



East West University

Spring-2019

Project Report

# Railway Online Booking System Design and Implementation

Submitted by:

Md. Asik Iqbal	2015-1-55-003
Tridib Biswas	2015-1-55-021
Rokebul Ehsan	2015-1-55-027

This paper is an essential fragmentary of the total completeness of the degree of Bachelor Science in Electronics and Telecommunications Engineering at East West University.

Project Supervisor

Dr. Anup Kumar Paul  
Assistant Professor

Department of Electronics and Communications  
Engineering

Chairperson

Dr. Mohammed Moseur Rahman  
Assistant Professor & Chairperson

Department of Electronics and  
Communications Engineering

## *Declaration*

We hereby declare that we complete our project on the topic entitle Railway Online Booking System Design and Implementation. We prepared the project report and submitted to the Department of Electronics and Communications Engineering. This project is submitted to fulfil the requirement the degree of B.Sc. in Information and Communications Engineering.

We claim that project work which is demonstrated in this report is out own work. We also declare that this work has not been submitted anywhere for publication.

---

Md. Asik Iqbal  
2015-1-55-003

---

Tridib Biswas  
2015-1-55-021

---

Rokebul Ehsan  
2015-1-55-027

### Signature of Supervisor

---

Dr. Anup Kumar Paul  
Assistant Professor

Department of Electronics and Communications  
Engineering

### Signature of Chairperson

---

Dr. Mohammed Moseeur Rahman  
Assistant Professor & Chairperson

Department of Electronics and  
Communications Engineering

## *Abstract*

Our designed Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, Tracking the current position etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passenger. It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before.

In our country, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model. The design of the railway reservation database system is also based on the relation model.

# Railway Online Booking System Design and Implementation

## *Letter of Acceptance*

This Project entitled by “Railway Online Booking System Design and Implementation” submitted by Md. Asik Iqbal (2015-1-55-003), Tridib Biswas (2015-1-55-021), Rokebul Ehsan (2015-1-55-027) to the Department of Electronics and Communications Engineering, East West University, Dhaka, Bangladesh is granted by the dept. in some achievement of assertion for the Reward of the Degree of Bachelor of Science in Information and Communications Engineering in April 2019.

---

Date

---

Supervisor  
Dr. Anup Kumar Paul  
Assistant Professor  
Department of Electronics and Communications Engineering  
East West University, Dhaka, Bangladesh

## *Acknowledgement*

We would like to express our sincere thanks to our project head *Assistant Professor Dr. Anup Kumar Paul* for his endeavor help and cooperation in completion of the project. He constantly encouraged me with his valuable suggestions throughout the project lifecycle.

We would also like to thank faculty members of our university for their kind interest at various points to follow the project lifecycle.

We wish to thank the great people of department of Electronics and Communications Engineering at East West University.

## *Table of Contents*

CHAPTER-01 .....	1
Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Objective .....	2
CHAPTER-02.....	3
2.1 System Overview .....	3
2.2 System Function.....	4
2.3 System design .....	5
2.4 Database design .....	6
2.5 ER Model.....	6
2.6 Modification and Implementation over existing model .....	8
CHAPTER-03.....	9
Requirements .....	9
3.1 Case Diagram.....	9
3.2 Activity Diagram .....	10
3.3 System Requirements.....	10
1.Web Server (Apache Server (XAMPP) Version: 3.2.2 and above .....	10
2. Database Management System (MySQL):.....	11
3. Server-side Language (PHP).....	11
4. HTML .....	11
5. CSS .....	11
6. BOOTSTRAP .....	12
7. JavaScript.....	12
8. JQuery .....	12
9. Version Control Software (GitHub).....	12
10. Text Editor (Brackets).....	12
11. Browser:.....	12
CHAPTER-04.....	13
Functionalities.....	13
4.1 Registration and login: .....	13
4.1.1: Registration.....	13
4.1.2: Login.....	13

4.1.3 Password Recovery .....	14
4.2 Home Page and Booking tickets .....	14
4.2.1 Home page .....	14
4.2.2 Book ticket .....	15
4.2.3 Seat Selection .....	15
4.2.4 Invoice.....	16
4.2.5 Payment Method .....	16
4.3 Cancelling Ticket .....	17
4.3.1 Cancel the Ticket .....	17
4.3.2 Confirmation .....	17
4.3.3 Confirmation Page: .....	18
4.4. Route Search: .....	18
4.4.2 Available Trains.....	18
4.5 Get Location.....	19
4.5.1 Location of train.....	20
4.6 Ticket details.....	20
4.6.1 Entering mail address.....	20
4.6.2 Ticket details: .....	21
4.7 Your Profile .....	21
4.8 Logged in user.....	22
CHAPTER-05 .....	23
Database Management .....	23
Database Design: .....	23
5.1 Database Tables .....	23
5.2 List of Tables: .....	23
5.2.1 Passenger Table.....	24
5.2.2 Ticket Table: .....	24
5.2.3 Train Table.....	25
5.2.4 Users Table .....	25
CHAPTER-06.....	26
Specification of code.....	26
6.1 Index.php .....	26
6.2 Search route .....	34
6.3 Get location.....	36
6.4 Delete reservation: .....	38
6.5 Contact.php .....	39

