rows

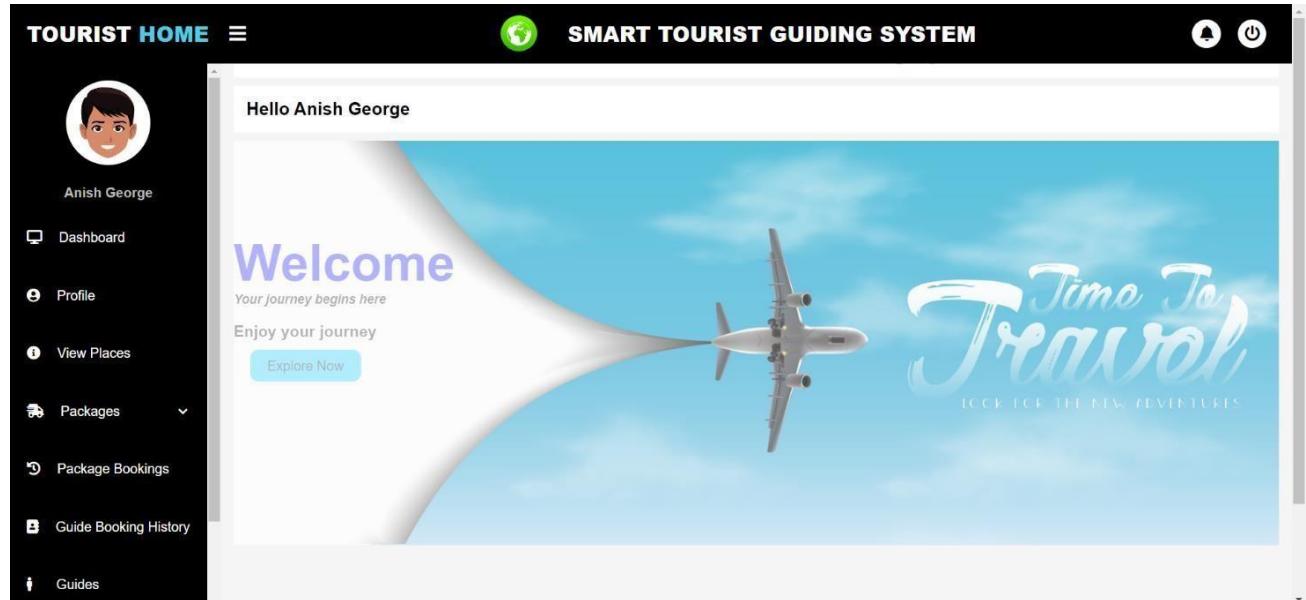
Copyright © Template Reserved For Smart Tourist Guiding System.

## Manage Tourist Bookings Page

Search

PACKAGE NAME	EMAIL	PHONE	BOOKING DATE START	BOOKING DATE END	NUMBER OF PERSON	PACKAGE TYPE	STATUS	ACTION	ACTION
Munnar Package	tommanomaani@gmail.com	6078762083	2022-07-02	2022-07-03	2	adventure_Tour	Rejected	<button>Accept</button>	<button>Reject</button>
Kumaly Package	tommanomaani@gmail.com	6078762083	2022-07-01	2022-07-02	1	Single_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Munnar Package	tommanomaani@gmail.com	6078762083	2022-07-20	2022-07-21	5	adventure_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Munnar Package	tommanomaani@gmail.com	6078762083	2022-07-20	2022-07-21	5	adventure_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Munnar Package	tommanomaani@gmail.com	6078762083	2022-07-13	2022-07-14	9	adventure_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Malampuzha Package	tommanomaani@gmail.com	6078762083	2022-07-28	2022-07-31	1	adventure_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Malampuzha Package	tommanomaani@gmail.com	6078762083	2022-07-28	2022-07-31	1	adventure_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>
Kumaly Package	tommanomaani@gmail.com	6078762083	2022-07-13	2022-07-14	1	Single_Tour	Not Checked	<button>Accept</button>	<button>Reject</button>

Showing 1 to 8 of 8 rows

**User: Tourist****Tourist Dashboard****Explore New Place for Tourists**

## Packages Views for Tourists

The screenshot shows the 'TOURIST HOME' interface. On the left sidebar, there is a user profile picture of Anish George and a list of navigation options: Dashboard, Profile, View Places, Packages (selected), Package Bookings, Guide Booking History, and Guides. The main content area is titled 'SMART TOURIST GUIDING SYSTEM' and 'Welcome To Smart Tourist Guiding System'. It displays a 'Single Tour Packages' section with a card for the 'Palakkad Package'. The card features a thumbnail image of a large, ornate building, a title 'Palakkad Package', and a list of details: Place: Palakkad, Trek Duration: 2, Best Season: spring, Package Type: Single\_Tour, and Package Price: 7000 INR. At the bottom of the card are two buttons: 'Book Now' (orange) and 'View Package' (grey).

