### ONLINE LOCAL TRAIN TICKET BOOKING

**A PROJECT REPORT SUBMITTED TO SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

### IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF COMPUTER APPLICATIONS**

### BY

**PRASATH M**

**REG No : RA2031241040002**

### UNDER THE GUIDANCE OF DR.S.MURUGANANDHAM



**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION COLLEGE OF SCIENCE AND HUMANITIES**

### SRM INSTITUTE OF SCIENCE & TECHNOLOGY

**Vadapalani-600026 Chennai, Tamil Nadu**

**APRIL – 2023**

## BONAFIDE CERTIFICATE

This is to certify that the project report titled **“ONLINE LOCAL TRAIN TICKET BOOKING”** is a bonafide work carried out by **PRASATH M (Reg. No: RA2031241040002)** under my supervision for the award of the Degree of bachelor to my knowledge the work reported herein is the original work done by these student.

### SIGNATURE OF THE GUIDE

Dr.S.Muruganandham Department of Computer Science and Applications, CSH,

SRM Institute of Science And Technology, Vadapalani 600026.

### SIGNATURE OF THE HOD

Dr. J. Padmavathi,Ph.D Associate Professor & Head, Department of Computer Science and Applications, CSH,

SRM Institute of Science And Technology, Vadapalani 600026.

Submitted for Project Work Viva-voce Examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

## ACKNOWLEDGEMENT

First and foremost, I would like to thank heartfelt and with deep sense of gratitude the Management of **SRM Institute of Science & Technology** for their constant support and endorsement.

I wish to express our sincere gratitude to our Dean-In charge, **Dr. Ananthapadmanaban**, College of Science & Humanities for his constant support and encouragement.

I express my gratitude to **Dr. J. Padmavathi,** Associate Professor and Head of Department of Computer Science and Applications**, SRM Institute of Science & Technology** for permitting to do my work in department and provide necessary computational and laboratory facilities to all of us.

I am highly indebted to my guide **Dr.S.Muruganandham**, **Department of Computer Science and Applications,** who generously accepted me under his/her valuable guidance and for the endless help and inspiration provided to me in the tenure of the project.

Special thanks are recorded to our Class-Coordinator **Dr.S.Muruganandham, Department of Computer Science and Applications,** for her constant support and guidance to all of us throughout the project.

Finally, my greatfulness goes to my parents who were my strength and driving force in completion of my project report.

##### PRASATH M

**(Reg. No: RA2031241040002)**

## ABSTRACT

This Project “ ONLINE LOCAL TRAIN TICKET BOOKING” is designed using HTML-PHP as front end and MySQL Server as backend which works in any browsers. The coding language used HTML and PHP. Online Local Train Ticket Booking is very useful nowadays. If the passenger was late to the railway station he/she cannot able to buy ticket(s) when train arrives at platform so he/she after entering into the train he/she can book ticket(s) using Online Local Train Ticket Booking website. If a passenger wants to book a ticket(s), firstly, he/she has to log in to the Online Local Train Ticket Booking website with valid userid and password or you can sign up a new account. Then, the passenger has to select origin station details & destination station details, first class or second class, one-side or return ticket and how many ticket required. Click the pay button to book the ticket using payment gateways. After successful payment of the ticket fare the System will generate the ticket. Passenger click the print button to print the ticket or you can take screenshot. The maintenance controlled by the Admin.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **Bonafide Certificate** | **(i)** |
| **Abstract** | **(ii)** |
| **Acknowledgement** | **(iii)** |
| **List Of Diagrams** | **(iv)** |
| **Table Of Contents** | **(v)** |
| **1** | * 1. **Introduction**   2. **Objective**   3. **Scope of project** |  |
| **2** | **System analysis**   * 1. **Existing System**   2. Proposed **System**   3. **System Requirements** |  |
| **3** | **Software Description**   * 1. **PHP**   2. **Html**   3. **Css**   4. **Javascript**   5. **MySQL** |  |
| **4** | **System Design**   * 1. **Data Flow Diagram**   2. **Use-Case Diagram**   3. **Architecture diagram** |  |

|  |  |  |
| --- | --- | --- |
|  | * + 1. **Functions Of An Architecture Diagram**     2. **How Can Architecture Diagrams Help You?**   1. **Activity diagram**   2. **Class diagram**   3. **Component diagram**   4. **deployment diagram** |  |
| **5** | **Modules**   * 1. **User Registration**   2. **Train Route**   3. **Ticket Module**   **5.4User/Customer Details**   * 1. **Generate QR ticket** |  |
| **6** | **Sample coding**   * 1. **Home Page**   2. **Booking Page** |  |
| **7** | **Software Testing**   * 1. **Introduction**   2. **Strategic Approach To Software Testing**   3. **Unit Testing**      1. **White Box Testing**      2. **Basic Path Testing**      3. **Conditional Testing**      4. **Data Flow Testing** |  |

|  |  |  |
| --- | --- | --- |
|  | **7.3.5 Loop Testing** |  |
| **8** | **System Security**   * 1. **Introduction**   2. **Security software**      1. **Client Side Validation**      2. **Server Side Validation** |  |
| **9** | **Output** |  |
| **10** | **Conclusion** |  |
| **11** | **Reference** |  |

# CHAPTER 1 INTRODUCTION

## CHAPTER 1 INTRODUCTION

### INTRODUCTION:

Online local train ticket booking is a convenient and efficient way for passengers to book tickets for their local train journeys. With the advent of digital technologies, many railway authorities have implemented online ticketing systems to streamline the ticket booking process and improve customer experience. Gone are the days when passengers had to wait in long queues at ticket counters to buy local train tickets. With online ticket booking, passengers can simply log on to the railway website or mobile app, select their destination and travel date, and pay for their tickets using a variety of payment options. This not only saves time but also enables passengers to plan their journeys in advance. Online local train ticket booking systems also benefit railway authorities by enabling them to better manage ticket sales and allocation. The systems can provide valuable data on ticket sales patterns, enabling railway authorities to optimize their pricing strategies and allocate seats more efficiently. This can potentially increase revenue for railway authorities and improve overall efficiency.

Overall, online local train ticket booking is a win-win for both passengers and railway authorities. It simplifies the ticket booking process, improves customer experience, and increases revenue for railway authorities. As digital technologies continue to evolve, we can expect online ticketing systems to become even more efficient and user-friendly in the future.

Top of Form

### OBJECTIVE:

The objective of online local train ticket booking is to provide a convenient and efficient way for passengers to book tickets for their local train journeys. This enables passengers to avoid long queues and save time, while also enabling railway authorities to streamline the ticket booking process.

Some specific objectives of online local train ticket booking might include:

* Improving the customer experience: By offering a user-friendly interface, simple payment options, and quick ticket booking process, passengers can have a better overall experience while booking local train tickets.
* Reducing waiting times: Online ticket booking can reduce the time that passengers have to spend waiting in line at ticket counters, thereby improving overall efficiency.
* Streamlining ticket management: With online ticket booking, railway authorities can better manage ticket sales and allocation, reducing errors and improving overall ticketing efficiency.
* Increasing revenue: Online ticketing systems can potentially increase revenue for railway authorities by allowing them to sell tickets more efficiently and also by providing better data for planning and optimizing ticket sales.