6.6 Database search.....	41
6.7 DB config:.....	42
CHAPTER-07.....	43
Conclusion .....	43
7.1 Future Aspects .....	43
7.2 Conclusions.....	43
Appendix.....	44
Acronyms.....	44
Reference: .....	45



# ***CHAPTER-01***

## **Introduction**

Rail transport is one of the most important means of transport, has played an important role in the transport industry in Bangladesh. With rapid economic development, the railway lines and passengers have been increasing year by year in the country. With such a huge customer base, buying train tickets problem has been very prominent. The electronic commerce could solve the problem of railway ticketing. Introduced a new online ticketing system is not only technological innovation, but also will improve the railway services, to a certain extent, solve the difficult problem of railway ticketing.

### **1.1 Purpose**

The purpose of this project is to describe the railway reservation system which provides details for the railway reservation system like the train timing details, reservation, billing and cancellation on various types of reservations namely,

- 1) Confirm reservation for confirm sent
- 2) Reservation against Cancellation
- 3) Online reservation

### **1.2 Scope**

Technology has transformed many aspects of life in the 21st century, including the way many of us make train reservations. For example, to make ticketing more convenient for travelers, an online reservation system helps us in booking tickets from the comfort of our homes or offices. While this is convenient for most people, it has made things particularly easier for people residing in remote locations.

This project has a substantially large scope. It includes the following features which help it easy to use, understand and modify it:

- Automation of Reservation Status.
- No need to do paper work.
- To save the environment by using paper free processing
- To increase efficiency and accuracy of the software

- Management of online database
- Management of online payment

This web package can be used by non-programming personal avoiding human handled chance of error. This project is used by two types of users:

Railway Administrator

- Authorized railway reservation counters

Main points are:

- Simplified management of passengers
- Can be used online
- Online payment system
- Human friendly interface

The various advantages of using the online reservation system are as follows:

- 1) Convenient – You can book or cancel your tickets sitting in the comfort of your home or office.
- 2) Saves Time and Effort - You can save the time needed to travel to the railway reservation office and waiting in the queue for your turn.
- 3) Towards a greener planet – Instead of printing your ticket you can also choose to travel with the soft copy of your booked ticket in your laptop or even on your mobile Freight Revenue enhancement.
- 4) Passenger Revenue enhancement. Improved & optimized service.

### **1.3 Objective**

User can visit the page and buy ticket according to their choice via online payment. The information of user will be collected and saved in the database for the purpose of safety and reliability. Administrator has the authority to add, delete or edit any information of use or any data in the database.

## CHAPTER-02

### 2.1 System Overview

A typical three-layer structure is used in the system: the database layer, the application service layer, the user interface layer. System architecture as shown in Figure 1.

**The database layer** - The database is used to hold data, including user registration information, ticket ordering information, ticket information and all of the other information.

**The application service layer** - The application service layer is the core of this three-layer structure, the system functions and business logic are handled in this layer. In this layer, the system's business logic is encapsulated, the application service interfaces is provided for the user interface layer and the system modules between the function calls. The application service layer also updates data in the database, according to the service request of the top layer.

**The user interface layer** - The user interface layer is a program that runs on a remote user computer. It displays the provided services by the server to the user. When the user selects a service, this program sends request to the server. When the server returns the processed result, this program shows it to the user.

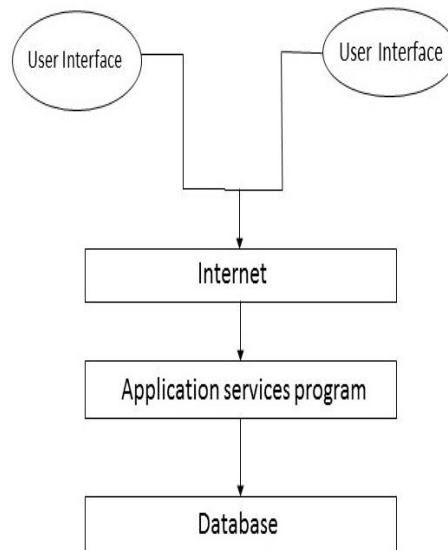


Figure-1.1: System Overview.

## 2.2 System Function

The system functions are as following: customer register function, customer cancellation function, searching function, booking function, refunding function. Functional blocks of the system as shown in Figure 2. Figure 2 Functional blocks of the system:

**Customer register function:** Customers could order tickets after they login on the system. The registration information includes user's name, user's telephone, user's email and user's address.