## List of Package Booking

The screenshot shows the 'TOURIST HOME' interface. The left sidebar includes a user profile picture of Anish George and navigation options: Dashboard, Profile, View Places, Packages (selected), Package Bookings, Guide Booking History, and Guides. The main content area is titled 'SMART TOURIST GUIDING SYSTEM' and 'Welcome To Smart Tourist Guiding System'. It displays a 'Booking History' section with a table showing two entries:

Package	Phone	Book Date Start	Book Date End	No.of.Persons	Package Type	Status	Total Amount	Cutomization Amount	Payment Status	Action
Kumaly Package	8789654512	2022-07-12	2022-07-14	2	adventure_Tour	Not Checked	<b>10000</b>	20200	Paid	<button>Cancel</button>
Palakkad Package	8789654512	2022-07-23	2022-07-25	1	Single_Tour	Not Checked	<b>7000</b>	5200	Pending	<button>Pay</button>

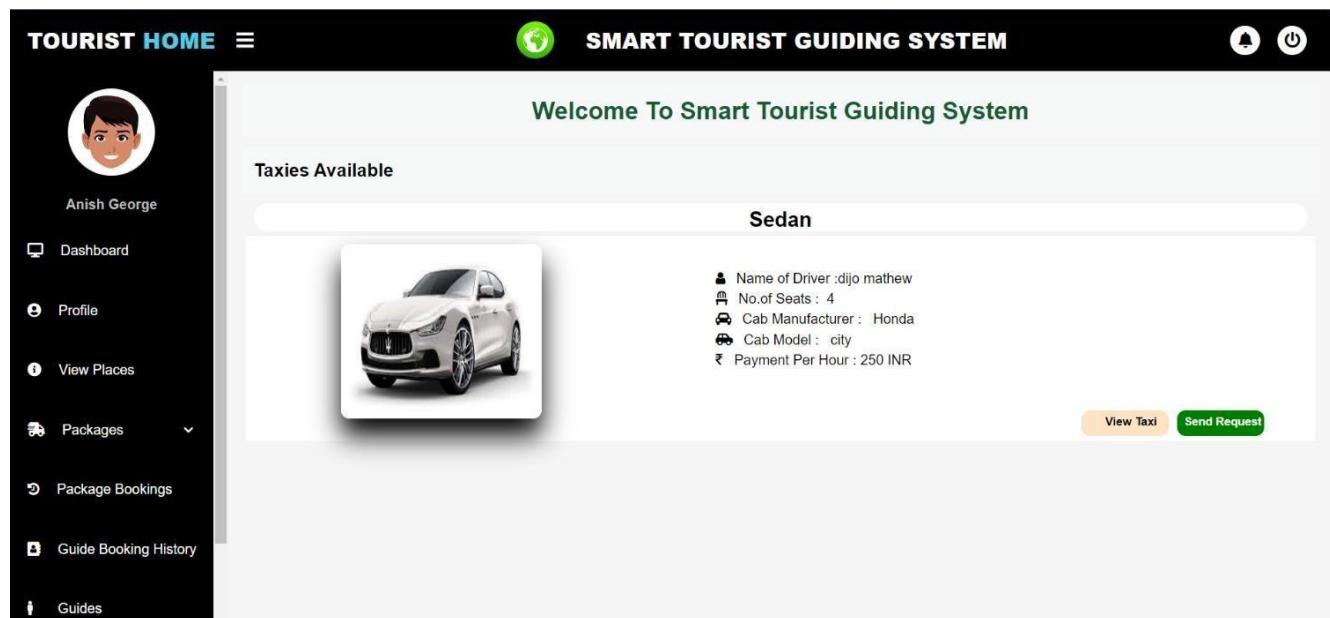
## List of Guide Booking

The screenshot shows the 'Guide Booking History' section of the system. On the left, there's a sidebar with a user profile picture of 'Anish George' and a list of navigation options: Dashboard, Profile, View Places, Packages, Package Bookings, Guide Booking History (which is currently selected), and Guides. The main content area has a header 'Welcome To Smart Tourist Guiding System' and a sub-header 'Guide Booking History'. Below these are two tables. The first table has columns: Booked Guide Name, Guide Phone, Guide Email, Book Date Start, Book Date End, Total Hours, Total Charge, Book Date, Payment Status, and Request Status. It contains one row for 'Anoop Chacko' with values: anoop@gmail.com, 9586231442, 2022-07-18, 2022-07-11, 168, 2016, 2022-07-11 12:34:30, Paid, and Accepted. The second table is partially visible below it.

