ABSTRACT

The **Simple Ticket Booking System** has 2 modules which are the **Management Side** and **Public Side**. The **Management Module** is the side of the system where the management can manage the system information, schedules, and reservations. On this side, the admin user can update also the content pages that are shown on the website such as the 'About' and 'Contact' Information Contents. The **Public Module** serves as the website that can be accessed by visitors or possible passengers. The visitors can read the information about the station, list the schedules, and reserve their seats on their selected schedule. The passenger can submit multiple seat reservations at once.

The researchers conducted the capstone project entitled “Bus or Train Booking System” to help commuters reserve a seat on the train or bus via the this php application, this will also help the management by promoting innovation to their customers. Train or Bus Booking System was developed in PHP, MySQL and Bootstrap using the Rapid Application Development software development life cycle. The said project had undergone several testing and evaluation using standardized set of questionnaires. The evaluation was rated by the end-users (commuters and bus operator management) and IT experts, results showed that by using the said it system, it would make the transactions easier, faster and convenient to both the commuters and management. In the findings of the study, researchers conclude that the “Train and Bus Booking System” meets the requirements of the end-users and it is highly recommended for implementation as an alternative way of reserving a seat on a train and bus.

CHAPTER-1: INTRODUCTION

1.1 About the Project:

This Software project is aimed at automation of Ticket booking. Objective of the project is to develop customize software package for Ticket booking.

In our daily life we have to book ticket for Bus, movies, Flight and cricket match or foot boll match. When we do this task manually then it becomes very hard to manage the ticket booking. So are developing this system to manage booking of ticket automatically. In this Bus and Railway Ticket Booking project we develop the system that can help the user to book their ticket online and offline manner. In this project we take care of every service related to travelling and online ticket booking.

When you start your traveling tour you have to book ticket so we help you from the starting point to the end. we help you in booking your traveling ticket. Bus and Railway Ticket Booking System in php then when you reached your Destination we help you with booking details. you can also book more than one ticket from this project system.

1.1.1. Admin Panel:

1. Admin Login:
   Admin can login through login form.
2. Admin Profile:
   Admin can manage his own profile. Admin can also change his password
3. Booking history:
   Admin can manage the booking history
4. Registration:
   Admin can create User's profile
5. Manage the Registration:
   Admin can manage the all the User's Profile. Take a print out of all profiles and also delete the profile.
6. Forgot Password:
   Admin can also retrieve the password if admin forgot the password.

1.1.2. User Panel:

1. User Registration- User can register through user registration form
2. User Login- User can login through login form
3. Forgot Password-user can retrieve password through forgot password link
4. User Dashboard
5. User Profile—User can manage own profile
6. Book Ticket – User can book Tickets
7. Ticket Details- Booked Ticket Details
8. Change Password**-** User Can change own password

CHAPTER-2: LITERATURT SURVEY

2.1 EXISTING SYSTEM AND PROPOSED SYSTEM:

2.1.1 Existing System:

The existing system is very complex as every work is done manually. By using the present system, work is done manually. So, each and every work takes much time to complete. Whenever the users needs the information it is very difficult for the users to search for that particular train or bus or source and destination details and the information to be ordered. Every time we should search the records at the shelves. Initially, customers used to book tickets manually by queuing in lines at bus stations or go to travel agents who in turn used to book tickets for them. This was actually a tedious process and was leading to wastage of time. It also had issues like having incorrect names or other information used to book tickets.

2.1.2 Proposed System:

The present system has obvious problems, inhibiting growth and more usage of man power. The present system which has been proposed is very easy to work .The computerization of the every department in the Ticket booking system will reduce the work that is done manually. The man power is reduced to the maximum extent. The users at the railway registration office are registered within no time, because every time there is no need search for the particular opon in the stations .The details about the trains or bus and users are maintained without any complexity and all the calculations are made automatically by this system there is no need for the calculations. We can eliminate the drawbacks by developing an application which will allow customers to register themselves and book tickets, cancel tickets or postpone or prepone travel dates with feasibility. This actually is a welcome step for customers as they can access the application from anywhere and will also avoid wastage of time that was caused due to the drawbacks in the previous way of booking tickets manually.

2.2 FEASIBILITY STUDY:

After doing the T system, study and analyzing all the existing are required functionalities of the system. The next task is to do the feasibility study for the project. All projects are feasible given unlimited resource and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem.

- Economical feasibility: This very important aspect to be considered while developing a project. We decide the technology based on minimum possible cost factor.

- Technical feasibility: This included the study of function performance and constraints that may affect the ability to achieve an acceptable system.

- Operational feasibility: No doubt the propose system is fully GUI based that is very user friendly. A proper training has be conducted to let know the essence of the system to the users so that they feel comfortable with new system.

ECONOMIC FEASIBILITY:

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any additional hardware or software.

TECHNICAL FEASIBILITY:

· The technical issue usually raised during the feasibility stage of the investigation includes the following:

· Does the necessary technology exist to do what is suggested?

· Do the proposed equipments have the technical capacity to hold the data required to use the new system?

· Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

· Can the system be upgraded if developed?

SOME OF TOOLS ARE USED TO IMPLEMENT THE PROJECT:

1. Xampp Server
2. MySQL – DataBase
3. Eclipse IDE

XAMPP SERVER:

XAMPP is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver. It was created by Apache Friends, and the public can revise or modify its native source code. It includes MariaDB, Apache HTTP Server, and interpreters for PHP and Perl, among other computer languages. Because of XAMPP's simplicity of deployment, a developer can quickly and easily install a WAMP or LAMP stack on an operating system, with the added benefit that common add-in apps like WordPress and Joomla can also be loaded.

Need for a XAMPP

- XAMPP is simply a local host or server.
- This local server runs on your personal computer, whether it's a desktop or a laptop.
- It is used to test clients or websites before publishing them to a remote web server.
- On a local computer, the XAMPP server software provides a suitable environment for testing MYSQL, PHP, Apache, and Perl projects. Because most real-world web server deployments share the same components as XAMPP, moving from a local test server to a live server is straightforward.

MYSQL DATABASE:

MySQL is a database management system that is used to maintain relational databases. It is an open-source software backed by Oracle Corporation. This was originally founded by a Swedish company called MYSQL AB which was later acquired by sun microsystems and finally is with Oracle Corporation. As it is an open-source database system, the source code can be modified according to our needs. It also offers premium services if a commercial license is purchased from Oracle Corporation. MySQL is a scalable, fast, and reliable database management system which can run on any platform like Windows, Unix, Linux, etc., and can be installed on the desktop or any server machine.

It is also very to master compared to other database management systems existing in the market like Microsoft SQL Server, Oracle Database, etc. MySQL is most suitable for web applications. MySQL is an essential component of the LAMP stack, which includes Linux, Apache, MySQL, and PHP. LAMP is a platform for web development using Linux as the operating system, the webserver of apache, relational database management system of mysql and object-oriented scripting of PHP. There are many top websites using mysql. Apart from this, there are numerous corporations using mysql as their relational database management system. Few examples include Youtube, Facebook, Twitter, etc. MySQL works on a client-server model, the MySQL server being the core handling all commands.

ECLIPSE IDE:

The Eclipse is defined as platform for developing the computer-based applications using various programming language like JAVA, Python, C/C++, Ruby and many more. The Eclipse is IDE (Integrated development kit) and mainly JAVA based programming is done in this platform. There are several plug-ins and other additional plug-ins can be installed in the platform. The advanced client applications can be developed. The JDT is used for doing the programming in Eclipse IDE.

Importance of Eclipse IDE

Given below are the several importance factors that are mentioned:

1. Code insight or code competition

For any type of IDE platform, the identifying of function and keywords for any particular programming language is always a crucial part. In the Eclipse IDE the plugins are available for every programming type language so that it is capable of identifying the keywords very easily and helps to develop the applications in this platform.

2. Provide workspace

It provides the workspace in which user can bundle all the projects in single workspace. In that single workspace the source files, artifacts, images all can be stored in the workspace. The user has the complete functionality to select the name of workspace and manage the projects in single workspace.

3. Provide editors and views

It provides the editors and views for navigating in between IDE and change the content. These different views are known as perspective in eclipse IDE. For every particular group of data separate view is provided to user. Every view has its own hierarchical data and when user click on other view the data hierarchy is changed and get displayed of that particular view. For e.g. the project explorer view displays the list of all projects on which user is currently working. The user can access the files and projects in the project explorer view.

2.2.1 Usage:

PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server, taking PHPhnvnhbvhjb code as its input and creating web pages as output. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

2.2.2 Speed Optimization:

As with many scripting languages, PHP scripts are normally kept as human-readable source code, even on production web servers. In this case, PHP scripts will be compiled at runtime by the PHP engine, which increases their execution time. PHP scripts are ableto be compiled before runtime using PHkjhhhjP compilers as with other programming languages such as C (the language PHP and its extensions are written in). Code optimizers aim to reduce the computational complexity of the compiled code by reducing its size andmaking other changes that can reduce the execution time with the overall goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the Zend Optimizer PHP extension.

2.2.3 Security:

The National Vulnerability Database stores all vulnerabities found in computer software.The overall proportion of PHP-related vulnerabilities on the database amounted to: 12%in 2003, 20% in 2004, 28% in 2005, 43% in 2006, 36% in 2007, and 35% in 2008. Most of these PHP-related vulnerabilities can be exploited remotely: they allow hackers to stealor destroy data from data sources linked to the webserver (such as an SQL database), sendspam or contribute to DOS attacks using malware, which itself can be installed on the vulnerable servers.

2.2.4 Syntax:

```
<html>
<head>
<title>PHP  Test  </title>
</head>
<body>
<?php  echo  "<p>  Hello  World </p>";  ?>
</body></html>
```

Note : - Code in bold letters shows the PHP code embedded within HTML

PHP only parses code within its delimiters. Anything outside its delimiters is sent directlyto the output and is not parsed by PHP. The most common delimiters are <?php and ?>, which are open and close delimiters respectively. <script language=”php”> and </script> delimiters are also available. Short tags can be used to start PHP code, <? or <?= (whichis used to echo back a string or variable) and the tag to end PHP code, ?>. These tags arecommonly used, but like ASP-style tags (<% or <%= and %>), they are less portable as they can be disabled in the PHP configuration. For this reason, the use of short tags and ASP-style tags is discouraged. The purpose of these delimiters is to separate PHP code from non-PHP code, including HTML.

2.2.5 Datatypes:

PHP stores whole numbers in a platform-dependent range. This range is typically that of32-bit signed integers. Unsigned integers are converted to signed values in certainsituations; this behavior is different from other programming languages. Integer variablescan be assigned using decimal (positive and negative), octal, and hexadecimal notations

2.2.6 Functions:

PHP has hundreds of base functions and thousands more from extensions. These functionsare well documented on the PHP site, but unfortunately, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming.

2.2.7 Version 5.2 and Earlier:

Functions are not first-class functions and can only be referenced by their name—directly or dynamically by a variable containing the name of the function. User-defined functions can be created at any time without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero argument class constructor functions called withthe PHP new operator, where parentheses are optional.

2.2.8 Version 5.3 and Newer:

PHP gained support for first-class functions and closures. True anonymous functions are supported

function getAdder($x) using the following syntax :

```
function   getAdder($x)
{
return function ($y) use ($x)

{ return $x + $y; };}
$adder   =   getAdder(8);

echo   $adder(2);   //   prints   “10”
```

Here, getAdder() function creates a closure using parameter $x (keyword “use” forces getting variable from context), which takes additional argument $y and returns it to the caller. Such a function can be stored, given as the parameter to another functions, etc. Formore details see Lambda functions and closures RFC

2.2.9 MySql:

What is a database?

Quite simply, it’s an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities toadd, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.
MySQL is a multithreaded,multi-user SQL database management system(DBMS). The basic program runs as a server providing multi-user access to a number of databases.Originally financed in a similar fashion to the JBoss model, MySQL was ownedand sponsored by a single for-profit firm, the Swedish company MySQLAB now a subsidiary of Sun Microsystem , which holds the copyright to most of the codebase. Theproject’s source code is available under terms of the GNU General Public Licence, as well as under a variety of proprietory agreements.

MySQL is a database.The data in MySQL is stored in database objects called tables.A table is a collections of related data entries and it consists of columns and rows.Databasesare useful when storing information categorically. A company may have a database withthe following tables:

“Employees”, “Products”, “Customers” and “Orders”.

2.3,11 MySql Functions:

- mysqli_affected_rows — Get number of affected rows in previous MySQL operation
- mysqli_change_user — Change logged in user of the active connection
- mysqli_client_encoding — Returns the name of the character setmysqli_close — Close

MySQL connection

- mysqli_connect — Open a connection to a

- mysqli_create_db — Create a database
- mysqli_data_seek — Move internal result pointer
- mysqli_db_name — Get result data
- mysqli_db_query — Send a MySQL query

2.3.12 PHP MyAdmin:

phpMyAdmin is an open source tool written in PHP intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL.Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage users and permissions, and manage keys on fields. while youstill have the ability to directly execute any SQL statement. phpMyAdmin can manage a whole MySQL server (needs a super-user) as well as a single database. To accomplish the latter you’ll need a properly set up MySQL user who can read/write only the desired database. It’s up to you to look up the appropriate part in the MySQL manual.

phpMyAdmin can:

- browse and drop databases, tables, views, fields and indexes
- create, copy, drop, rename and alter databases, tables, fields and indexes
- maintenance server, databases and tables, with proposals on server configuration
- execute, edit and bookmark any SQL-statement, even batch-queries
- load text files into tables
- create and read dumps of tables
- export data to various formats: CSV, XML, PDF, ISO/IEC 26300 - OpenDocumentText and
- Spreadsheet, Word, Excel and L^{A}T$_{E}$X formats
- administer multiple servers
- manage MySQL users and privileges
- check referential integrity in MyISAM tables
- using Query-by-example (QBE), create complex queries automatically connectingrequired tables
- create PDF graphics of your Database layout
- search globally in a database or a subset of it
- transform stored data into any format using a set of predefined functions, likedisplaying

2.4 HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirements: -

- Pentium-IV(Processor).
- 256 MB Ram
- 512 KB Cache Memory
- Hard disk 10 GB
- Microsoft Compatible 101 or more Key Board

Software Requirements: -

● Operating System :	Windows
● Web-Technology:	PHP
● Front-End:	HTML,CSS,JAVASCRIPT
● Back-End:	MySQL
● Web Server:	Apache SERVER.

CHAPTER-3: SOFTWARE REQUIRMENTS SPECIFICATION

3.1 User requirement:

1. User-Friendly Interface:
- Users should find the system easy to navigate with an intuitive interface.

2. Search Flexibility:
- Users should be able to search for tickets using various parameters, such as date, time, destination, and type of service (e.g., express, economy, first class).

3. Real-Time Information:
- Users need access to real-time information about schedules, availability, and pricing.

4. Secure Authentication:
- User accounts should require secure login credentials to protect personal information and bookings.

5. Booking Options:
- Users should have the option to book one-way, round-trip, or multi-destination tickets.

6. Payment Convenience:
- The system should support a variety of payment methods, including credit/debit cards, mobile wallets, and net banking.