### SCOPE:

* Convenience: Online ticket booking system provides convenience to passengers by allowing them to book tickets from the comfort of their homes or offices. Passengers can easily access the system through a website or mobile app and book their tickets in just a few clicks.
* Time-Saving: Online ticket booking system saves a significant amount of time for passengers by eliminating the need to stand in long queues at ticket counters. This is particularly beneficial during peak travel seasons when ticket counters are overcrowded.
* Improved Revenue: Online ticket booking system can potentially increase revenue for railway authorities by providing better data for planning and optimizing ticket sales.
* Eco-Friendly: Online ticket booking system reduces the need for paper tickets, which can help to reduce environmental impact and promote sustainability.

# CHAPTER 2

# SYSTEM ANALYSIS

## CHAPTER 2 SYSTEM ANALYSIS

### EXISTING SYSTEM:

UTS (Unreserved Ticketing System) is an Indian Railways initiative that allows passengers to book unreserved train tickets using a mobile app. Here are some details about the existing UTS app:

Features: The UTS app provides several features, including booking and cancellation of unreserved train tickets, viewing booking history, and checking train schedules.

* Availability: The UTS app is available for Android and iOS platforms and can be downloaded from the respective app stores.
* Registration: To use the UTS app, passengers need to register with the app using their mobile number and create a login ID and password.
* Payment options: The app supports multiple payment options, including debit cards, credit cards, and net banking.
* Coverage: The UTS app is currently available for select regions in India, including Mumbai, Chennai, Delhi, Kolkata, and Secunderabad.
* QR code-based ticketing: In some regions, the UTS app also supports QR code-based ticketing, where passengers can generate a QR code on their mobile app and use it to enter the train without the need for a physical ticket.
* Ticket validity: Tickets booked through the UTS app are valid for the same day and can only be used for unreserved train travel.

##### DRAWBACKS OF THE EXISTING SYSTEM:

* Ticket booking restrictions: The UTS app only allows passengers to book tickets for the same day, which means that passengers cannot book tickets in advance.
* QR code issues: In some cases, passengers may encounter issues with QR code-based ticketing, such as difficulty in generating or scanning the QR code.
* GPS accuracy issues: The UTS app allows passengers to book ticket at a certain range and you cannot able to book ticket at railway stations.

### PROPOSED SYSTEM:

The proposed system is user can able to book the ticket at any time, anywhere and advance booking allowed. Converted the disadvantage feature into advantage. Now the user can able to book the tickets with no restrictions, no QR code issues and no GPS accuracy issues.

##### ADVANTAGES OF THE PROPOSED SYSTEM:

* Convenience: Online local train ticket booking allows passengers to book tickets from the comfort of their homes or offices, without the need to physically visit a ticket counter or stand in long queues.
* Time-saving: Booking tickets online saves time for passengers, as they can complete the booking process in a matter of minutes.
* Availability: Online ticket booking systems are available 24/7, which means that passengers can book tickets at any time, even outside of the usual ticket counter working hours.

### SYSTEM REQUIREMENTS:

#### HARDWARE REQUIREMENT

Operating System : Windows 7 or Higher

Processor : Intel Core Duo 2.0 GHz or more

RAM : 1 GB or More

Hard disk : 80GB or more

#### SOFTWARE REQUIREMENT

Application : XAMPP or WAMP Server

Front End : PHP,HTML,CSS, Java Script

Back End : MySQL

Editor Tools : Notepad, Visual Studio Code

Web Browser : Google Chrome, Firefox, Brave

# CHAPTER 3 SOFTWARE DESCRIPTION

## CHAPTER 3 SOFTWARE DESCRIPTION

### PHP:

PHP is now officially known as “**PHP: Hypertext Preprocessor**”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL further increases its popularity. Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it. The combination of Apache, MySQL and PHP is all but unbeatable.

That doesn’t mean that PHP cannot work in other environments or with other tools. In fact, PHP supports an extensive list of databases and web-servers. While in the mid-1990s it was ok to build sites, even relatively large sites, with hundreds of individual hard-coded HTML pages, today’s webmasters are making the most of the power of databases to manage their content more effectively and to personalize their sites according to individual user preferences.

**REASONS FOR USING PHP:**

There are some indisputable great reasons to work with PHP. As an open source product, PHP is well supported by a talented production team and a committed user community. Furthermore, PHP can be run on all the major operating systems with most servers.

* + 1. Learning PHP is easy
    2. Its performance
    3. The low cost
    4. It’s Open Source, We can modify it
    5. Its Portability
    6. It has interfaces to a large variety of database systems

You cannot view the PHP source code by selecting “View source” in the browser – you will only see the output from the PHP file, which is plain HTML. This is because the scripts are executed on the server before the result is sent back to the browser.

A PHP scripting block always starts with **<?**php and ends with **?>**. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with <? And end with ?>. However, for maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

* 1. **HTML:**

HTML or **Hyper Text Markup Language** is the standard markup language used to create web pages.HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets(like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to

interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.HTML is descriptive markup language. Library of various markup languages is defined in various browsers.

1. **HTML Images - The <img> Tag and the Src Attribute:**

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for “source”.

The value of the src attribute is the URL of the image you want to display.

1. **HTML FORMS:**

HTML forms are used to pass data to a server. The <form> tag is used to create an HTML form:

<form>

.

*inputelements*

.

</form>

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

1. **IMAGE TAG (<img>):**

To add an image to an HTML document, we just need to include an <IMG> tag with a

reference to the desired image. The <IMG> tag is an empty element i.e. it doesn’t require a closing tag and we can use it to include from small icons to large images.

Syntax: <imgsrc=”URL” alt=”alternative text”>

1. **HTML LISTS :**

HTML5 will be the new standard for HTML. The previous version of HTML, HTML 4.01, came in 1999. The web has changed a lot since then. HTML5 is still a work in progress.

However, the major browsers support many of the new HTML5 elements and APIs. HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

List item

List item

List item







An unordered list:

The third list item

The second list item





An ordered list:

* The first list item

WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established:

* 1. New features should be based on HTML, CSS, DOM, and JavaScript
  2. Reduce the need for external plug-ins (like Flash)
  3. Better error handling
  4. More markup to replace scripting
  5. HTML5 should be device independent
  6. The development process should be visible to the public
  7. **CSS:**

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colors and sizes of text and tables throughout your pages. If

you want to change a certain element you will therefore have to work your way through the document and change it. With CSS you define the colors and sizes in "styles". Then as you write your documents you refer to the styles. Therefore: if you change a certain style it will change the look of your entire site. Another big advantage is that CSS offers much more detailed attributes than plain HTML for defining the look and feel of your site.

* 1. **JAVASCRIPT:**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the

user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

JavaScript was formalized in the ECMA Script language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to objects within a host environment.

JavaScript is the most popular programming language in the world.

It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more.