## Guide View

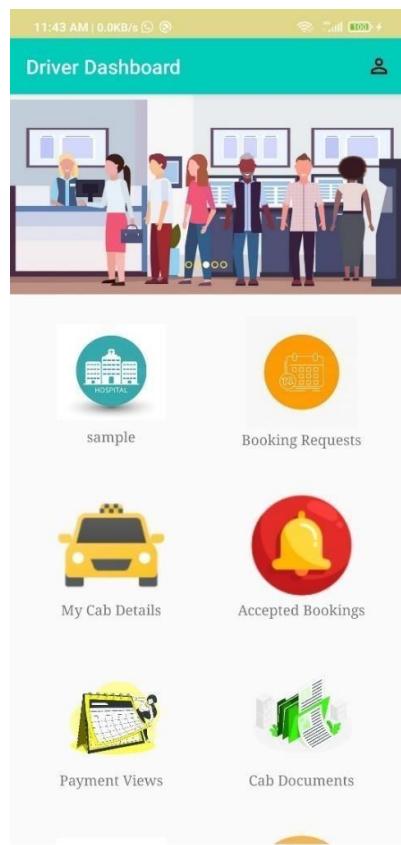
The screenshot shows the 'Guides Available' section. The sidebar on the left is identical to the previous screenshot. The main content area has a header 'Guides Available' and a sub-header 'Anoop Chacko'. It displays a circular profile picture of Anoop Chacko. To the right of the picture, there are details: Guiding Place :Idukki, Contact : 9586231442, Email : anoop@gmail.com, Language Known : Malayalam, French, Arabic, Italian, and Payment Per Hour : 12 INR. Below this information is a 5-star rating icon and a note 'Total Review:1'. At the bottom right are 'View Guide' and 'Send Request' buttons. Below this section is another one for 'Bretto John' with similar details, including a profile picture, contact info, language known, payment per hour, a 5-star rating, and review count. It also has 'View Guide' and 'Send Request' buttons.

## Cab View



## User: Driver

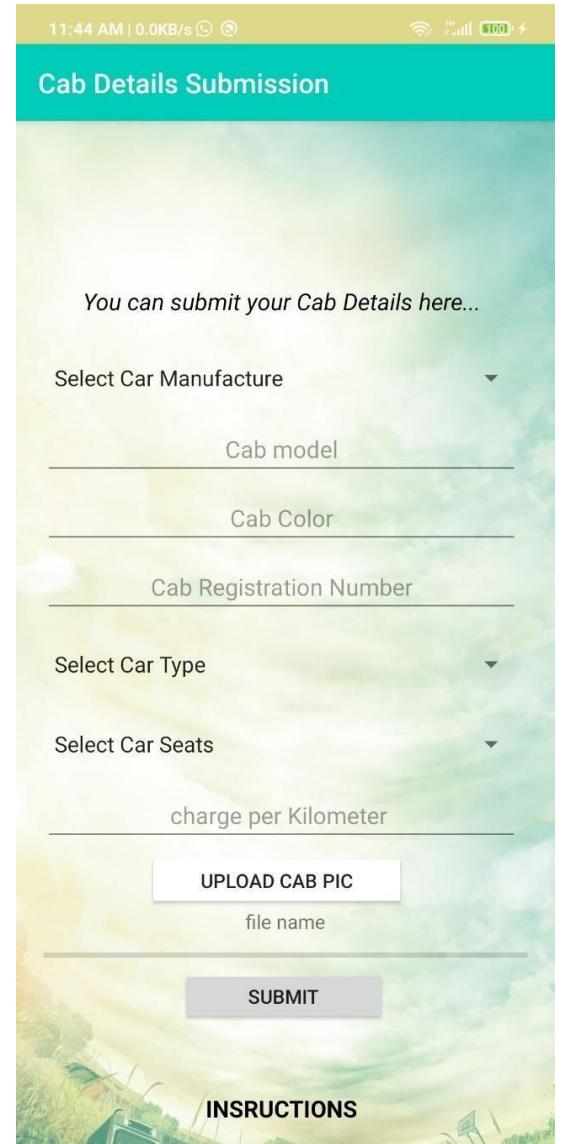
### Driver Dashboard



## Driver Accepted Bookings



## Car Details Submission



**Payment Completed Lists****Cab Document Submission**A screenshot of a mobile application interface titled "Cab Details Submission". At the top, it shows the time as 11:44 AM and the signal strength. Below the title, there is a message: "You can submit your assignments here...". There are three sections for "Select Car Document" with dropdown menus, each followed by an "Enter No Here" input field. Below these is a "UPLOAD LICENSE PHOTO" button with a placeholder "file name". At the bottom right is a "SUBMIT" button and the word "INSRUCIONS" (likely a typo for "INSTRUCTIONS").