7. Booking Confirmation:
- Users should receive clear booking confirmations via email or SMS with all relevant details.

8. Cancellation and Refunds:
- The refund process should be transparent and timely.

9. User Profiles:
- Users should be able to create and manage their profiles, including updating contact information and preferences.

10. Notifications and Alerts:
- Users should receive notifications about booking confirmations, upcoming trips, and special promotions.

3.1 Functional Requirements:

1. User Registration and Login:
- Users should be able to register with the system using their personal details.

2. Search and Booking:
- Users should be able to search for available train and bus routes.

3. Payment Processing:
- The system should support secure online payment methods for booking tickets.

4. User Profile Management:
- Users should be able to edit their profile information.

5. Admin Panel:
- Administrators should have access to an admin panel to manage routes, schedules, and user accounts.

6. Seat Selection:
- Users should be able to select specific seats on a bus or train, if applicable.

7. Cancellation and Refunds:
- Users should be able to cancel bookings, and the system should process refunds based on cancellation policies.

8. Real-time Updates:
- The system should provide real-time updates on the status of buses and trains, including delays or cancellations.

9. Ticket Generation:
- The system should generate digital tickets with unique booking IDs.

10. Notifications:
- Users should receive notifications (email or SMS) about booking confirmations, changes in schedule, and cancellations.

11. Security and Privacy:
- The system should ensure the security of user data and payment information.

12. Multiple Payment Options:
- Users should be able to pay for tickets using various methods like credit/debit cards, mobile wallets, and net banking.

3.2 Non Functional Requirements:

1. Performance:

- Response Time: The system should respond to user requests within a specified time frame, e.g., booking requests should be processed within seconds.
- Scalability: The system should be scalable to handle increased user load during peak times, such as holidays or festivals.
- Concurrency: It should support a high level of concurrent users without performance degradation.

2. Reliability:

- Availability: The system should be available 24/7 with minimal downtime for maintenance.
- Fault Tolerance: It should continue to operate smoothly in the event of server failures or other technical issues.
- Data Integrity: User data and booking information should be stored securely and with data integrity.

3. Security:

- Data Encryption: User data, payment information, and transactions should be encrypted to protect against data breaches.
- Authentication and Authorization: Only authorized users should have access to certain system functionalities.
- Secure Payment Processing: The system should comply with industry security standards for online payments.

4. Usability:

- User Interface Design: The user interface should be intuitive and user-friendly, catering to both novice and experienced users.
- Accessibility: The system should be accessible to users with disabilities, following accessibility standards.

5. Performance Testing:

- Load Testing: The system should undergo load testing to ensure it can handle the expected user load.
- Stress Testing: It should be tested under extreme conditions to determine its breaking point.

6.Backup and Recovery:

- Regular backups of critical data should be performed, and there should be a well-defined disaster recovery plan in place.

7. Response to Failures:

- In the event of a system failure, the system should gracefully handle errors and provide informative error messages to users.

SYSTEM DESIGN

System design is the solution to the creation of a new system. This phase is composed of several system. This phase focuses on the detailed implementation of the feasible system. It translates design specification into performance specification. System design has two phases of development logical and physical design.

During logical design phase the analyst describes inputs (sources), outputs (destination), databases (data sources) and procedure (data flows) all in a format that meets the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagram and database design.

The physical design is followed by physical design or coding. Physical design procedure the working system by defining the specification, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

As this website is static, we need not worry about the logical design. The only thing we should think about is the physical design on the website.

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn’t usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: - Define the boundaries along which to break; Determine into how money pieces to break.

PHYSICAL DESIGN

The systems objectives outlined during the feasibility study serve as the basis from which the work of system design is initiated. Much of the activities involved at this stage is of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and thorough understanding of computers available in the market and the various facilities provided by the vendors

System Design Considerations

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer. When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system-designing phase:

a. Design Methodology: Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs with in the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements. The following structured diagram can appropriately represent the relationship between various modules .

b. Design Overview: In analyzing the present system a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions. It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

c. Process Modeling: System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs 17 (destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements.

4.2 DATA FLOW DIAGRAM

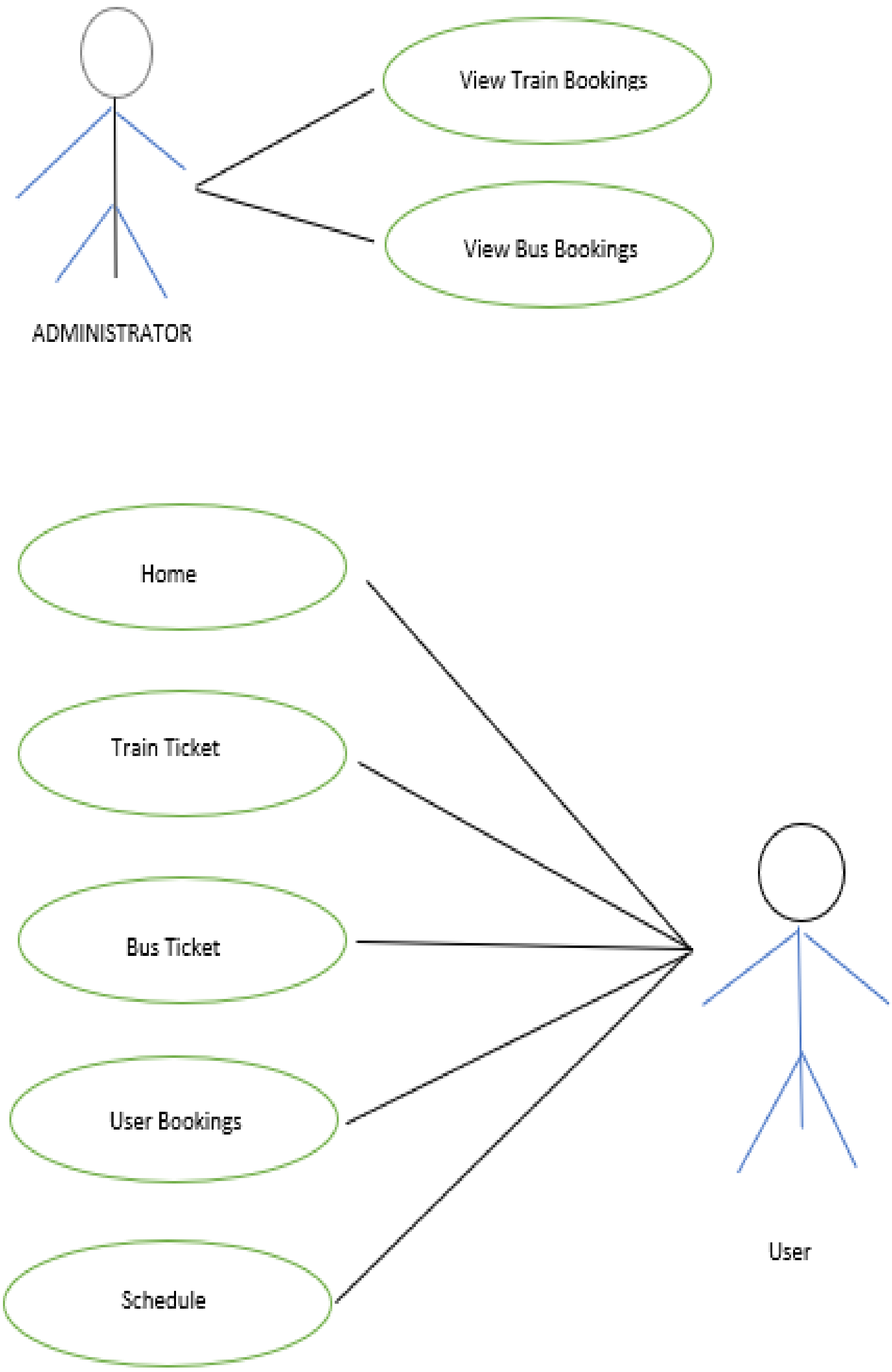
Data flow diagrams are the most commonly used way of documenting the processing of the candidate system. As their name suggest they are a pictorial way of representing the flow of data into, around, and out of the system.

- **External entities** - represents the sources of the data that enter the system or the recipients of the system that leave the system. for example - passenger is the usual receiver of information and supplier of data during form filling.

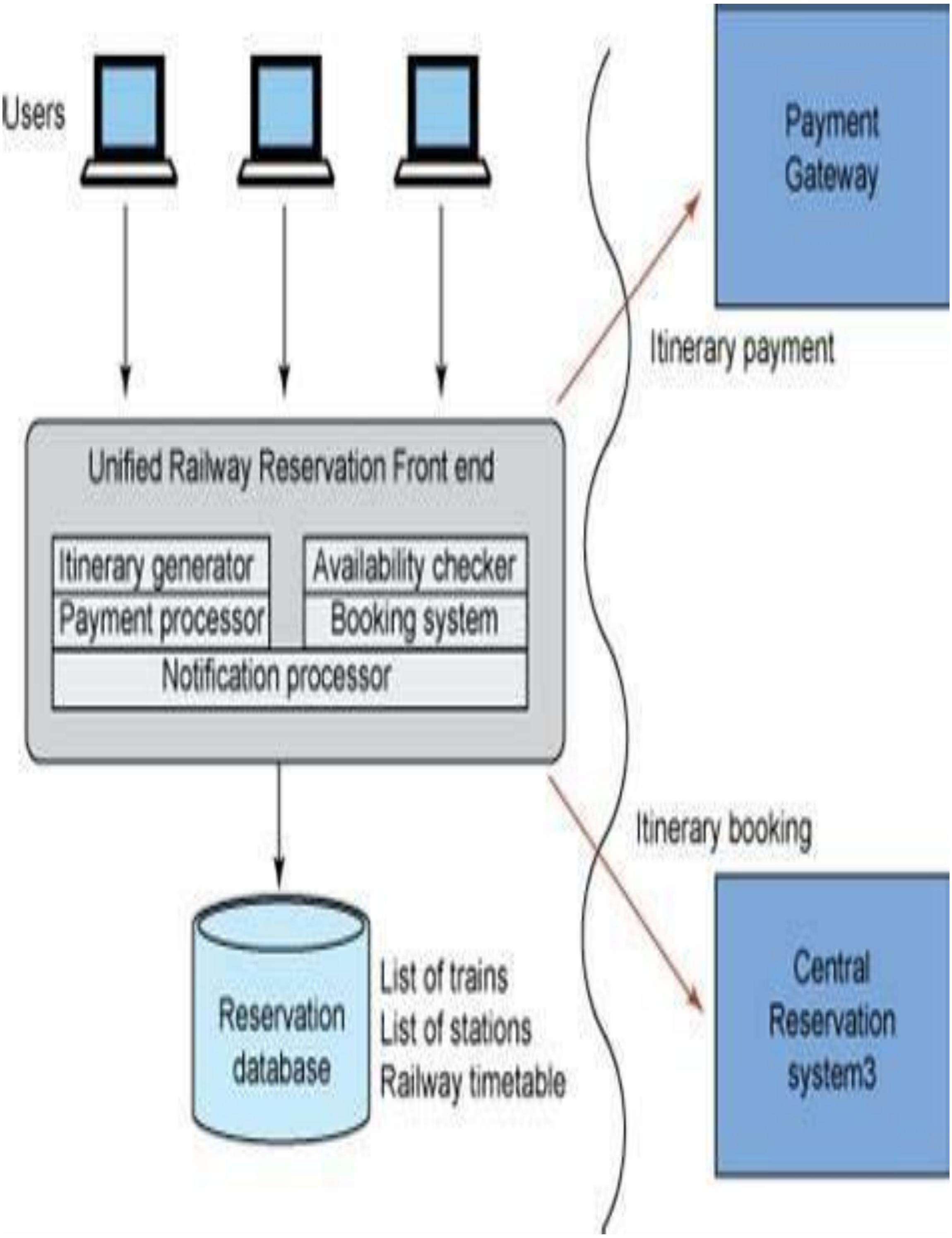
- **Data stores** - represent the stores of the data within the system example: computer files, databases or in the manual system files, etc. data stores can not be linked directly by data flows either to each other or to external entities without an intervening process to transform them.

- **Processes** - represent activities in which data is manipulated by being stored or retrieved or transformed in some way. Process names are generally unambiguous and convey as much meaning as possible without being too long. Example: verify data, acquired time schedule etc.

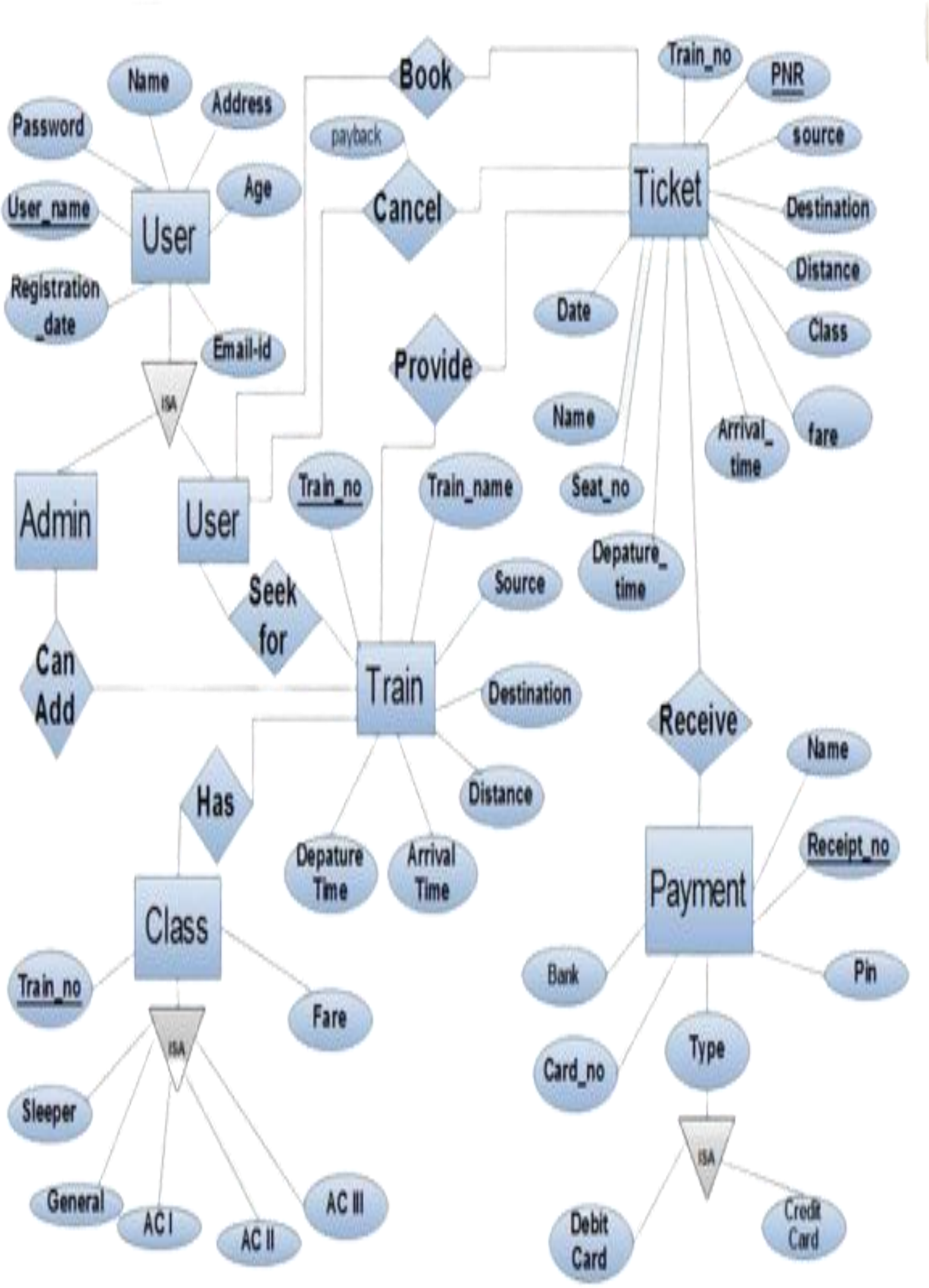
USE CASE DIAGRAM:



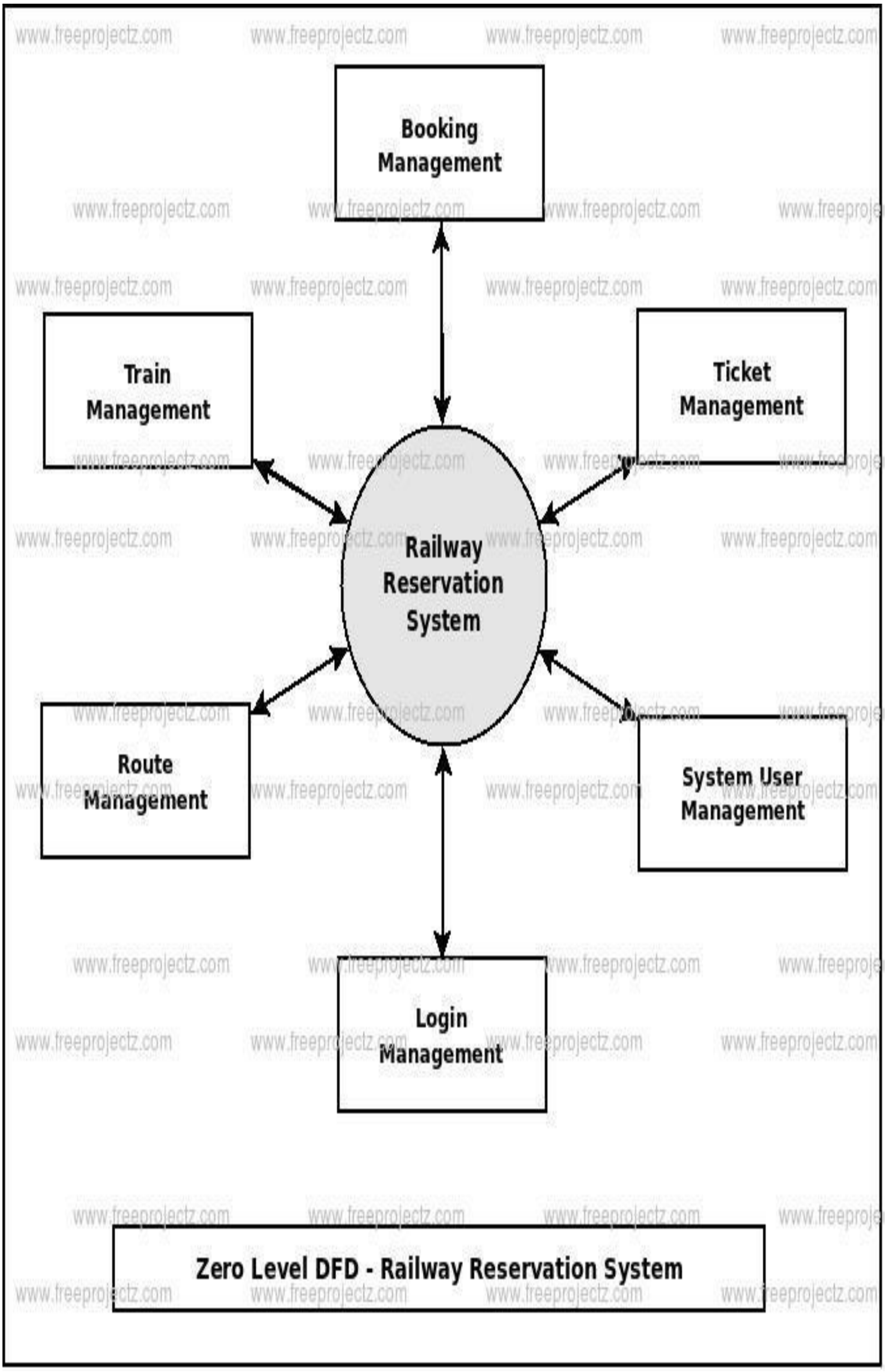
ARTITECTURE DIAGRAM:



ENTITY RELATIONSHIP DIAGRAM OF TICKET BOOKING SYSTEM:



DFD DIAGRAM


Booking Management
Train Management
Ticket Management
Railway Reservation System
Route Management
System User Management
Login Management
Zero Level DFD - Railway Reservation System

CHAPTER-4: MODULES OF TICKET BOOKING MANAGEMENT SYSTEM

This section attempts to describe each module of the project in brief, and the detailed description of each of these modules is spread throughout this document.

The Ticket Booking Management System project has been divided into three modules. They are

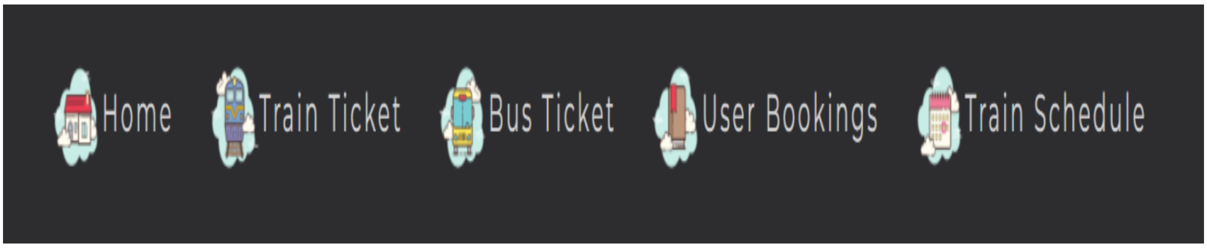
1. **User Register**

In the User register module, if there is new user, If they need to book train or bus, they should need Registered Account according to that reason user need to must have registered account before accessing any information from the official booking web portal, so they can create them account by using this User Registration module, this module ask the personal details of the new users for new accounts.

1. **User Sign In**

After successfully creation of new account, user need to sign in to the portal. User Sign In portal provides five different modules, They are.

1.1 Home
1.2 Train Ticket
1.3 Bus Ticket
1.4 User Bookings
1.5 Train Schedule



2. **Admin Sign In**

Here, In Admin Panel, Admin can manage all the user information and the train and bus bookings. It had two different modules in the admin section, they are.

2.1 View Train Bookings

2.2 View Bus Booking

DEVELOPMENT PROCESS MODEL

When building a web system, it is important to go through a series of predictable steps, a road map that helps you create a timely, high quality result and it is known as process model. Otherwise system ends up with lots of pits and falls. It is important because it provides stability, control, and organization to an activity that can, if left uncontrolled, become quite chaotic.

Varieties of models are available for web engineering and design. Each represents an attempt to bring order to an inherently chaotic activity; each of the models has been characterized in a way that assists in the control and co-ordination of the real software project.

The selection of process model is depending on the system type. One process might be appropriate for creating software for an aircraft avionics system, while an entirely different process would be indicated for the creation of a web site.

The process model choice is based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. For this project the waterfall model is selected because of familiarity, small size of project.

The other reasons are, this process provides clear progress, easy management and each tasks are clear here.

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD) mainly deals with the software development activities.

Defining a System A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

8 In a system the different components are connected with each other and they are interdependent.

For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth.

System Life Cycle System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system. System development life cycle means combination of various activities.

In other words we can say that various activities put together are referred as system development life cycle.

Following are the different phases of software development cycle:

♣ System study
♣ Feasibility study
♣ System analysis
♣ System design
♣ Coding
♣ Testing
♣ Implementation

THE WATERFALL MODEL

The waterfall model is also known as linear sequential or classic life cycle model. It suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, implementation, testing and support.

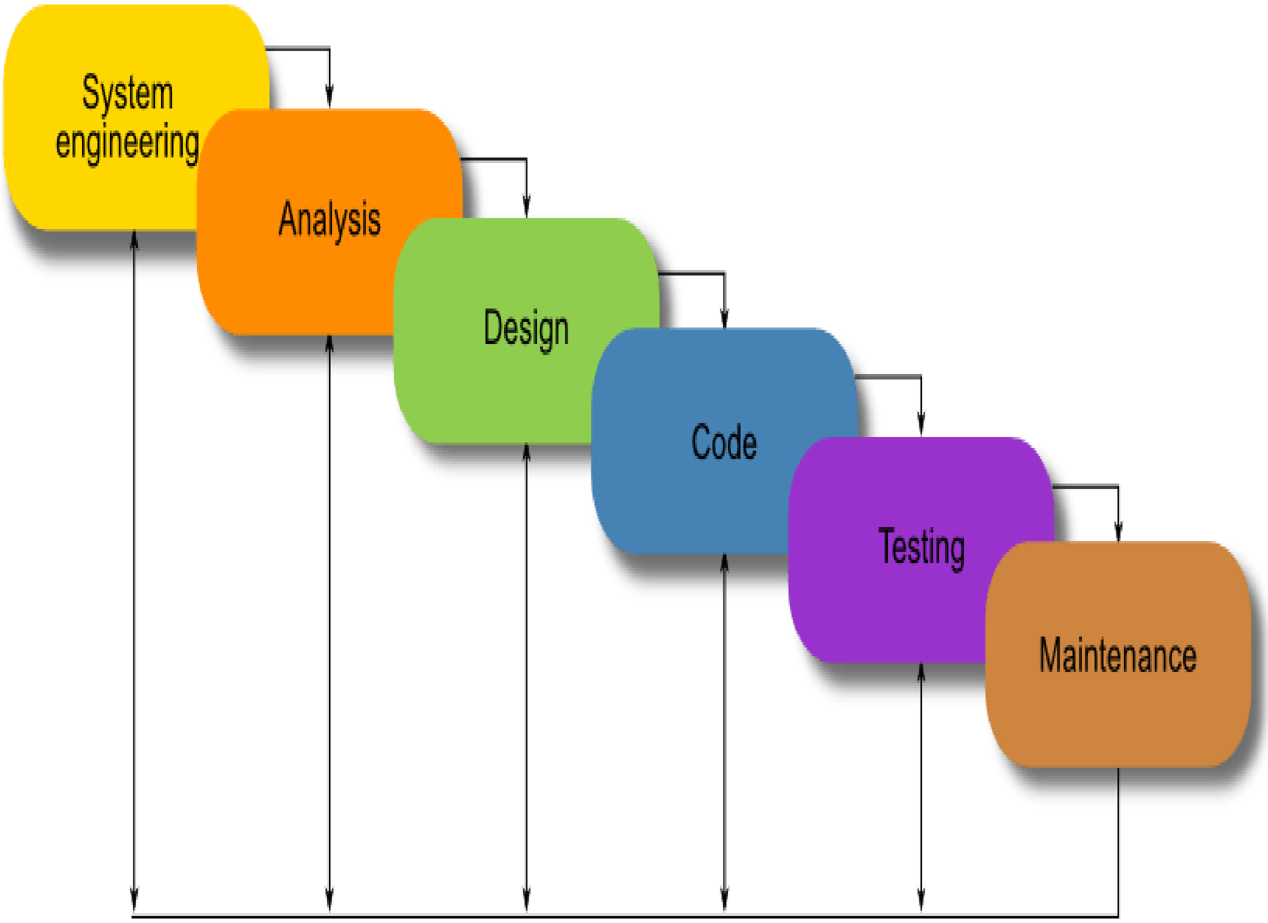
The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.

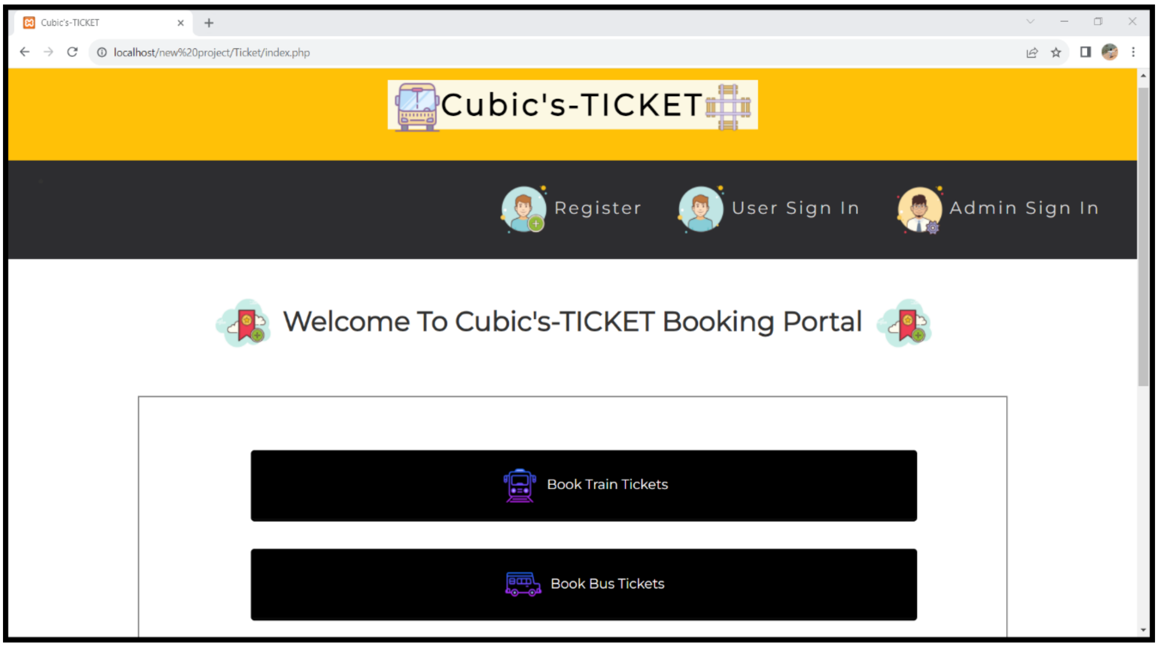
The waterfall model shown in the figure below:



CHAPTER-5: IMPLEMENTATION

1.1 Screenshots:

1.1.1 Ticket Booking Portal :



Cubic's-TICKET

localhost/new%20project/Ticket/index.php

Cubic's-TICKET

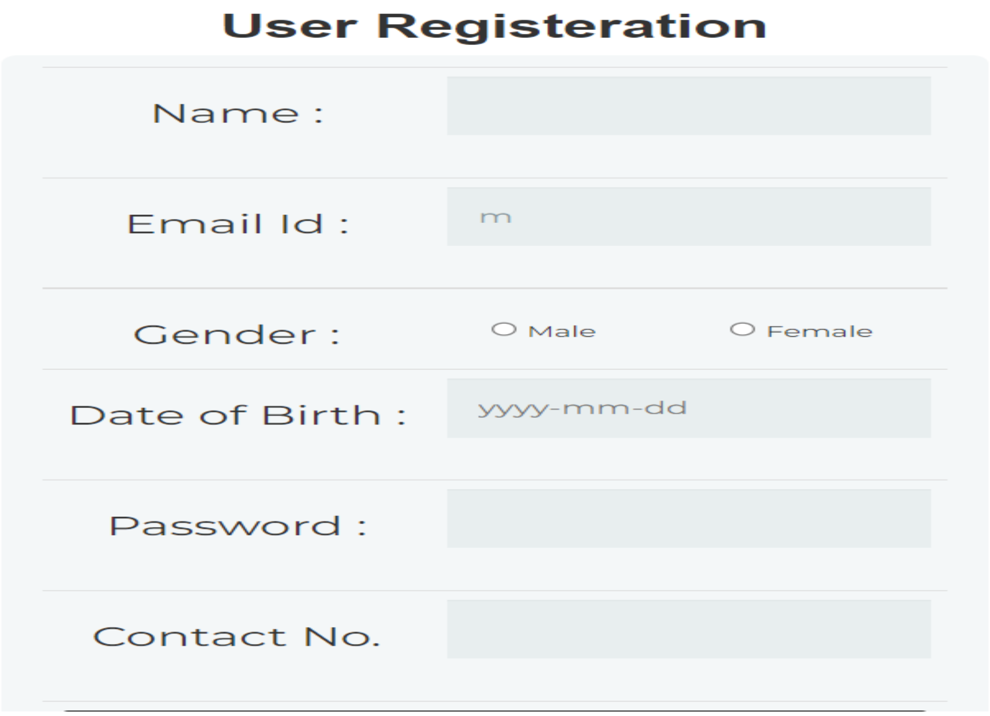
Register User Sign In Admin Sign In

Welcome To Cubic's-TICKET Booking Portal

Book Train Tickets

Book Bus Tickets

1.1.2 User Registeration:



User Registeration

Name :

Email Id : m

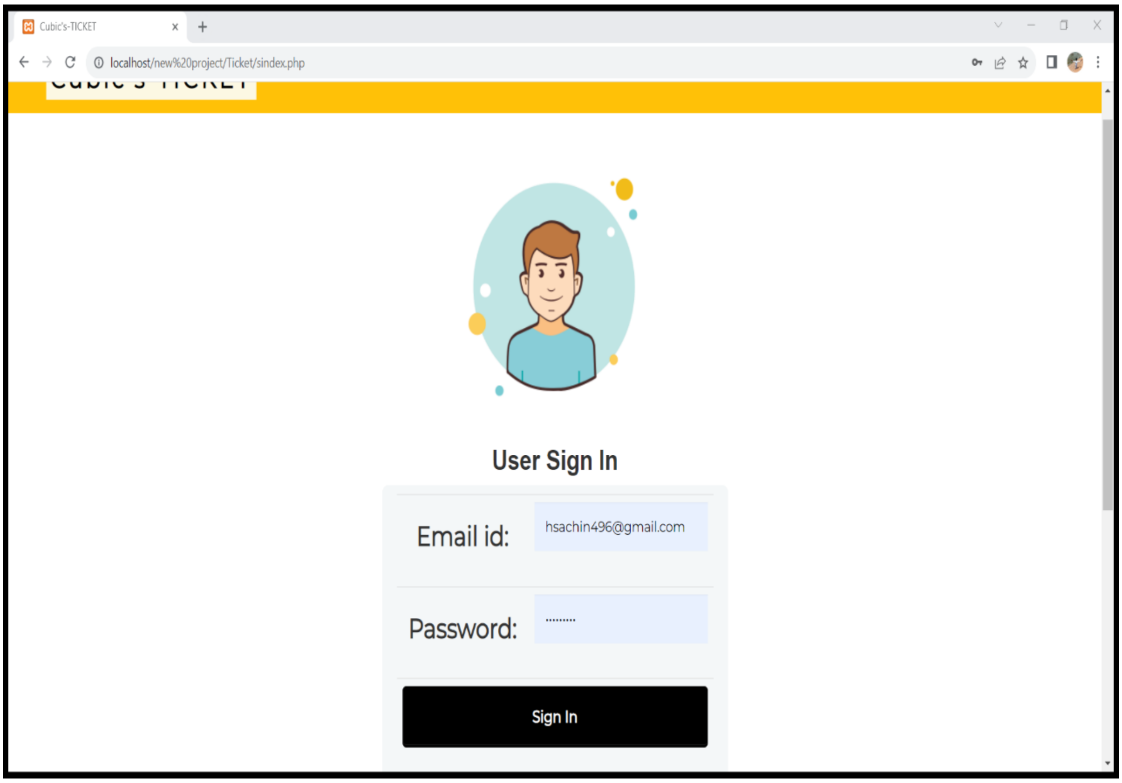
Gender : ○ Male ○ Female

Date of Birth : yyyy-mm-dd

Password :

Contact No.

1.1.3 User sign in:

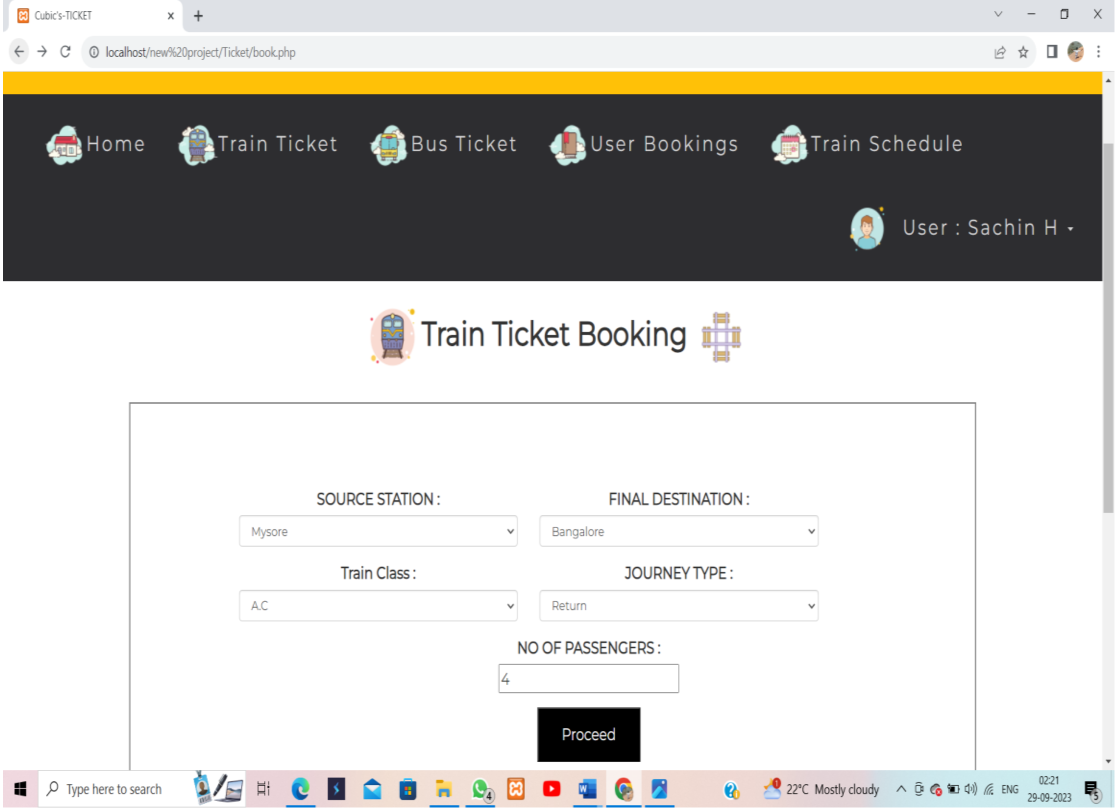


User Sign In

Email id: hsachin496@gmail.com

Password: ·········

1.1.4 User Train Ticket Booking:



Train Ticket Booking

SOURCE STATION : Mysore

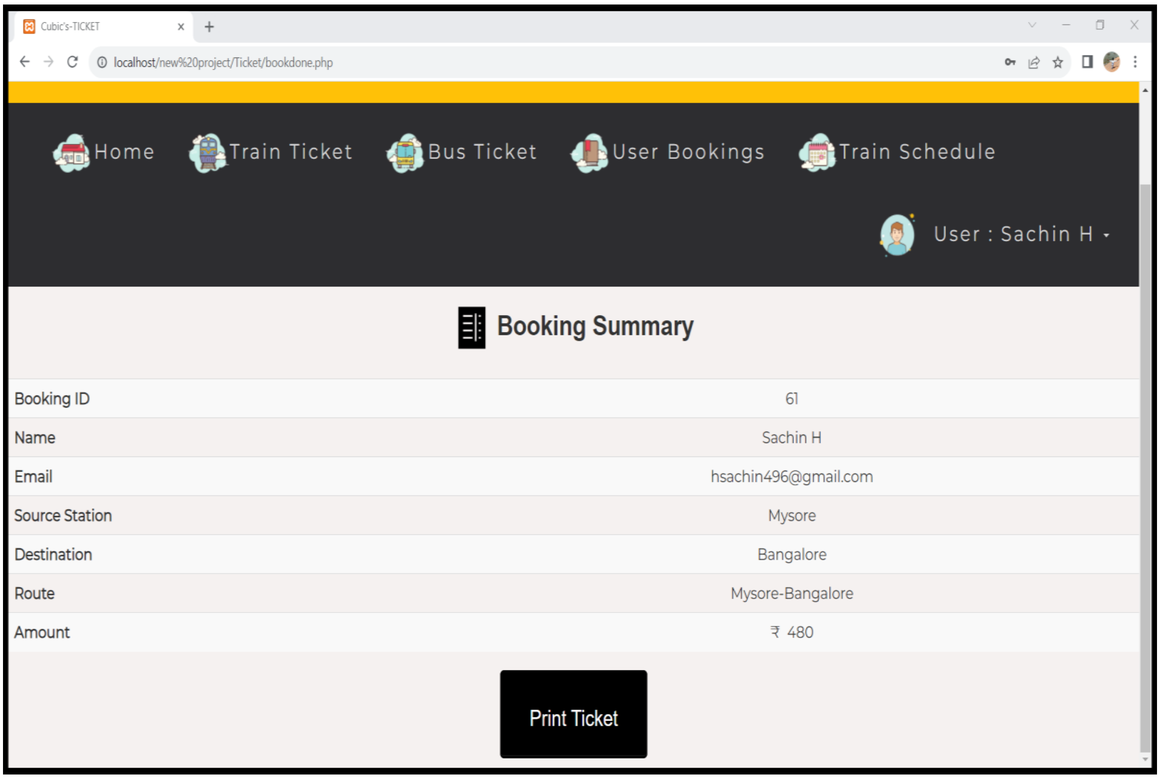
FINAL DESTINATION : Bangalore

Train Class : A.C

JOURNEY TYPE : Return

NO OF PASSENGERS : 4

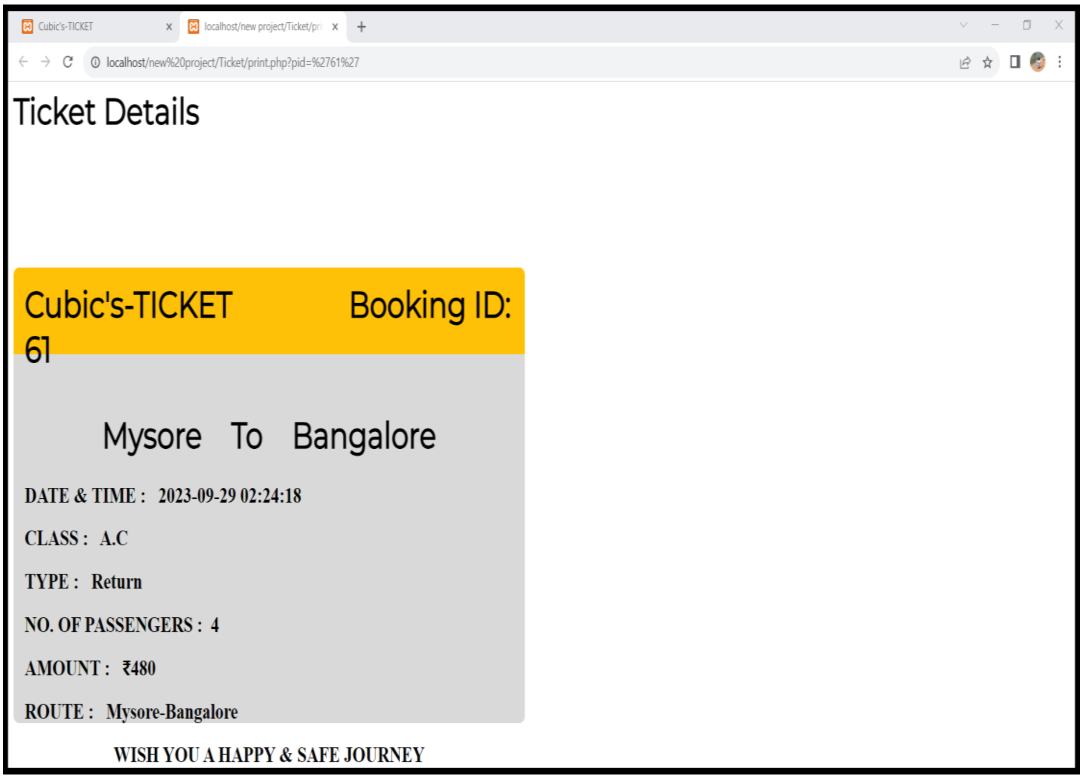
1.1.5 User Booking Summary:



Booking Summary

Booking ID	61
Name	Sachin H
Email	hsachin496@gmail.com
Source Station	Mysore
Destination	Bangalore
Route	Mysore-Bangalore
Amount	₹ 480

1.1.6 User Train Ticket Details:


Cubic's-TICKET
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/print.php?pid=%2761%27