**Customer cancellation function:** According to the customer's cancellation request, the system could complete the cancelling operation of the ticket by his/her ticket ID.

**Searching function:** Customers can use the train route to site to query. When they type the information, the system will return the relevant information to customers, including the type of train, the type of the tickets, departure time, capacity, current location of the train and the price of tickets.

**Booking function:** Customers could order tickets after they login on the system. They could select the Station From, Station To, Train name, class selection, seat number and coach number.

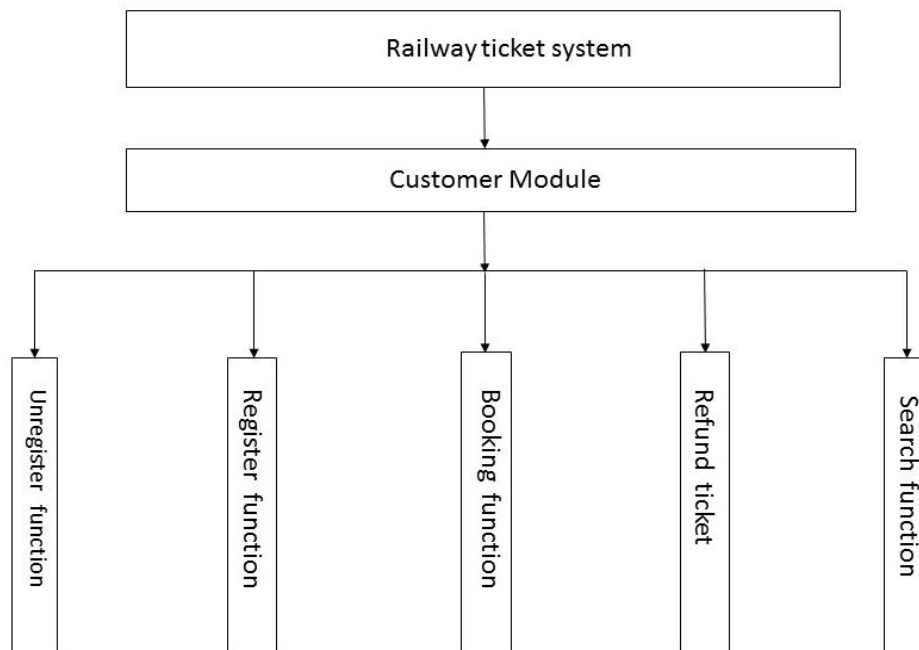


Figure-2.1: System Function.

**Location of the Train:** Maintaining time schedule of Train is not reliable in Bangladesh. Passenger have to wait long time in the station for the train without any knowledge when the train will actually arrive. From location customer can easily find the location of the train.

## 2.3 System design

The system design includes business process design and database design. Business process design implements the order of various functions and links between the various functions. Database design mainly realizes data tables and the relationship between data tables.

- **Business process design** -The following business process is made after a detailed analysis of business functions about railway online booking system. Customers register personal information, so they can order tickets in the system. Customers search train information through the system and see whether having appropriate tickets. Customers order tickets on the user interface. The system returns the result of ordering tickets information. Customers can select cancelling operation for some reasons, so the personal information will be removed from the system. The systems business process chart is shown in figure below:

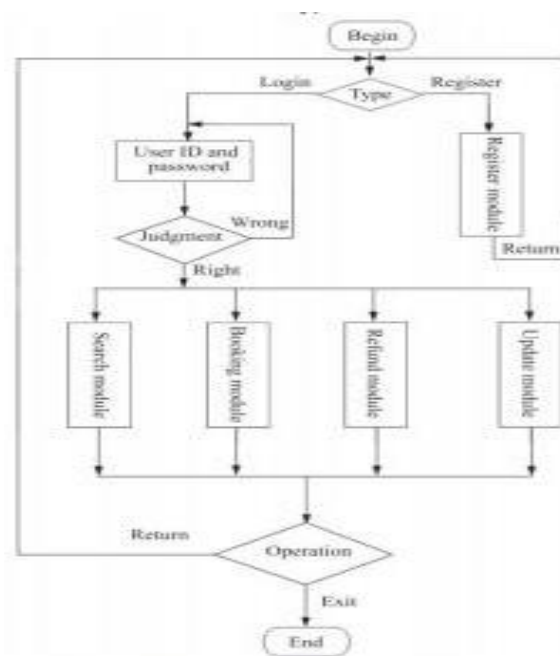


Figure-2.2: System Design.