## User: Guide

### Add Places

GUIDE HOME

SMART TOURIST GUIDING SYSTEM

Add Places

YOUR NAME \*

Anoop Chacko

EMAIL \*

anoop@gmail.com

PHONE \*

9586231442

NAME OF PLACE \*

PLACE SITUATED DISTRICT\*

Select District

DESCRIPTION OF THE PLACE \*

FIRST PHOTO OF PLACE \*

Choose File | No file chosen

SECOND PHOTO OF PLACE \*

Choose File | No file chosen

THIRD PHOTO OF PLACE \*

Choose File | No file chosen

FOUR PHOTO OF PLACE \*

Choose File | No file chosen

Submit

### Guide Bookings

TOURIST HOME

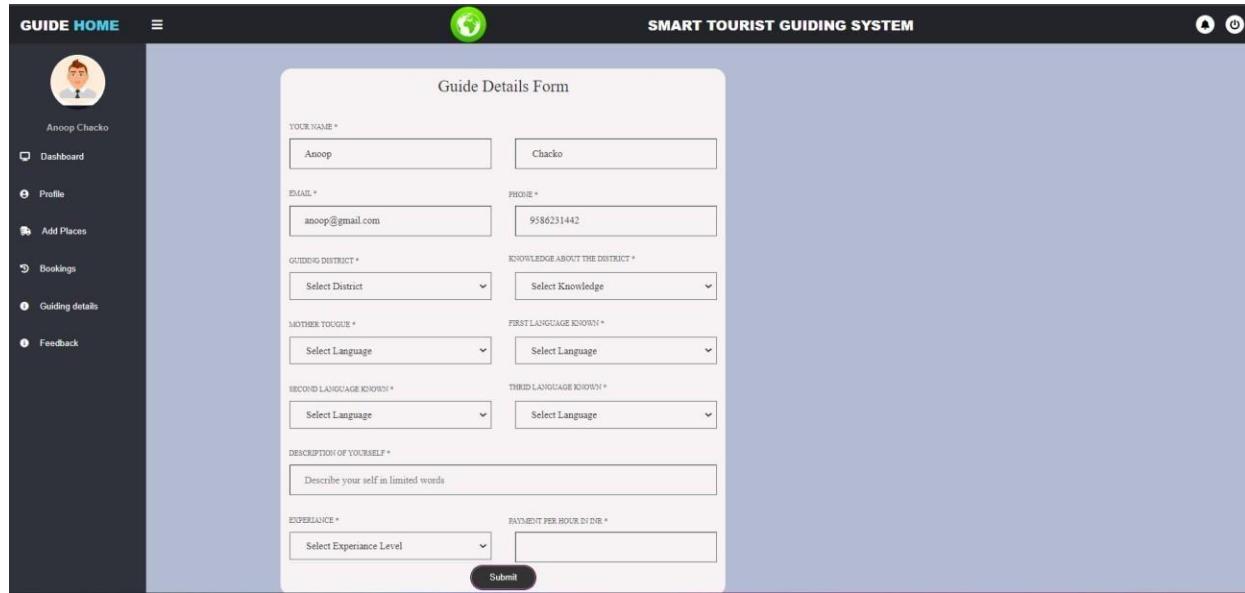
SMART TOURIST GUIDING SYSTEM

Welcome To Smart Tourist Guiding System

Booking History

Tourist Name	E-mail	Phone	Book Date Start	Book Date End	Booked Date	Request Status	Action
Anish George	tomjoseph11042000@gmail.com	8789654512	2022-07-18	2022-07-11	2022-07-11 12:34:30	Accepted	

## Guide Details



The screenshot shows the 'Guide Details' form within the 'SMART TOURIST GUIDING SYSTEM'. The form is titled 'Guide Details Form' and includes the following fields:

- YOUR NAME \*: Two input fields for First Name ('Anoop') and Last Name ('Chacko').
- EMAIL \*: Input field containing 'anoop@gmail.com'.
- PHONE \*: Input field containing '9586231442'.
- GUIDING DISTRICT \*: Two dropdown menus for 'Select District' and 'Select Knowledge'.
- MOTHER TONGUE \*: Two dropdown menus for 'Select Language'.
- SECOND LANGUAGE KNOWN \*: Two dropdown menus for 'Select Language'.
- THIRD LANGUAGE KNOWN \*: Two dropdown menus for 'Select Language'.
- DESCRIPTION OF YOURSELF \*: A text area with placeholder text 'Describe your self in limited words'.
- EXPERIENCE \*: A dropdown menu for 'Select Experience Level'.
- PAYMENT PER HOUR IN INR \*: An input field for payment amount.

A 'Submit' button is located at the bottom right of the form. The left sidebar shows the user profile 'Anoop Chacko' and navigation links: Dashboard, Profile, Add Places, Bookings, Guiding details, and Feedback.



| ●

|

| •

• |

| •

|

| ||

|



|



:



:



:



:



:



:



:



:



100

100  
100100  
100  
100100  
100

100



100



100



100



100







---



.....

|

|

|

|