You can use JavaScript to:

1. Change HTML elements
2. Delete HTML elements
3. Create new HTML elements
4. Copy and clone HTML elements

**3.4 MySQL:**

MySQL is an [open-source](https://en.m.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.m.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius'](https://en.m.wikipedia.org/wiki/Michael_Widenius)s daughter, and "[SQL](https://en.m.wikipedia.org/wiki/SQL)", the

abbreviation for [Structured Query Language](https://en.m.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.m.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.m.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is [free and open-source software](https://en.m.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public](https://en.m.wikipedia.org/wiki/GNU_General_Public_License) [License,](https://en.m.wikipedia.org/wiki/GNU_General_Public_License) and is also available under a variety of [proprietary](https://en.m.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.m.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.m.wikipedia.org/wiki/MySQL_AB), which was bought by [Sun](https://en.m.wikipedia.org/wiki/Sun_Microsystems) [Microsystems](https://en.m.wikipedia.org/wiki/Sun_Microsystems) (now [Oracle Corporation](https://en.m.wikipedia.org/wiki/Oracle_Corporation)). In 2010, when Oracle acquired Sun, Widenius [forked](https://en.m.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.m.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB.](https://en.m.wikipedia.org/wiki/MariaDB)

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the [LAMP](https://en.m.wikipedia.org/wiki/LAMP_(software_bundle)) [web](https://en.m.wikipedia.org/wiki/Web_application) [application](https://en.m.wikipedia.org/wiki/Web_application) [software stack](https://en.m.wikipedia.org/wiki/Software_stack) (and [others](https://en.m.wikipedia.org/wiki/List_of_AMP_packages)), which is an acronym for [*Linux*](https://en.m.wikipedia.org/wiki/Linux)*,* [*Apache,*](https://en.m.wikipedia.org/wiki/Apache_HTTP_Server) *MySQL,* [*Perl*](https://en.m.wikipedia.org/wiki/Perl)*/*[*PHP*](https://en.m.wikipedia.org/wiki/PHP)*/*[*Python*](https://en.m.wikipedia.org/wiki/Python_(programming_language)). MySQL is used by many database-driven web applications, including [Drupal,](https://en.m.wikipedia.org/wiki/Drupal) [Joomla,](https://en.m.wikipedia.org/wiki/Joomla) [phpBB,](https://en.m.wikipedia.org/wiki/PhpBB) and [WordPress](https://en.m.wikipedia.org/wiki/WordPress). MySQL is also used by many popular [websites,](https://en.m.wikipedia.org/wiki/Website) including [Facebook,](https://en.m.wikipedia.org/wiki/Facebook) [Flickr,](https://en.m.wikipedia.org/wiki/Flickr) [MediaWiki](https://en.m.wikipedia.org/wiki/MediaWiki), [Twitter](https://en.m.wikipedia.org/wiki/Twitter), and YouTube. The MySQL server software itself and the client libraries use [dual-licensing](https://en.m.wikipedia.org/wiki/Dual_license) distribution. They are offered under [GPL](https://en.m.wikipedia.org/wiki/GNU_General_Public_License) version 2, or a proprietary license.

1. Cross-platform support
2. [Stored procedures,](https://en.m.wikipedia.org/wiki/Stored_procedure) using a procedural language that closely adheres to [SQL/PSM](https://en.m.wikipedia.org/wiki/SQL/PSM)
3. [Triggers](https://en.m.wikipedia.org/wiki/Database_trigger)
4. [Cursors](https://en.m.wikipedia.org/wiki/Cursor_(databases))
5. Updatable [views](https://en.m.wikipedia.org/wiki/View_(SQL))
6. Online [Data Definition Language](https://en.m.wikipedia.org/wiki/Data_Definition_Language) (DDL) when using the InnoDB Storage Engine.
7. [Information schema](https://en.m.wikipedia.org/wiki/Information_schema)

**CHAPTER 4 SYSTEM DESIGN**

## CHAPTER 4 SYSTEM DESIGN

### DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

It is usually beginning with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1&2 diagram with lower-level functions decomposed from the major functions of the system. Please bear in mind that the level of detail for decomposing a particular function depending on the complexity that function.

HOME

LOGIN

ABOUT

LOGOUT

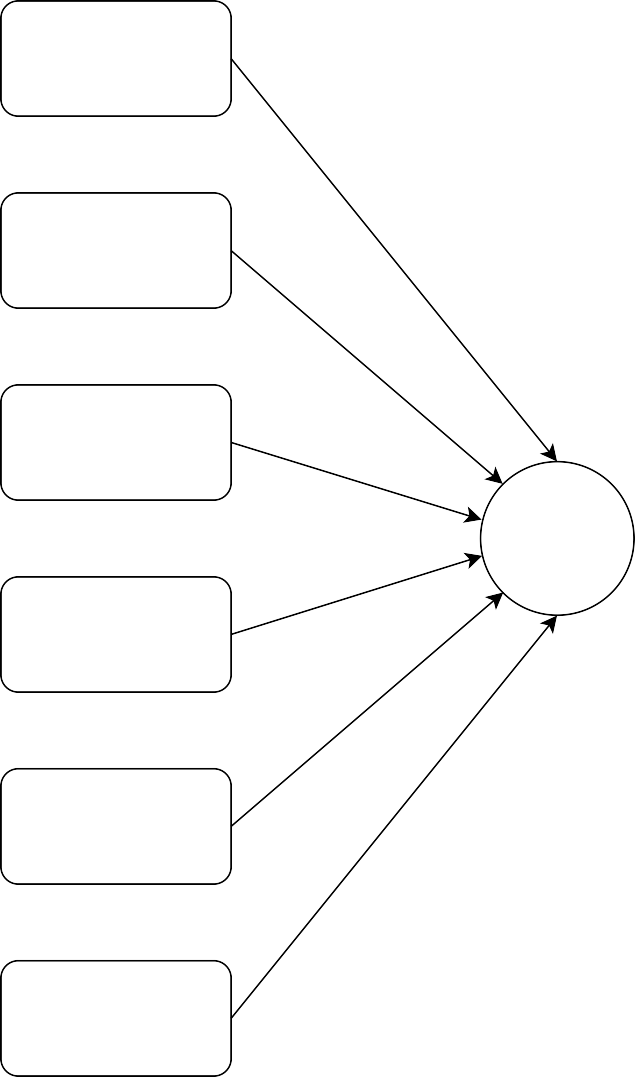
ONLINE LOCAL TRAIN TICKET BOOKING

TRANSACTIONS

TRAIN TIMINGS

BOOK TICKET

Fig 1: 0th level data flow diagram



From

Login

To

Home

Class

About us

BOOK

TICKETS

Type

Train Timeing

No of

Tickets

Transactions

Pay now

Logout

Logout

Tickets

Fig 2: 1st level data flow diagram

Amount

Email id

PAY NOW Payment

Phone no

pay

success

Fig 3: 2nd level data flow diagram – Pay now

Home

Print the page

Payment

Gateway

Pay

Logout

Generate Qr ticket

Fig 4: 2nd level data flow diagram – Payment Gateway

* 1. **USE CASE DIAGRAM:**

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real- world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

* + The boundary, which defines the system of interest in relation to the world around it.
  + The actors, usually individuals involved with the system defined according to their roles.
  + The use cases, which are the specific roles played by the actors within and around the system.
  + The relationships between and among the actors and the use cases.



Fig 10 : Use Case Diagram

### ARCHITECTURE DIAGRAM:

An Architecture diagram is a diagram that depicts a system that people use to abstract the software system's overall outline and build constraints, relations, and boundaries between components. It provides a complete view of the physical deployment of the evolution roadmap of the software system.