Ticket Details

Cubic's-TICKET Booking ID: 61

Mysore To Bangalore

DATE & TIME : 2023-09-29 02:24:18

CLASS : A.C

TYPE : Return

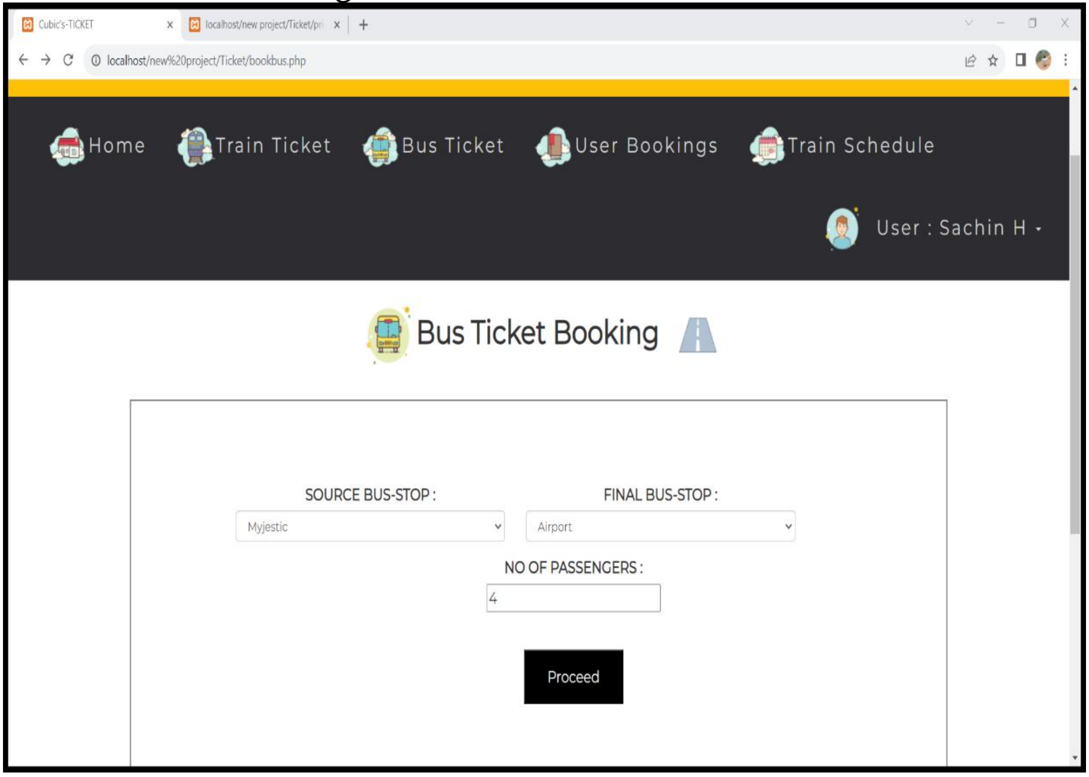
NO. OF PASSENGERS : 4

AMOUNT : ₹480

ROUTE : Mysore-Bangalore

WISH YOU A HAPPY & SAFE JOURNEY

1.1.7 User Bus Ticket Booking:



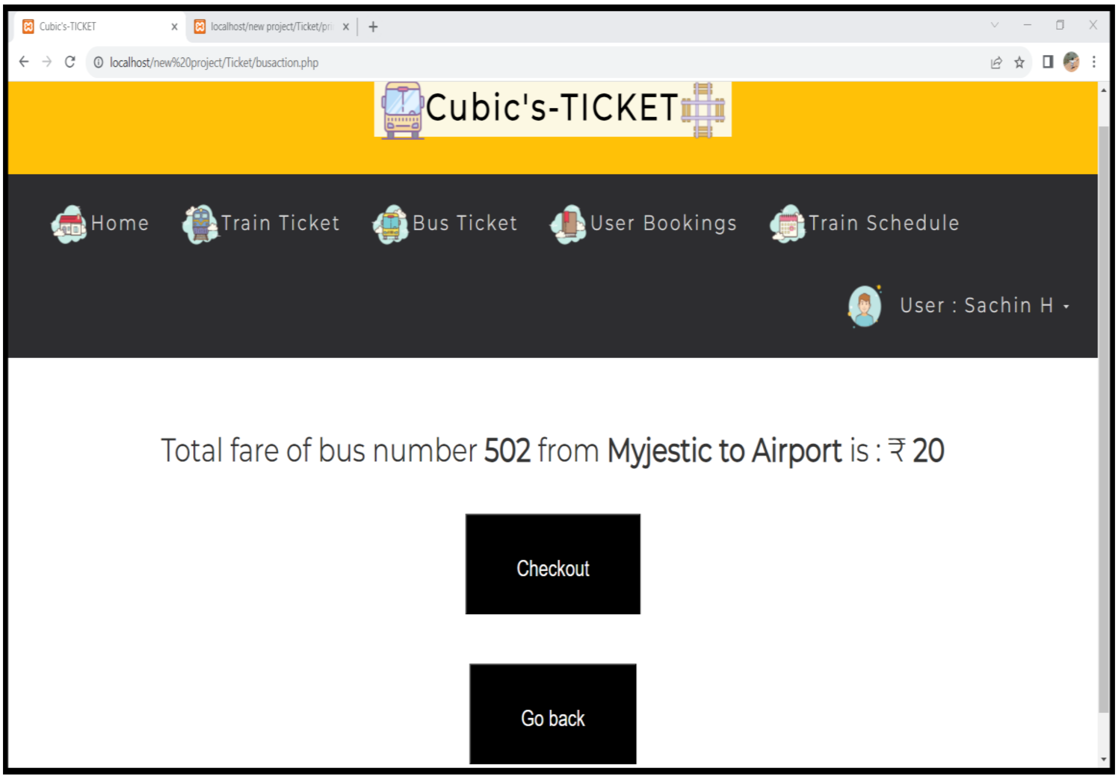
Bus Ticket Booking

SOURCE BUS-STOP : Myjestic

FINAL BUS-STOP : Airport

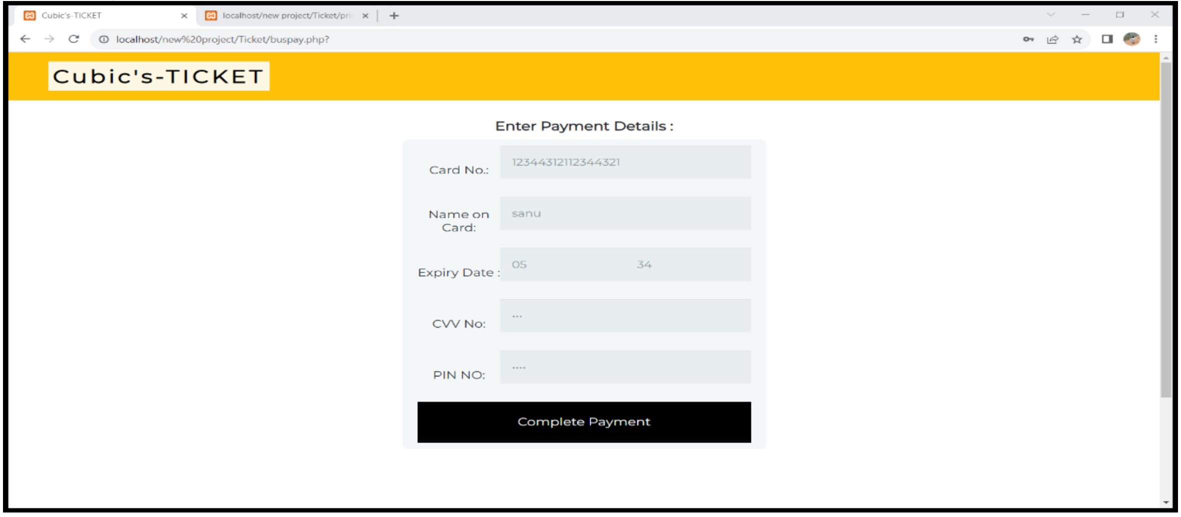
NO OF PASSENGERS : 4

1.1.8 User Total fare:



Total fare of bus number 502 from Myjestic to Airport is : ₹ 20

1.1.9 User Payment Details:



Cubic's-TICKET
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/buspay.php?

Cubic's-TICKET

Enter Payment Details :

Card No.: 12344312112344321

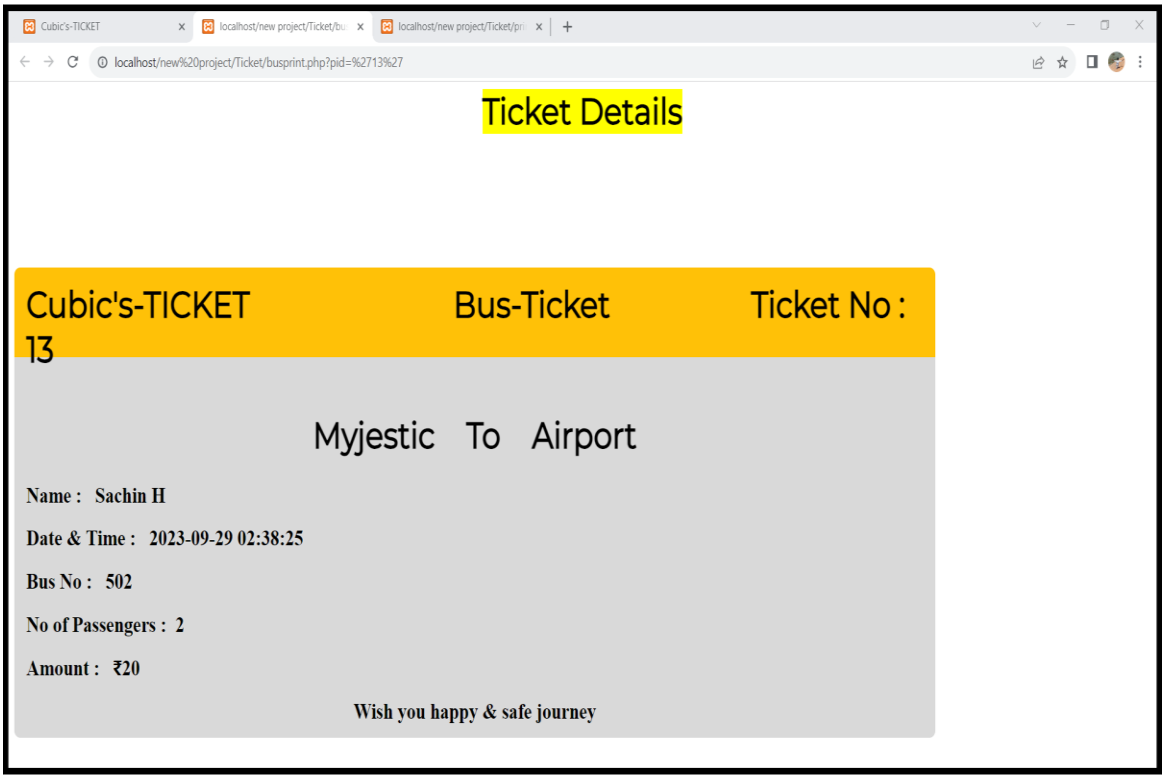
Name on Card: sanu

Expiry Date : 05 34

CVV No: ...

PIN NO: ....

1.1.10 User Bus Ticket Details:



Cubic's-TICKET
localhost/new project/Ticket/bu
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/busprint.php?pid=%2713%27

Ticket Details

Cubic's-TICKET Bus-Ticket Ticket No : 13

Myjestic To Airport

Name : Sachin H

Date & Time : 2023-09-29 02:38:25

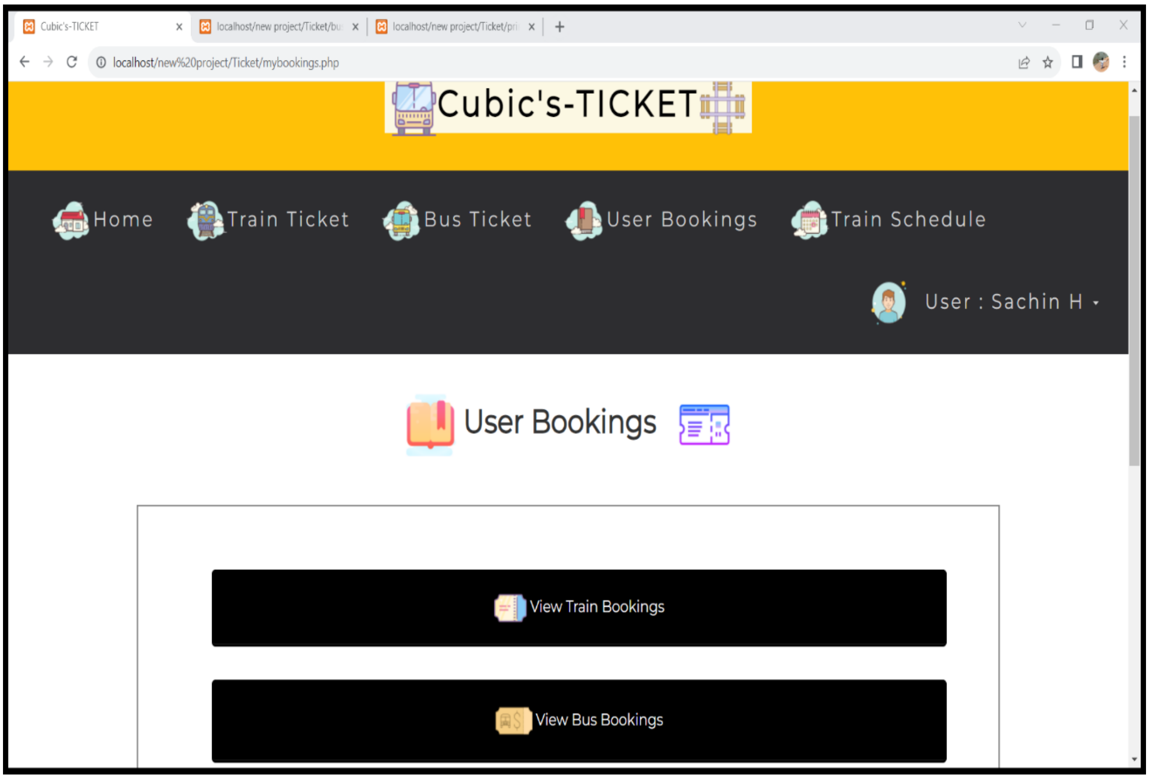
Bus No : 502

No of Passengers : 2

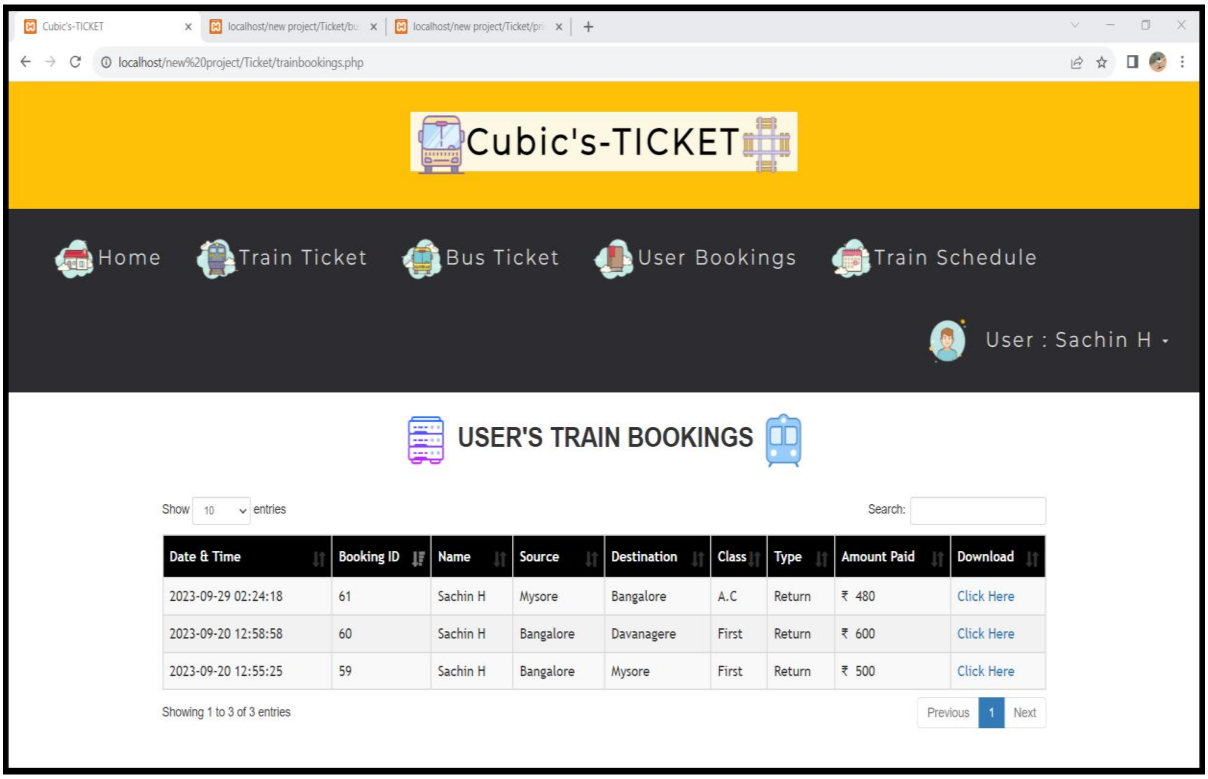
Amount : ₹20

Wish you happy & safe journey

1.1.11 User Bookings:



1.1.12 :Users Train Bookings:



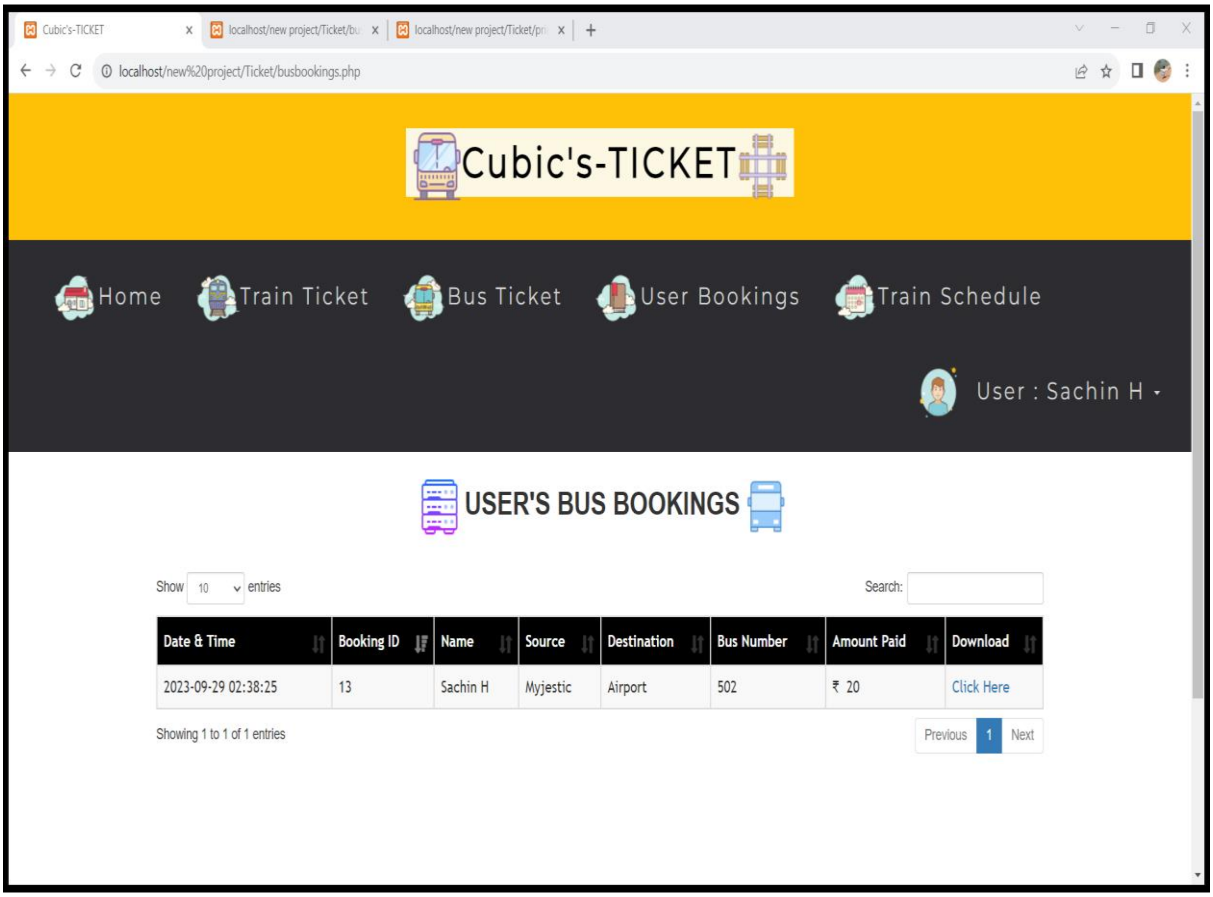
USER'S TRAIN BOOKINGS

Show 10 entries Search:

Date & Time	Booking ID	Name	Source	Destination	Class	Type	Amount Paid	Download
2023-09-29 02:24:18	61	Sachin H	Mysore	Bangalore	A.C	Return	₹ 480	Click Here
2023-09-20 12:58:58	60	Sachin H	Bangalore	Davanagere	First	Return	₹ 600	Click Here
2023-09-20 12:55:25	59	Sachin H	Bangalore	Mysore	First	Return	₹ 500	Click Here

Showing 1 to 3 of 3 entries

1.1.13 User Bus Bookings:



USER'S BUS BOOKINGS

Show 10 entries Search:

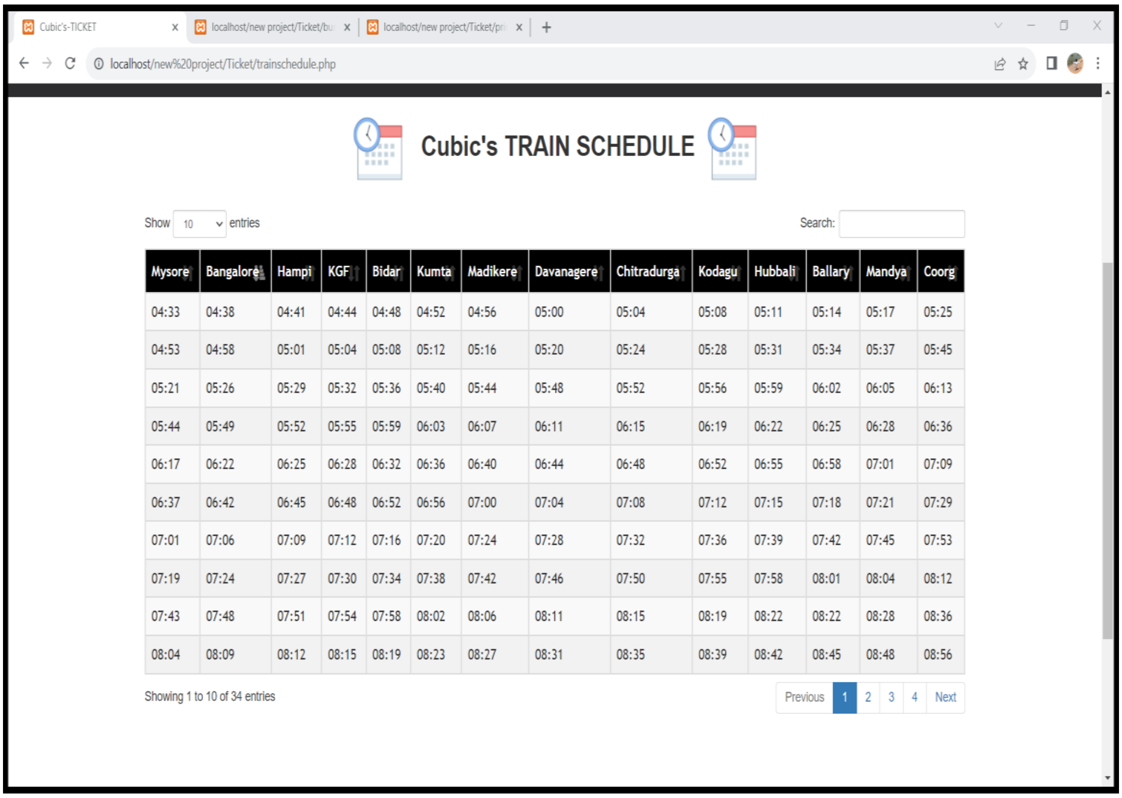
Date & Time	Booking ID	Name	Source	Destination	Bus Number	Amount Paid	Download
2023-09-29 02:38:25	13	Sachin H	Myjestic	Airport	502	₹ 20	Click Here

Showing 1 to 1 of 1 entries

Previous 1 Next

:

1.1.14 Train Schedule:



Cubic's-TICKET
localhost/new project/Ticket/bu
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/trainschedule.php

Cubic's TRAIN SCHEDULE

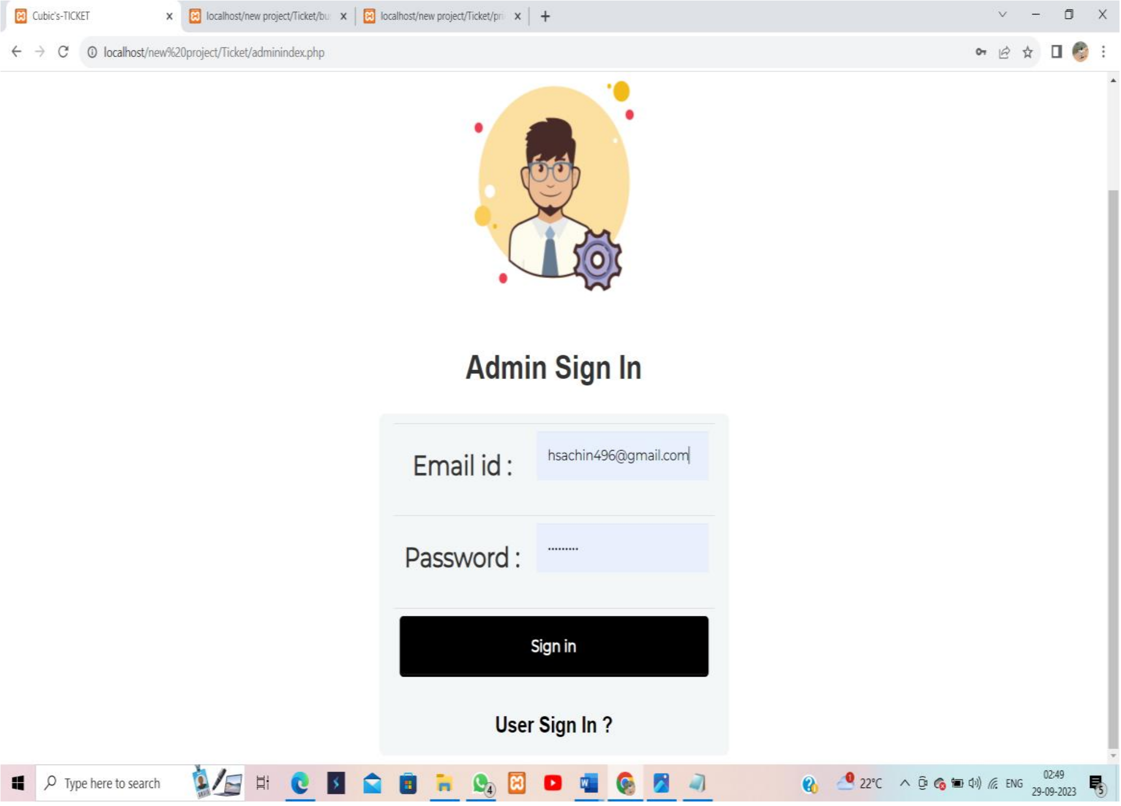
Show 10 entries

Search:

Mysore	Bangalore	Hampi	KGF	Bidar	Kumta	Madikere	Davanagere	Chitradurga	Kodagu	Hubbali	Ballary	Mandya	Coorg
04:33	04:38	04:41	04:44	04:48	04:52	04:56	05:00	05:04	05:08	05:11	05:14	05:17	05:25
04:53	04:58	05:01	05:04	05:08	05:12	05:16	05:20	05:24	05:28	05:31	05:34	05:37	05:45
05:21	05:26	05:29	05:32	05:36	05:40	05:44	05:48	05:52	05:56	05:59	06:02	06:05	06:13
05:44	05:49	05:52	05:55	05:59	06:03	06:07	06:11	06:15	06:19	06:22	06:25	06:28	06:36
06:17	06:22	06:25	06:28	06:32	06:36	06:40	06:44	06:48	06:52	06:55	06:58	07:01	07:09
06:37	06:42	06:45	06:48	06:52	06:56	07:00	07:04	07:08	07:12	07:15	07:18	07:21	07:29
07:01	07:06	07:09	07:12	07:16	07:20	07:24	07:28	07:32	07:36	07:39	07:42	07:45	07:53
07:19	07:24	07:27	07:30	07:34	07:38	07:42	07:46	07:50	07:55	07:58	08:01	08:04	08:12
07:43	07:48	07:51	07:54	07:58	08:02	08:06	08:11	08:15	08:19	08:22	08:22	08:28	08:36
08:04	08:09	08:12	08:15	08:19	08:23	08:27	08:31	08:35	08:39	08:42	08:45	08:48	08:56

Showing 1 to 10 of 34 entries

1.1.15 Admin Sign In:



Cubic's-TICKET
localhost/new project/Ticket/bu
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/adminindex.php

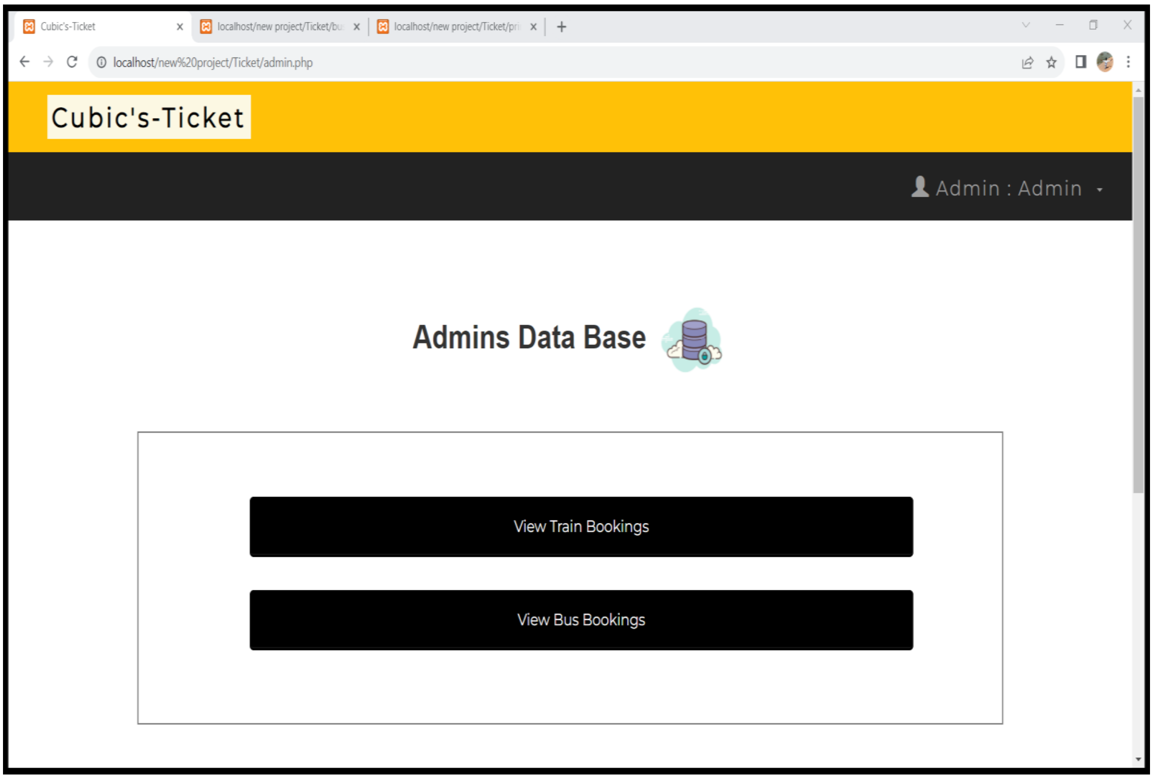
Admin Sign In

Email id : hsachin496@gmail.com

Password : ....