### FUNCTIONS OF AN ARCHITECTURE DIAGRAM:

A diagram is similar to a picture. The architecture diagram examples serve various functions. It always helps the relevant users to learn about system architecture and apply it in the decision-making procedures. It is crucial to communicate information regarding architecture. However, people must follow specific steps before making a diagram for architecture. These are:

* + - * Breaking down communication barriers
      * Reaching a consensus
      * Decreasing ambiguity

### HOW CAN ARCHITECTURE DIAGRAMS HELP YOU?

An architecture diagram:

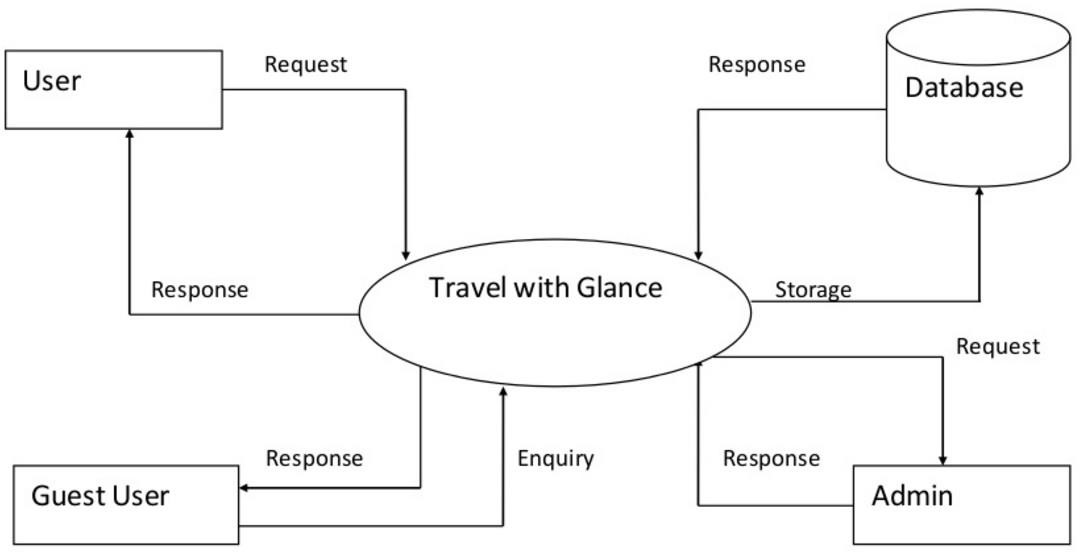
##### Help in making the comprehension process easy

With the help of such diagrams, complex information becomes easy to understand in a single image. Viewers can see the ways things interact. People can also witness the downstream effects.

It makes layers in the place of complex systems.

##### The diagrams help improve collaboration and communication

Consistency is one of the poignant problems that software engineers come across. Several discrepancies and miscommunication may take place between the developers and project teams. The diagrams must follow accuracy, standardization, and detail.



**Online Local Train Ticket Booking**

Fig 11 : Architecture Diagram

* 1. **ACTIVITY DIAGRAM:**

When it comes to a project, the entire project is divided into many interdependent tasks.

In this set of tasks, the sequence or the order of the tasks is quite important.

If the sequence is wrong, the end result of the project might not be what the management expected.

Some tasks in the projects can safely be performed parallel to other tasks. In a project activity diagram, the sequence of the tasks is simply illustrated.

There are many tools that can be used for drawing project activity diagrams. Microsoft Project is one of the most popular software for this type of work.

In addition to that, Microsoft Vision (for Windows) and Omni Graffle (for Mac) can be used to draw activity diagrams.

Have you seen process flow diagrams? If yes, then activity diagrams takes the same shape. Usually there are two main shapes in activity diagrams, boxes and arrows.

Boxes of the activity diagram indicate the tasks and the arrows show the relationships.

Usually, the relationships are the sequences that take place in the activities.

Following is an example of activity diagram with tasks in boxes and relationship represented by arrows.

This type of activity diagram is also known as activity-on-node diagram. This is due to the fact that all activities (tasks) are shown on the nodes (boxes).

Alternatively, there is another way of presenting an activity diagram. This is called activity-on-arrow diagram. In this diagram, activities (tasks) are presented by the arrows.

Compared to activity-on-node diagrams, activity-on-arrow diagrams introduce a little confusion. Therefore, in most instances, people often use activity-on-nodes diagrams. Following is an activity-on-arrow diagram:

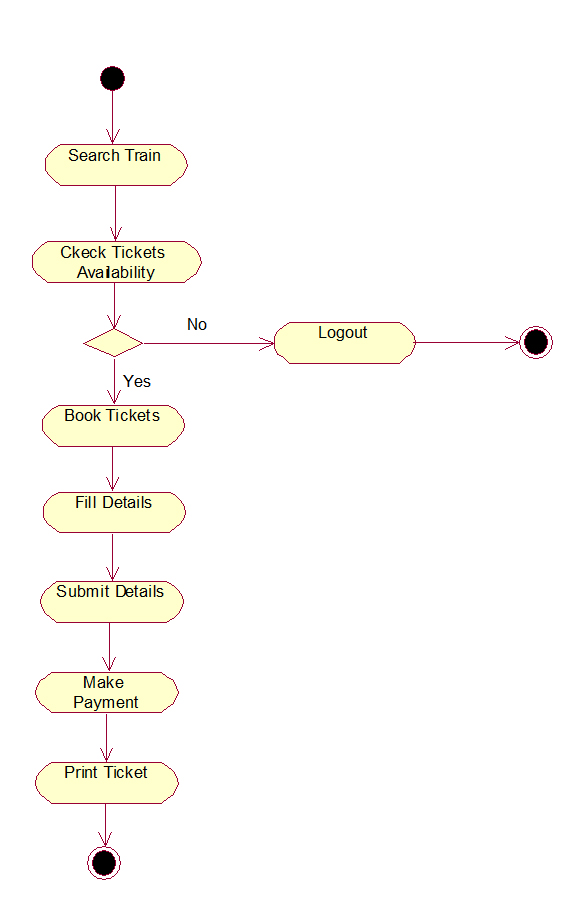


Fig 12 : Activity Diagram

### CLASS DIAGRAM:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object- oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

* Analysis and design of the static view of an application.
* Describe responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering.

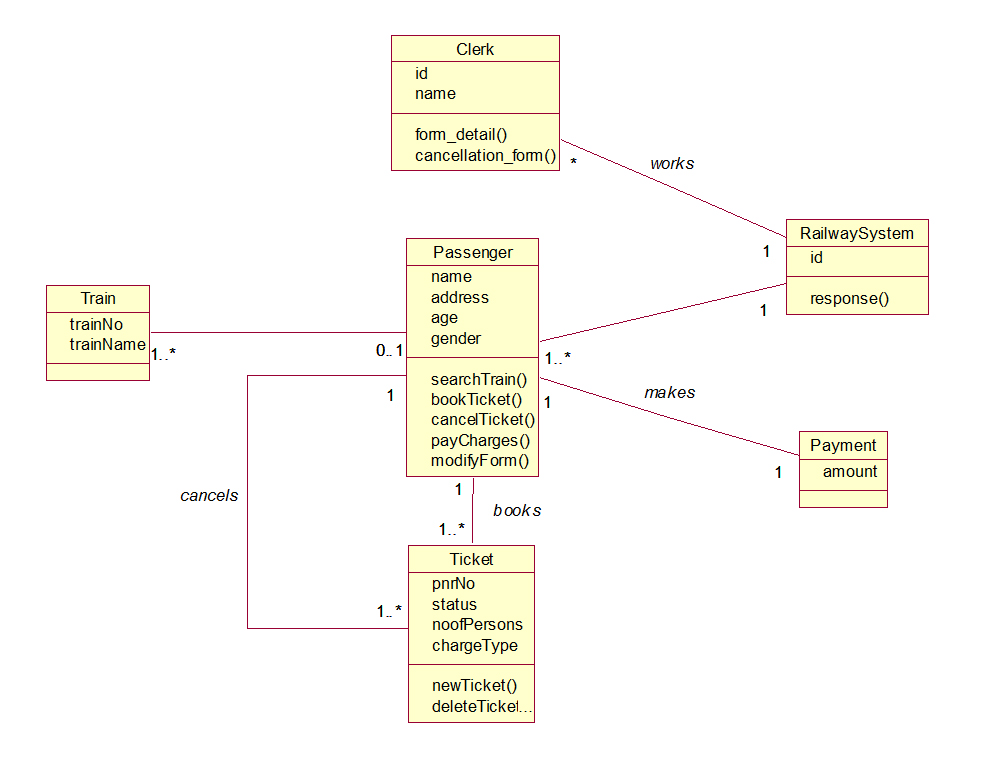


Fig 13 : Class Diagram

### COMPONENT DIAGRAM:

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system.

Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as −

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.

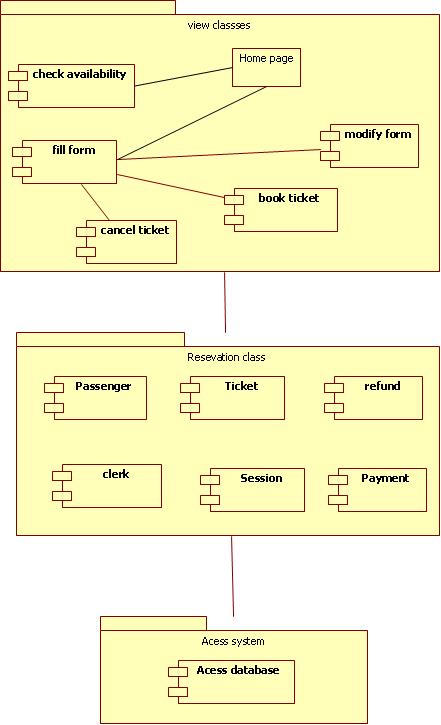


Fig 14 : Component Diagram

### DEPLOYMENT DIAGRAM:

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as −

* Visualize the hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe the runtime processing nodes.

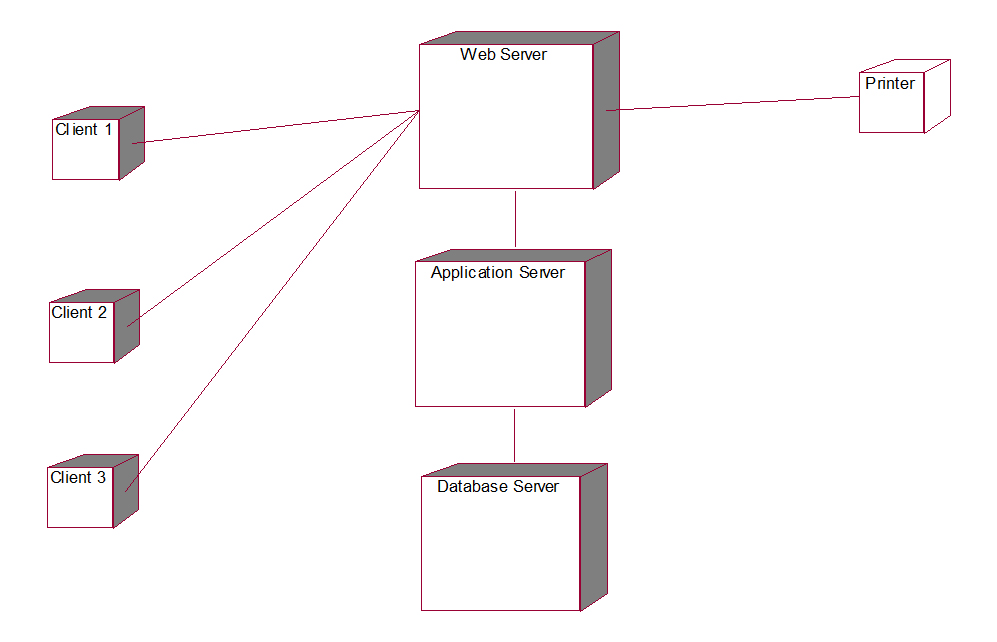


Fig 15 : Deployment Diagram

# CHAPTER 5 MODULES

## CHAPTER 5 MODULES

### MODULES:

* User Registration
* Train Route
* Ticket Module
* User/Customer Details
* Generate QR ticket

### User Registration:

This module covers the details about the registration of users which they can be register by itself by adding data like name, password, email id and further details. After registration they can be sign in by their username and password**.**

### Train Route:

All the operations related to Train Route like source, destination, class, type of journey, no of tickets, ticket prices are managed in MYSQL database by this module.

### Ticket Module:

All the user can able to create a ticket by clicking Book ticket button in homepage & To read a ticket by clicking Transactions button in homepage and delete ticket is managed admin from this module.

### User/Customer Details:

All the user details like Name, Email Id, Password are entered by the users are managed by the admin in MYSQL database. Admin can able to remove the spam users from the database.

### Generate QR ticket:

After the payment successful the user can able to generate the QR ticket and user can able to print the ticket page by this module.

# 

# CHAPTER 6

# SAMPLE CODING

**CHAPTER 6**

**SAMPLE CODING**

### SAMPLE CODING:

* 1. **HOME PAGE:**

<?php

include 'connect.php' ;

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Chennai Railways</title>

<link rel="stylesheet" href="style4.css">

<link rel="shortcut icon" href="logo.png">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<style>

body {

background-image:url("train img3.png");

background-repeat:no-repeat;

background-attachment: fixed;

background-position: center;

background-color:Black;

background-size: cover;

}

</style>

<div class="wrapper">

<div class="title-text">

<div class="title login">Login Form</div>

<div class="title signup">Signup Form</div>

</div>

<div class="form-container">

<div class="slide-controls">

<input type="radio" name="slide" id="login" checked>

<input type="radio" name="slide" id="signup">

<label for="login" class="slide login">Login</label>

<label for="signup" class="slide signup">Signup</label>

<div class="slider-tab"></div>

</div>

<div class="form-inner">

<form action='authenticate.php' method='post' class="login">

<div class="field">

<input type="email" name="email" placeholder="Email Address" required>

</div>

<div class="field">

<input type="password" name="password" placeholder="Password" required>

</div>

<div class="field btn">

<div class="btn-layer"></div>

<input type="submit" name='login\_submit' value="Login">

</div>

<div class="signup-link">Not a member? <a href="">Signup now</a></div>

</form>

<form action='register\_insert.php' method='post' class="signup">

<div class="field">

<input type="Text" name="name" placeholder="Name" required>

</div>

<div class="field">

<input type="email" name="email" placeholder="Email Address" required>

</div>

<div class="field">

<input type="password" name="password" placeholder="Password" required>

</div>

<div class="field btn">

<div class="btn-layer"></div>

<input type="submit" name='register\_submit' value="Signup">

</div>

</form>

</div>

</div>

</div>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Authenticate Page</title>

<style>

.button {

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;

-webkit-transition-duration: 0.4s; /\* Safari \*/

transition-duration: 0.4s;