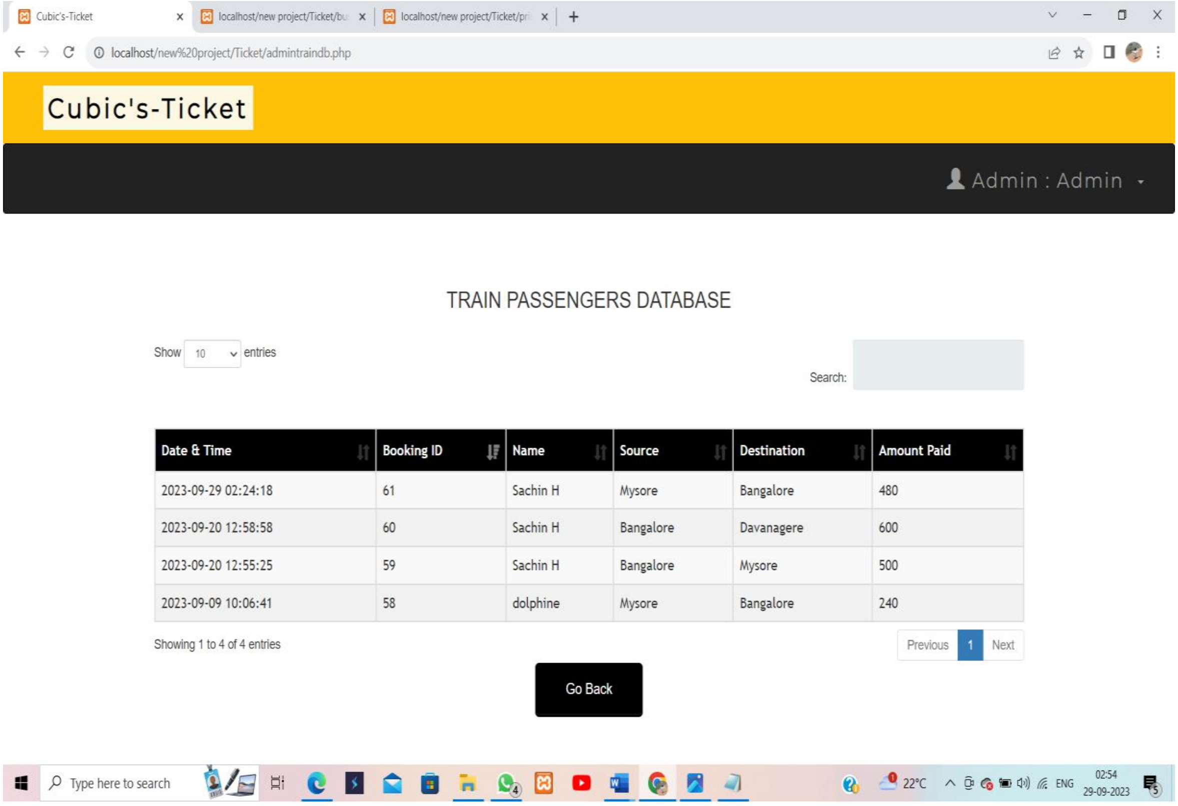
Sign in
User Sign In ?
Type here to search
22°C
ENG
02:49
29-09-2023

1.1.1 Admin Database:



Cubic's-Ticket
localhost/new project/Ticket/bu
localhost/new project/Ticket/pri
localhost/new%20project/Ticket/admin.php
Cubic's-Ticket
Admin : Admin
Admins Data Base
View Train Bookings
View Bus Bookings

1.1.2 Train Passengers Database:



TRAIN PASSENGERS DATABASE

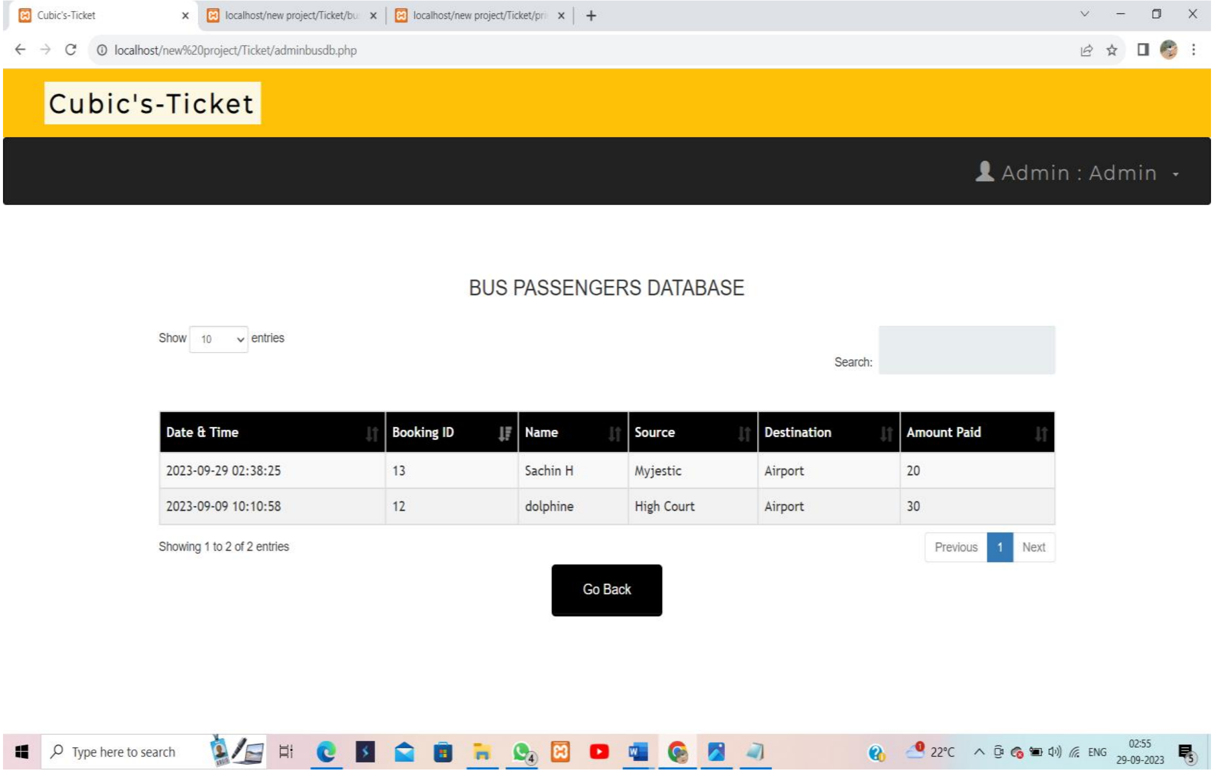
Show 10 entries

Search:

Date & Time	Booking ID	Name	Source	Destination	Amount Paid
2023-09-29 02:24:18	61	Sachin H	Mysore	Bangalore	480
2023-09-20 12:58:58	60	Sachin H	Bangalore	Davanagere	600
2023-09-20 12:55:25	59	Sachin H	Bangalore	Mysore	500
2023-09-09 10:06:41	58	dolphine	Mysore	Bangalore	240

Showing 1 to 4 of 4 entries

1.1.3 Bus Passenger Database:



BUS PASSENGERS DATABASE

Show 10 entries

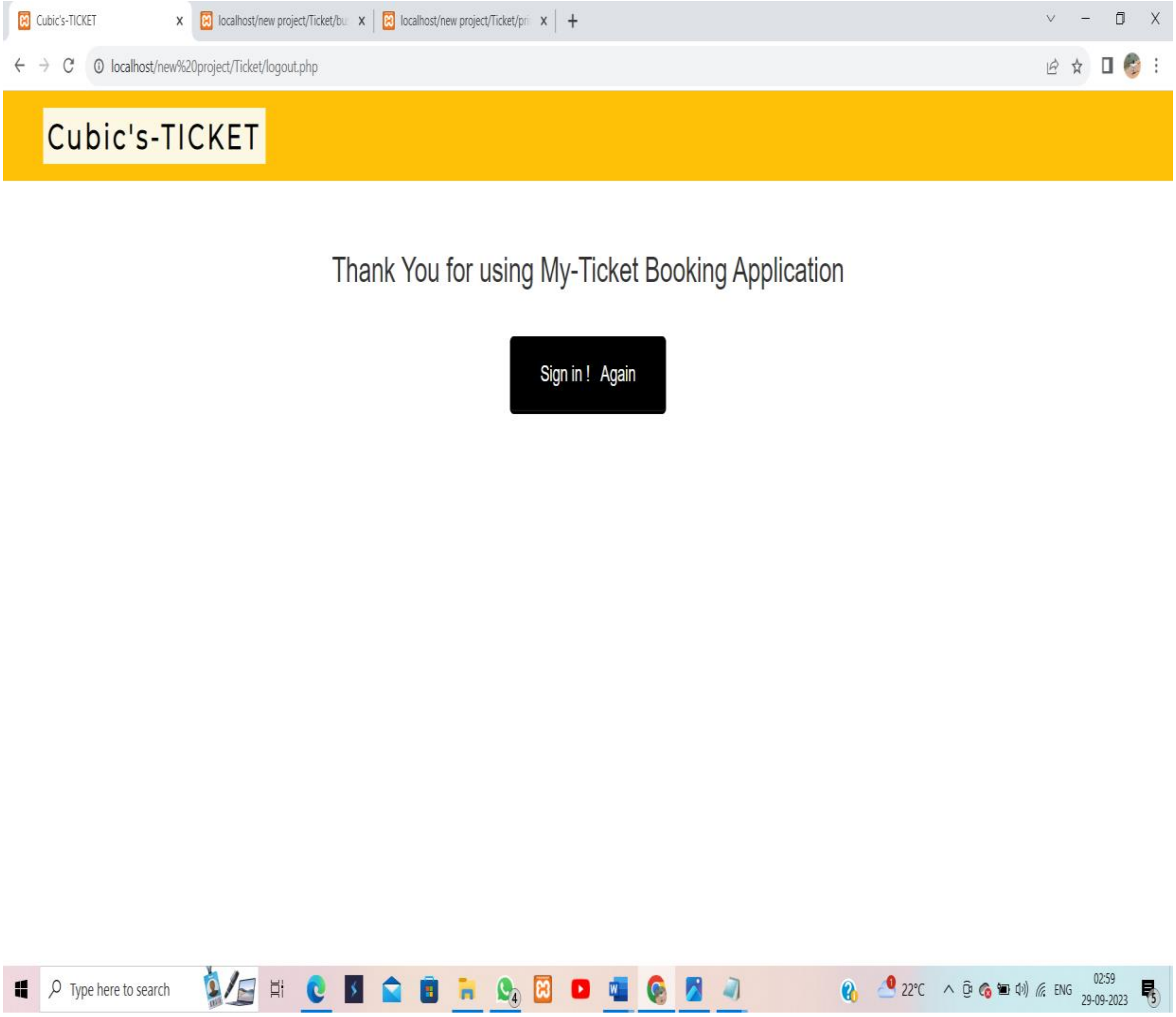
Search:

Date & Time	Booking ID	Name	Source	Destination	Amount Paid
2023-09-29 02:38:25	13	Sachin H	Myjestic	Airport	20
2023-09-09 10:10:58	12	dolphine	High Court	Airport	30

Showing 1 to 2 of 2 entries

t

1.1.4 Thank You for Using My-Ticket Booking Application:



Cubic's-TICKET
localhost/new project/Ticket/bu
localhost/new project/Ticket/pr
localhost/new%20project/Ticket/logout.php
Cubic's-TICKET

Thank You for using My-Ticket Booking Application

Sign in ! Again
Type here to search
22°C
ENG
02:59
29-09-2023

SOURCE CODE

Index Page:

```
<?php

include 'homehead.php';
?>


<style >
  .container{
     border-spacing: 10px;
     font-family: Montserrat, sans-serif;
     font-size: 18px !important;
     border: 2px solid grey;
     margin-top: 50px;
     margin-bottom: 200px;
     padding-top: 50px;
     padding-right: 120px;
     padding-bottom: 50px;
     padding-left: 150px;
     align-content: center;

    }
</style>

<html>
<head>
<link rel='stylesheet' href='index.css'>
<br><br>
  <h1   text-transform: capitalize;><center><b><img
src="https://img.icons8.com/clouds/90/000000/add-bookmark.png"/> Welcome To Cubic's-
TICKET Booking Portal <img src="https://img.icons8.com/clouds/90/000000/add-
bookmark.png"/></center></b></h1>

<div class="container">

  <td><a href='book.php'><button style="background-color: black ; border-color:black "
><img src="https://img.icons8.com/nolan/54/train.png"/>&nbsp&nbspBook Train Tickets
</button></a></td>
 </tr>
<br> <br>

<tr>

<td><a href='bookbus.php'><button style="background-color: black ; border-color:black
"><img src="https://img.icons8.com/nolan/54/bus.png"/>&nbsp&nbspBook Bus
Tickets</button></a></td></td>
</tr>
```

```
</table>
</tr>
</div>
<?php include 'footer.php';
?>

</html>
```

Admin Page:

```
<?php
include 'connect.php' ;
session_start();
if ($_SESSION['log'] == '')
{
    header("location:adminindex.php");
}
include 'adminheader.php';
?>

<br>
<br>

<style >
          .container{
        border-spacing: 10px;

      font-family: Montserrat, sans-serif;
     font-size: 18px !important;
      border: 2px solid grey;
        margin-top: 50px;
        margin-bottom: 200px;
       padding-top: 50px;
      padding-right: 120px;
      padding-bottom: 50px;
      padding-left: 150px;
      align-content: center;
    }
</style>

<html>
<head>
<link rel='stylesheet' href='index.css'>
<br><br>
  <h1><center><b> Admins Data Base <img
src="https://img.icons8.com/clouds/100/000000/lock-database-.png"/> </center></b></h1>

<div class="container">

  <td><a href='admintraindb.php'><button style="background-color: black ; border-
color:black "  >View Train Bookings </button></a></td>
```

```
 </tr>
<br> <br>

<tr>

<td><a href='adminbusdb.php'><button style="background-color: black ; border-color:black
">View Bus Bookings</button></a></td></td>
</tr>
</table>
</tr>
</div>

<?php include 'footer.php';
?>
</html>
```

Train Booking:

```
<?php
include 'connect.php' ;
session_start();
if ($_SESSION['log'] == '')
{
   header("location:sindex.php");
}
include 'header.php';
?>

  <div>

  </div>
<style>

table#database_table {
   font-size:16px;
   font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
   border-collapse: collapse;
   border-spacing: 0;
}

#database_table td, #database_table th {
   border: 1px solid #ddd;
   text-align: left;
   padding: 8px;
}

#database_table tr:nth-child(even){background-color: #f2f2f2}

#database_table th {
   padding-top: 11px;
   padding-bottom: 11px;
   background-color: black;
```

```
    color: white;
}


</style>


<head>
        <title>Cubic's-TICKET DATABASE</title>

    </head>
    <body >
          <h2><center><b><img src="https://img.icons8.com/nolan/64/database.png"/>
USER'S TRAIN BOOKINGS <img
src="https://img.icons8.com/ultraviolet/60/000000/train.png"/></center></b></h2>

        <div class="container">
            <br />
                <table id="database_table" class="table table-striped table-bordered">
                    <thead>
                      <tr>
  <th>Date & Time </th>
  <th>Booking ID </th>
  <th>Name</th>
  <th>Source</th>
  <th>Destination</th>
  <th>Class</th>
  <th>Type</th>
  <th>Amount Paid</th>
  <th>Download</th>
   </tr>
                    </thead>
                    <tbody>




<?php



$sql_transactions="SELECT * FROM `transactions`  WHERE
`email`='".$_SESSION['email']."' " ;
$result = $connect->query($sql_transactions);
while($row = $result->fetch_assoc()){

echo'<tr class="class="table table-striped table-bordered"">
      <td>'.$row["Date"].'</td>
      <td>'.$row["T_No."].'</td>
      <td>'.$row["Name"].'</td>
      <td>'.$row["source"].'</td>
      <td>'.$row["dest"].'</td>
      <td>'.$row["Class"].'</td>
      <td>'.$row["Type"].'</td>
```

```
        <td>₹&nbsp&nbsp'.$row["Amt"].'</td>
        <td><a  href="print.php?pid='.$row["T_No."].' " target="_blank">Click Here</a></td>


';
}
?>
</tbody>
</table>
</div>
 <script>
 $(document).ready(function() {
    $('#database_table').DataTable( {
        "order": [[ 1, "desc" ]]
    } );
} );
 </script>
```

Bus Booking:

```
<?php
include 'connect.php' ;
session_start();
if ($_SESSION['log'] == '')
{
    header("location:sindex.php");
}
include 'header.php';
?>

<style>



    .container{
        border-spacing: 10px;


      font-family: Montserrat, sans-serif;
     font-size: 18px !important;
      border: 2px solid grey;
        margin-top: 30px;
        margin-bottom: 50px;
       padding-top: 50px;
      padding-right: 50px;
      padding-bottom: 50px;
      padding-left: 150px;
      align-content: center;
    }
    .button {
```

```
  padding: 15px 32px;
  align-content: left;
  color: white;
background-color:black;
    }
#number {
  overflow: hidden;
  width: 600px;
}
input[type=number]{
    width: 250px;
}
  </style>



  </script>
</head>
<body>
  <h1><center><b><img src="https://img.icons8.com/bubbles/80/000000/bus.png"/>Bus
Ticket Booking &nbsp <img
src="https://img.icons8.com/officel/54/000000/road.png"/></center></b></h1>
 <form method='post' action ='busaction.php' align='center'>
<div class="container">
<section id="form" class="formborder">
<div class="container2">
       <form>
         <div class="form-row row justify-content-around" name="source">
  &nbsp

       <div class="form-group col-md-5" name="source">
       <label for="inputEmail4">SOURCE BUS-STOP :</label>
       <select id="inputState" class="form-control" name="source">
  <option>UBDT College</option>
   <option>Stadium</option>
   <option>High Court</option>
   <option>Myjestic</option>
   <option>Dental College</option>
   <option>Union Park</option>
   <option>Zoo Stop</option>
   <option>SS Mall</option>
   <option>Tin Factory</option>
   <option>Ganapati Temple</option>
   <option>KSRTC Stop</option>
   <option>Airport</option>
   <option>Railway Station</option>
   <option>Vaibhav Resto</option>
   <option>Desi Street</option>
   <option>Bull Temple</option>
   <option>Cinema Max</option>
   <option>Udaya Hospital</option>
   <option>Boys Hostel Road</option>
   </select>

     <!-- <input type="email" class="form-control" id="inputEmail4" placeholder="FROM">-
-></div>
<div class="form-group col-md-5" name="dest">
      <label for="inputPassword4">
FINAL BUS-STOP :</label>
      <select id="inputState" class="form-control" name="dest">

   <option>UBDT College</option>
   <option>Stadium</option>
   <option>High Court</option>
   <option>Myjestic</option>
   <option>Dental College</option>
   <option>Union Park</option>
   <option>Zoo Stop</option>
   <option>SS Mall</option>
   <option>Tin Factory</option>
   <option>Ganapati Temple</option>
   <option>KSRTC Stop</option>
   <option>Airport</option>
   <option>Railway Station</option>
   <option>Vaibhav Resto</option>
   <option>Desi Street</option>
   <option>Bull Temple</option>
   <option>Cinema Max</option>
   <option>Udaya Hospital</option>
   <option>Boys Hostel Road</option>
   </select>

      <!--<input type="password" class="form-control" id="inputPassword4"
placeholder="TO">-->
  </div>



    <div class="form-row row justify-content-aroundd">
       <div class="form-group col-md-16" >
       <center><label for="inputState"  ><h8>NO OF PASSENGERS
:</h8></label></center>
```

```
       <center> <input type="number" name="number" required  min="1" max="5"
></center>
        </div>

<br>



    <div>
     <button type="submit"    class="button" name="login_submit" >Proceed</button>

 </div></form></div></section></div></body><?php include 'footer.php';?> </html>
```

Database Connecting Page:

```
<?php
$hostname  = 'localhost';
$username = 'root';
$password='';
$dbname = 'train';
$connect =  mysqli_connect($hostname , $username , $password ,$dbname) or die("Error
Connecting");
?>
```

Table Structure of Admin :

```
Table structure for table `admindatabase`
--

CREATE TABLE `admindatabase` (
  `UserID` int(10) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Email` varchar(70) NOT NULL,
  `Gender` varchar(10) NOT NULL,
  `password` varchar(50) NOT NULL,
  `DoB` date NOT NULL,
  `Joined_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admindatabase`
--

INSERT INTO `admindatabase` (`UserID`, `Name`, `Email`, `Gender`, `password`, `DoB`, `Joined_on`)
VALUES
(1, 'Admin', 'admin@gmail.com', 'M', '1234567', '2000-10-17', '2021-08-14 18:30:00');
```

TESTING

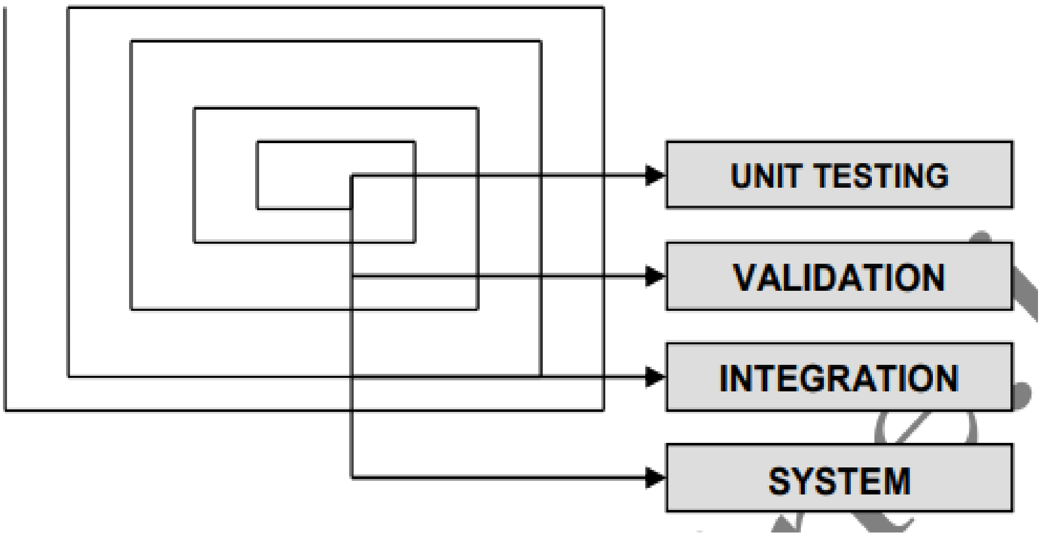
TESTING PHASE

In a software development project, errors can be injected at any stage during development. There are different techniques for detecting and eliminating errors that originate in that phase. However, no technique is perfect, and it is expected that some of the errors of the earlier phases will finally manifest themselves in the code. This is particularly true because in the earlier phases and most of the verification techniques are manual because no executable code exists. Ultimately, these remaining errors will be reflected in the code. Hence, the code developed during the coding activity is likely to have some requirement errors and design errors, in addition to errors introduced during the coding activity. Behavior can be observed, testing is the phase where the errors remaining from all the previous phases must be detected. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of software.

During testing, the program to be tested is executed with a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Due to its approach, dynamic testing can only ascertain the presence of errors in the program; the exact nature of the errors is not usually decided by testing. Testing forms the first step in determining the errors in a program. Clearly, the success of testing in revealing errors in programs depends critically on the test cases.
Testing a large system is a very complex activity, and like any complex activity it has to be broken into smaller activities. Due to this, for a project, incremental testing is generally performed, in which components and subsystems of the system are tested separately before integrating them to form the system for system testing. This form of testing, though necessary to ensure quality for a large system, introduces new issues of how to select components for testing and how to combine them to form subsystems and systems.

12.1 LEVELS OF TESTING

The different types of testing are as follows:



1. UNIT TESTING:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. INTEGRATION TESTING:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. SYSTEM TESTING:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. FIELD TESTING:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

5. ACCEPTANCE TESTING:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

WHY SYSTEM TESTING?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.

2. The effect of system errors on the files and records within the system.

Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system.

.

TEST CASES

User Registration and Login:

1. Test Case 1: Registration Validation
Input: Valid user details (name, email, password)
Expected Output: Successful registration with a confirmation message.

2. Test Case 2: Duplicate Email Check
- Input: Existing user's email during registration
- Expected Output: Error message indicating that the email is already in use.

3. Test Case 3: Login Authentication
- Input: Correct email and password
- Expected Output: Successful login and redirection to the user dashboard.

4. Test Case 4: Invalid Login Credentials
- Input: Incorrect email or password
- Expected Output: Error message indicating unsuccessful login.

Search and Booking:

5. Test Case 5: Search for Routes
- Input: Origin and destination, date, and time
- Expected Output: Display of available routes, including schedules and prices.

6. Test Case 6: Select Route and Seats
- Input: Selection of a route, date, time, and seat preferences
- Expected Output: Reservation of selected seats and calculation of the total fare.

7. Test Case 7: Payment Processing
- Input: Payment information (valid credit card details)
- Expected Output: Successful payment processing and issuance of a booking confirmation.

8. Test Case 8: Seat Availability
- Input: Booking of seats for a route that reaches maximum capacity
- Expected Output: Error message indicating that the seats are no longer available.

User Profile Management:

9. Test Case 9: Edit User Profile
- Input: Changes to user information (e.g., contact details)
- Expected Output: Successful update of user profile information.

10. Test Case 10: Booking History
- Input: Request to view booking history
- Expected Output: Display of a list of past bookings with details.

Admin Panel:

11. Test Case 11: Admin Login
- Input: Admin credentials (username and password)
- Expected Output: Successful login to the admin panel.

12. Test Case 12: Add New Route
- Input: Adding a new route with details
- Expected Output: Successful addition of the route to the system.

13. Test Case 13: Edit Route Details
- Input: Modifying route information (e.g., schedule or price)
- Expected Output: Successful update of route details.

14. Test Case 14: Remove Route
- Input: Deleting an existing route
- Expected Output: Successful removal of the route from the system.

These test cases cover various aspects of a Train and Bus Ticket Booking System, including user interactions, booking processes, and admin panel functionality. Each test case should have well-defined input, expected output, and verification criteria to ensure the system functions correctly and reliably.

CONCLUSION AND FUTURE ENHANCEMENT:

- ✓ We can provide SMS based alerts and email notifications for bookings.
- ✓ The wallet can be developed to handle discounts on the points basis. It can also be used to provide referral code concept for referring friends.
- ✓ Chat feature can be included in the application to chat with customer care.
- ✓ We can come with a feature that will allow the customer to save favourite routes and provide discounts for same.

Introduce coupon management for providing coupons which customers can use to avail discounts. Train and Bus Booking System php project report The main outcome and achievement of this project is to achieve all the objectives. Therefore, Train Ticket is developed as php based system to ease the process of buying the bus ticket in more efficiently and quickly and develop a website for staff and administrator to manage the bus booking system.

Some ideas have been suggested for future improvement of Train Ticket. Train Ticket can be enhanced into a better quality to benefits its users. The suggestion for future improvement on Train Ticket are integration with other online banking system to make the payment gateway more flexible so that user can be more convenient. Other suggestion is Train Ticket should be enhanced to ios platform since the number of user is rapidly increasing.

Taking into account all the mentioned details, we can conclude that the Train Ticket Booking & Management System Project is an inevitable part of the lifecycle of the Travelling. It automates numerous daily operations and enables smooth interactions of the users. Developing the Train management system software is a great opportunity to create distinct, efficient, and fast delivering ticket Bookings. Implementation of Train management system project helps to store all kinds of records, provide coordination and user communication, implement policies.

REFERENCES

1. https://www.phptpoint.com
2. https://code-projects.org
3. https://phpgurukul.com
4. www.youtube.com
5. PHP MySQL Website Programming: Problem - Design – Solution byChrisLea, Mike Buzzard, Dilip Thomas , Jessey White-Cinis
6. Beginning PHP5, Apache, and MySQL Web Development (Programmer to Programmer) by Elizabeth Naramore
7. MySQL/PHP Database Applications, 2nd Edition by Brad Bulger
8. How to Do Everything with PHP and MySQL by Vikram Vaswani