}

.button {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

.container {

height: 75px;

width:600px;

position: relative;

border: 3px solid green;

}

</style>

<link rel="shortcut icon" href="logo.png">

</head>

<body>

<center><div class="container">

<?php

include 'connect.php' ;

$email = $\_POST['email'];

$password = $\_POST['password'];

$sql\_userdatabase = "Select \* from userdatabase where email = '$email' and password='$password' ";

$result\_userdatabase = mysqli\_query($connect, $sql\_userdatabase);

if(mysqli\_num\_rows($result\_userdatabase) <= 0)

{

echo "<center><h1>Incorrect Email or Password</h1></center>";

echo '<center><table><tr><td><A href="index.php"><button class="button button1">Try Again</button></a></td></tr></table></center>';

}

else

{

header("location:home.php");

$row\_userdatabase = mysqli\_fetch\_array($result\_userdatabase);

session\_start();

$\_SESSION['email']=$email;

$\_SESSION['name']=$row\_userdatabase['Name'];

$\_SESSION['userid']=$row\_userdatabase['UserID'];

$\_SESSION['log']= '1' ;

}

?>

</body>

</html>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<title>Registration Page</title>

<style>

.button {

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;

-webkit-transition-duration: 0.4s; /\* Safari \*/

transition-duration: 0.4s;

}

.button1 {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

.container {

height: 85px;

width:600px;

position: relative;

border: 3px solid green;

}

</style>

<link rel="shortcut icon" href="logo.png">

</head>

<body style="background-color: F5F1F0;">

<center><div class="container">

<?php

include 'connect.php';

$name = $\_POST['name'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$sql\_userdatabase="Insert into userdatabase(Name , Email , password) values ('$name' , '$email' , '$password' )";

if(mysqli\_query($connect, $sql\_userdatabase) == true)

{

echo "<h1>You Have Been Successfully Registered</center></h1><br>";

echo '<center><table><tr><td><A href="index.php"><button class="button button1">Login</button></a></td></tr></table></center>';

}

else

{

echo "<h1><center>Registration Unsucessful , Please try again<center></h1> <br>";

echo "<h2><center><a href='register.php'>Register</a></center";

}

?>

</body>

</html>

<?php

include 'connect.php' ;

session\_start();

if ($\_SESSION['log'] == '')

{

header("location:index.php");

}

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Chennai Railways</title>

<link rel="stylesheet" href="style3.css">

<link rel="shortcut icon" href="logo.png">

<script src="https://kit.fontawesome.com/a076d05399.js"></script>

</head>

<body>

<nav>

<div class="menu">

<div class="logo">

<a href="home.php">CHENNAI RAILWAYS</a>

</div>

<ul>

<li><a href="index.php">LOGIN</a></li>

<li><a href="https://www.metrotraintimings.in/Chennai/Wimco\_Nagar-to-Chennai\_Central-sunday-Local-suburban-MRTS-Train-Timings.htm">TRAIN TIMINGS</a></li>

<li><a href="about.html">ABOUT US</a></li>

<li><a href="logout.php">LOGOUT</a></li>

</ul>

</div>

</nav>

<div class="img"></div>

<div class="center">

<div class="title">Welcome To Chennai Railways</div>

<div class="sub\_title">

<?php

echo "<center>Hello " . $\_SESSION['name'] . "</center>";

?>

</div>

<div class="btns">

<a href="Transac.php"><button>Transactions</button></a>

<a href="book.php" class="bts"><button>Book Tickets</button></a>

</div>

</div>

</body>

</html>

**6.2 BOOKING PAGE:**

<?php

include 'connect.php' ;

session\_start();

if ($\_SESSION['log'] == '')

{

header("location:index.php");

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Ticket Booking Page</title>

<link rel="stylesheet" href="bootstrap.min.css">

<link rel="stylesheet" href="tooplate-style.css">

<link rel="stylesheet" href="style3.css">

<link rel="shortcut icon" href="logo.png">

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700,800" rel="stylesheet">

</head>

<body>

<style>

body {

background-image:url("home.jpg");

background-repeat:no-repeat;

background-attachment: fixed;

background-position: center;

background-color:Black;

background-size: cover;

}

</style>

<nav>

<div class="menu">

<div class="logo">

<a href="home.php">CHENNAI RAILWAYS</a>

</div>

<ul>

<li><a href="index.php">LOGIN</a></li>

<li><a href="https://www.metrotraintimings.in/Chennai/Wimco\_Nagar-to-Chennai\_Central-sunday-Local-suburban-MRTS-Train-Timings.htm">TRAIN TIMINGS</a></li>

<li><a href="about1.html">ABOUT US</a></li>

<li><a href="logout.php">LOGOUT</a></li>

</ul>

</div>

</nav>

<section class="banner" id="top">

<div class="col-md-5 col-md-offset-1">

<section id="first-tab-group" class="tabgroup">

<div id="tab1">

<div class="submit-form">

<h4>BOOK TICKET</h4>

<form action="book\_action.php" method="post">

<div class="row">

<div class="col-md-6">

<fieldset>

<label for="from">From:</label>

<select required name='source' onchange='this.form.()'>

<option value="">Select a location...</option>

<option value="Wimco Nagar">Wimco Nagar</option>

<option value="Tiruvottiyur">Tiruvottiyur</option>

<option value="V.O.C Nagar">V.O.C Nagar</option>

<option value="Tondiarpet">Tondiarpet</option>

<option value="Korukkupet">Korukkupet</option>

<option value="Basin Bridge Junction">Basin Bridge Junction</option>

<option value="Chennai Suburban Terminal">Chennai Suburban Terminal</option>

</select>

</fieldset>

</div>

<div class="col-md-6">

<fieldset>

<label for="to">To:</label>

<select required name='dest' onchange='this.form.()'>

<option value="">Select a location...</option>

<option value="Wimco Nagar">Wimco Nagar</option>

<option value="Tiruvottiyur">Tiruvottiyur</option>

<option value="V.O.C Nagar">V.O.C Nagar</option>

<option value="Tondiarpet">Tondiarpet</option>

<option value="Korukkupet">Korukkupet</option>

<option value="Basin Bridge Junction">Basin Bridge Junction</option>

<option value="Chennai Suburban Terminal">Chennai Suburban Terminal</option>

</select>

</fieldset>

</div>

<div class="col-md-6">

<fieldset>

<label for="departure">Class Type:</label>

<select required name='class' onchange='this.form.()'>

<option value="">Select Class Type...</option>

<option value="1">First Class</option>

<option value="2">Second Class</option>

</select>

</fieldset>

</div>

<div class="col-md-6">

<fieldset>

<label for="return">TICKETS (MAX 6):</label>

<select required name='number' onchange='this.form.()'>

<option value="">Select No of Tickets...</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

</select>

</fieldset>

</div>

<div class="col-md-6">

<div class="radio-select">

<div class="row">

<div class="col-md-6 col-sm-6 col-xs-6">

<label for="round">Return</label>

<input type="radio" name="type" id="round" value="2" required="required"onchange='this.form.()'>

</div>

<div class="col-md-6 col-sm-6 col-xs-6">

<label for="oneway">Oneway</label>

<input type="radio" name="type" id="oneway" value="1" required="required"onchange='this.form.()'>

</div>

</div>

</div>

</div>

<div class="col-md-6">

<fieldset>

<button type="Submit" id="login\_submit" class="btn">Pay Now</button>

</fieldset>

</div>

</div>

</form>

</div>

</div>

</section>

</div>

</div>

</div>

</section>

</form>

</body>

</html>

<?php

include 'connect.php' ;

session\_start();

if ($\_SESSION['log'] == '')

{

header("location:index.php");

}

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Ticket Booking Page</title>

<link rel="stylesheet" href="style3.css">

<link rel="shortcut icon" href="logo.png">

<script src="https://kit.fontawesome.com/a076d05399.js"></script>

</head>

<body>

<nav>

<div class="menu">

<div class="logo">

<a href="home.php">CHENNAI RAILWAYS</a>

</div>

<ul>

<li><a href="index.php">LOGIN</a></li>

<li><a href="about2.html">ABOUT US</a></li>

<li><a href="logout.php">LOGOUT</a></li>

</ul>

</div>

</nav>

<div class="img"></div>

<div class="center">

<div class="sub\_title">

<?php

$source = $\_POST['source'];

$dest = $\_POST['dest'];

$class = $\_POST['class'];

$type = $\_POST['type'];

$no = $\_POST['number'];

if ($source == $dest)

{

echo"<center>Source and destination Selected are Same </center>";

}else {

$sql\_price="SELECT \* FROM `price` WHERE `source` LIKE '$source' AND `dest` LIKE '$dest' AND `class` = $class";

$result = $connect->query($sql\_price);

while($row = $result->fetch\_assoc()){

$final = $row["Price"] \* $type ;

$final=$final\*$no;

echo "<center>You Have To Pay : ₹ ".$final." </center>";

}

$\_SESSION['final']= $final ;

$\_SESSION['source']=$source;

$\_SESSION['dest']=$dest;

$\_SESSION['Class']=$class;

$\_SESSION['Type']=$type;

}

?>

</div>

<div class="btns">

<a href="book.php"><button>Refill Details</button></a>

<a href="https://rzp.io/l/JbEzM7U" class="bts"><button>Proceed</button></a>

</div>

</div>

</body>

</html>

<?php

session\_start();

?>

<html>

<head>

<title>Logout Page</title>

<style>

.button {

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;

-webkit-transition-duration: 0.4s; /\* Safari \*/

transition-duration: 0.4s;

}

.button1 {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

.container {

height: 85px;

width:600px;

position: relative;

border: 3px solid green;

}

</style>

<link rel="shortcut icon" href="logo.png">

</head>

<body style="background-color: F5F1F0;">

<center><div class="container">

<?php

session\_destroy();

echo '<h1>You Have Been Successfully Logged Out<h1></center><br>';

echo '<center><table><tr><td><A href="index.php"><button class="button button1">Start Again</button></a></td></tr></table></center>';

?>

</body style="background-color: F5F1F0;">

</html>

<?php

include 'connect.php';

session\_start();

if ($\_SESSION['log'] == '')

{

header("location:index.php");

}

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Generate Your QR Ticket </title>

<link rel="stylesheet" href="style6.css">

<link rel="stylesheet" href="style2.css">

<link rel="shortcut icon" href="logo.png">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<nav>

<div class="menu">

<div class="logo">

<a href="home.php">CHENNAI RAILWAYS</a>

</div>

<ul>

<li><a href="home.php">HOME</a></li>

<li><a onclick="myFunction()">PRINT THIS PAGE</a></li>

<li><a href="logout.php">LOGOUT</a></li>

</ul>

</div>

</nav>

</head>

<body>

<style>

body {

background-image:url("home.jpg");

background-repeat:no-repeat;

background-attachment: fixed;

background-position: center;

background-color:Black;

background-size: cover;

}

</style>

<script>

function myFunction() {

window.print();

}

</script>

<div class="wrapper">

<header>

<center><img src="tick.gif" alt="tick template"></center>

<center><h1>HAPPY JOURNEY</h1></center>

</header>

<div class="form">

<input type="text" spellcheck="false" value="<?php $name = $\_POST['name'];

$sql\_transactions="INSERT INTO transactions (email,source,dest,name,Class,Type,Amt)VALUES ('".$\_SESSION['email']."','".$\_SESSION['source']."','".$\_SESSION['dest']."','$name','".$\_SESSION['Class']."' ,'".$\_SESSION['Type']."', '".$\_SESSION['final']."')";

if(mysqli\_query($connect, $sql\_transactions) == true)

{

echo " Your Ticket Booked Successfully ";

$sql\_transactions="SELECT \* FROM `transactions` WHERE `source` LIKE '".$\_SESSION['source']."' AND `dest` LIKE '".$\_SESSION['dest']."' AND `Amt` = '".$\_SESSION['final']."' AND `email`='".$\_SESSION['email']."'";

$result = $connect->query($sql\_transactions);

while($row = $result->fetch\_assoc()){

echo "Ticket No. :".$row["T\_No."]." ";

echo "Source Station:".$row["source"]." ";

echo "Destination Station:".$row["dest"]." ";

echo "Class:".$row["Class"]." ";

echo "Type( 1 = one Sided , 2 = Return):".$row["Type"]." ";

echo "No of Passengers : ".$row["Type"]." ";

echo "Amt Paid: ".$row["Amt"]." ";

}

}

else

{

echo "Ticket Booking Unsucessful , Please Try Again <br>";

echo "<a href='Book.php'>Refill</a>";

}

?>">

</input>

<button>Generate Your QR Ticket</button>

</div>

<div class="qr-code">

<img src="" alt="qr-code">

</div>

</div>

<script src="script.js"></script>

</body>

</html>

const wrapper = document.querySelector(".wrapper"),

qrInput = wrapper.querySelector(".form input"),

generateBtn = wrapper.querySelector(".form button"),

qrImg = wrapper.querySelector(".qr-code img");

let preValue;

generateBtn.addEventListener("click", () => {

    let qrValue = qrInput.value.trim();

    if(!qrValue || preValue === qrValue) return;

    preValue = qrValue;

    generateBtn.innerText = "Generating Your QR Ticket...";

    qrImg.src = `https://api.qrserver.com/v1/create-qr-code/?size=200x200&data=${qrValue}`;

    qrImg.addEventListener("load", () => {

        wrapper.classList.add("active");

        generateBtn.innerText = "Your QR Ticket";

    });

});

qrInput.addEventListener("keyup", () => {

    if(!qrInput.value.trim()) {

        wrapper.classList.remove("active");

        preValue = "";

    }

});

@import url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;600&display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: "Poppins", sans-serif;

}

.center{

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 400px;

background: white;

border-radius: 10px;

box-shadow: 10px 10px 15px rgba(0,0,0,0.05);

}

.center h1{

text-align: center;

padding: 20px 0;

border-bottom: 1px solid silver;

}

.center form{

padding: 0 40px;

box-sizing: border-box;

}

form .txt\_field{

position: relative;

border-bottom: 2px solid #adadad;

margin: 30px 0;

}

.txt\_field input{

width: 100%;

padding: 0 5px;

height: 40px;

font-size: 16px;

border: none;

background: none;

outline: none;

}

.txt\_field label{

position: absolute;

top: 50%;

left: 5px;

color: #adadad;

transform: translateY(-50%);

font-size: 16px;

pointer-events: none;

transition: .5s;

}

.txt\_field span::before{

content: '';

position: absolute;

top: 40px;

left: 0;

width: 0%;

height: 2px;

background: #2691d9;

transition: .5s;

}

.txt\_field input:focus ~ label,

.txt\_field input:valid ~ label{

top: -5px;

color: #2691d9;

}

.txt\_field input:focus ~ span::before,

.txt\_field input:valid ~ span::before{

width: 100%;

}

.pass{

margin: -5px 0 20px 5px;

color: #a6a6a6;

cursor: pointer;

}

.pass:hover{

text-decoration: underline;

}

input[type="submit"]{

width: 100%;

height: 50px;

border: 1px solid;

background: #2691d9;

border-radius: 25px;

font-size: 18px;

color: #e9f4fb;

font-weight: 700;

cursor: pointer;

outline: none;

}

input[type="submit"]:hover{

border-color: #2691d9;

transition: .5s;

}

.signup\_link{

margin: 30px 0;

text-align: center;

font-size: 16px;

color: #666666;

}

.signup\_link a{

color: #2691d9;

text-decoration: none;

}

.signup\_link a:hover{

text-decoration: underline;

}

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ABOUT US</title>

<link rel="stylesheet" href="style3.css">

<link rel="shortcut icon" href="logo.png">

<script src="https://kit.fontawesome.com/a076d05399.js"></script>

</head>

<body>

<nav>

<div class="menu">

<div class="logo">

<a href="index.php">CHENNAI RAILWAYS</a>

</div>

<ul>

<li><a href="home.php">HOME</a></li>

<li><a href="logout.php">LOGOUT</a></li>

</ul>

</div>

</nav>

<div class="center">

<h1> ABOUT US </h1>

<h3>The Project was Done by PRASATH M 3rd BCA "B" Section at SRM Institute Of Science And Technology, Vdp Campus. Chennai Railways Website is a user-friendly portal which can used to book your local train tickets from anywhere. Online local train ticket booking system project for chennai local trains that allows users to book local train tickets and get QRticket online. This local train project provides login rights for normal users and admin. A normal user may login and get a ticket online, print it and travel by train. The ticketing process consists of a ticket booking form. The form allows the user to choose his source and destination. The source is the station from where the user will be boarding the train. Destination is the station he needs to get down at. The project database is already filled with stations from Wimco Nagar to Chennai Suburban Terminal.The system also consists of an option to select weather ticket is a single journey or a return ticket and the journey will be commenced on first class or a second class.You can able to book upto 6 ticket at a time</h3>

<script>

.body{

background-color:white;

}

</script>

</div>

</body>

</html>

<?php

$hostname = 'localhost';

$username = 'root';

$password='';

$dbname = 'LocalTrainTicketBookingSystem';

$connect = mysqli\_connect($hostname , $username , $password ,$dbname) or die("Error Connecting");

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<title>Transaction History</title>

<link rel="shortcut icon" href="logo.png">

<style>

.button {

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;

-webkit-transition-duration: 0.4s; /\* Safari \*/

transition-duration: 0.4s;

}

.center {

margin: 0;

position: absolute;

top: 50%;

left: 50%;

-ms-transform: translate(-50%, -50%);

transform: translate(-50%, -50%);

}

.container {

height: 75px;

width:400px;

position: relative;

border: 3px solid green;

}

.button1 {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

.button2 {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

</style>

<script>

function myFunction() {

window.print();

}

</script>

</head>

<body style="background-color: white;">

<div class="center">

<div class="container">

<center><h1>TICKET HISTORY</h1></center>

</div>

<?php

include 'connect.php';

session\_start();

if ($\_SESSION['log'] == '')

{

header("location:index.php");

}

$sql\_transactions="SELECT \* FROM `transactions` WHERE `email`='".$\_SESSION['email']."'";

$result = $connect->query($sql\_transactions);

while($row = $result->fetch\_assoc()){

echo "<center><h1>Ticket</h1></center>";

echo "<center>Ticket No. :".$row["T\_No."]."</center>";

echo "<center>Source:".$row["source"]."</center>";

echo "<center>Destination:".$row["dest"]."</center>";

echo "<center>Class:".$row["Class"]."</center>";

echo "<center>Type( 1 = one Sided , 2 = Return):".$row["Type"]."</center>";

echo "<center>No of Passengers : ".$row["Type"]."</center>";

echo "<center>Amt Paid: ".$row["Amt"]."</center>";

}

?>

<center><button class="button button1" onclick="myFunction()">Print this page</button>

<a href="home.php"><button class="button button2">Home</button></a></center>

</body>

</html>

# 

# CHAPTER 7

# SOFTWARE TESTING

* 1. **INTRODUCTION:**

**CHAPTER 7 SOFTWARE TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well- planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

### STRATEGIC APPROACH TO SOFTWARE TESTING:

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

### UNIT TESTING:

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

### WHITE BOX TESTING :

This type of testing ensures that

* + - * All independent paths have been exercised at least once
      * All logical decisions have been exercised on their true and false sides
      * All loops are executed at their boundaries and within their operational bounds
      * All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

### BASIC PATH TESTING :

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula: V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity, E is the number of edges,

N is the number of flow graph nodes, P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

### CONDITIONAL TESTING :

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

### DATA FLOW TESTING :

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

### LOOP TESTING :

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

All the loops were tested at their limits, just above them and just below them. All the loops were skipped at least once.

For nested loops test the inner most loop first and then work outwards.

For concatenated loops the values of dependent loops were set with the help of connected loop.

Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

# CHAPTER 8 SYSTEM SECURITY

### CHAPTER 8 SYSTEM SECURITY

* 1. **INTRODUCTION:**

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

### SECURITY SOFTWARE :

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employees two types of checks and controls:

### CLIENT SIDE VALIDATION:

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

* JavaScript in used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
* Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
* Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

### SERVER SIDE VALIDATION:

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

* Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
* User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
* Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
* Using server side validation, constraints on several restricted operations are imposed.

# CHAPTER 9 OUTPUT

### CHAPTER 9

### OUTPUT

### LOGIN SCREEN:

### 

### HOME SCREEN:

### 

### BOOKING SCREEN:

### 

### BOOK ACTION SCREEN:

### 

### PAYMENT GATEWAY SCREEN:

### 

### GENERATE QR TICKET SCREEN:

### 

### 

### 

# CHAPTER 10 CONCLUSION

**CONCLUSION:**

**CHAPTER 10 CONCLUSION**

This web application was successfully created and stored all the user details, transaction details, ticket price details into the database using this application. Online Local Train Ticket Booking system proves to be a strong system which has followed all the industrial standards .The application was tested very well and the errors were properly debugged. Testing also concluded that the performance of the system is satisfactory. All the necessary output is generated. This system thus provides an easy way to automate all the functionalities of consumption. If this application is implemented in few consumption, it will be helpful. Further enhancements can be made to the project, so that the website functions in a very attractive and useful manner than the present one. It is concluded that the application works well and satisfy the needs. The application is tested very well and errors are properly debugged. It also acts as the sharing of files to the valuable resources.

# CHAPTER 11 REFERENCE

### FOR PHP:

**CHAPTER 11 REFERENCE**

* + <https://www.w3schools.com/php/default.asp>
  + <https://www.sitepoint.com/php/>
  + <https://www.php.net/>

### FOR MySQL:

* + <https://www.mysql.com/>
  + [http://www.mysqltutorial.org](http://www.mysqltutorial.org/)

### FOR XAMPP:

* + <https://www.apachefriends.org/download.html>