## 2.4 Database design

Database Conceptual Design -In the database conceptual design phase, the entity - relationship diagram can be used to establish the data model to form an ER graph model independent of the machine and DBMS. ER diagram provides entities (ie data object), properties, and associated methods, in order to describing the conceptual model of the real world. The basic elements of ER diagram are constituted by entities, attributes and links. The representation is as following: Entity type: It is expressed by rectangle. The entity name is written in the rectangle. Attribute: It is expressed by oval and linked with the corresponding entity by line; multi-valued attributes is connected by the two-lane; line is added under the name of the main attribute. Contact: It is expressed by the diamond and linked with the corresponding entity by line. The contact name is written in the diamond box. The contact type is written near by the line. The relationship of 1 to 1, 1 to n, n to m should be clearly shown in the ER diagram. 1 to 1 relationship between the two entities connected in the direction of writing 1; 1 to n relations in the one to writing 1 and multi-party write N; n to m relationship between the two entities are connected in the direction of writing N, M.

## 2.5 ER Model

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.[1]

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here’s a glossary:

- Entity

A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

- Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.

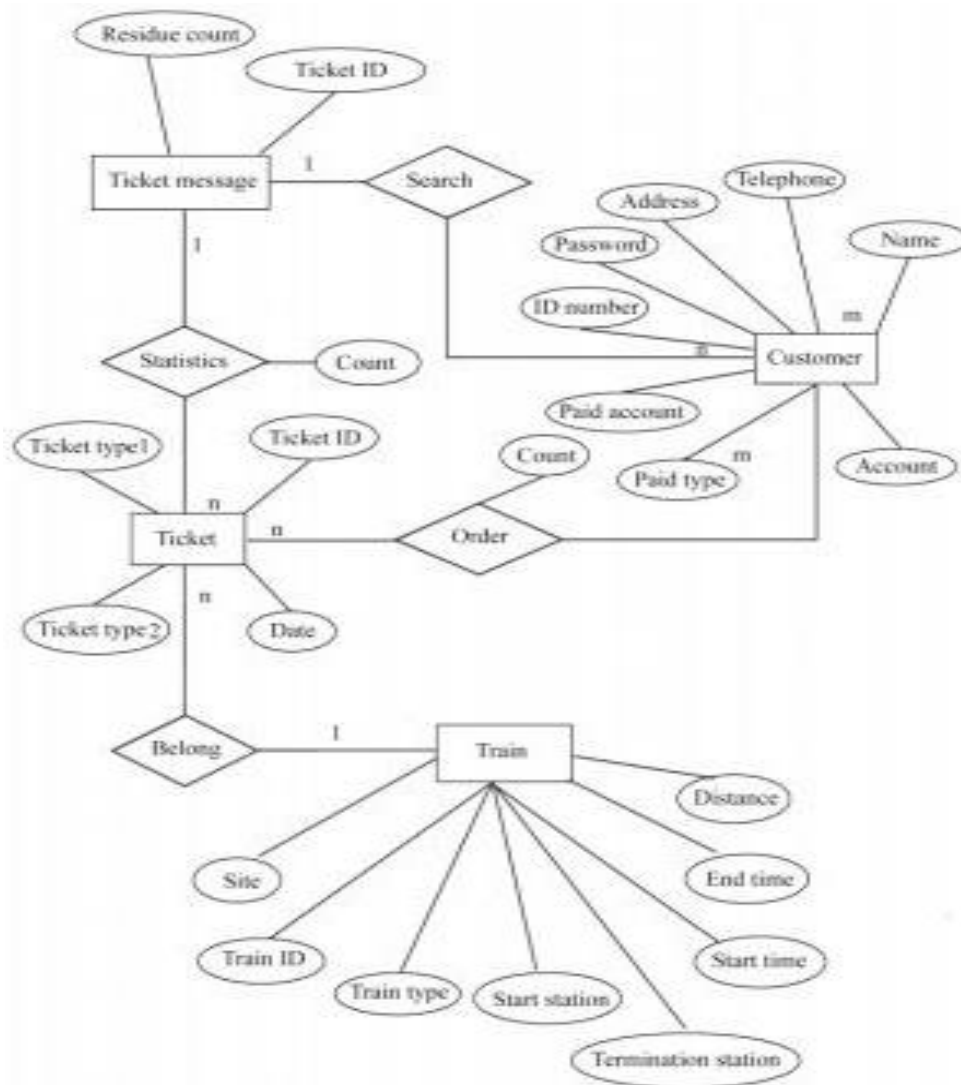


Figure-2.3: ER diagram.

## 2.6 Modification and Implementation over existing model

Present Site:

- 1) Currently used website lacks human friendly interface.
- 2) Time consumed in accessing the records of the database.
- 3) User has to wait for a long time in the queue to get the reservation.
- 4) Facilities of the software cannot be accessed online/other than department systems.
- 5) Have to wait for the seat selection.
- 6) No bKash payment system available in the present website.

After implementation of project:

- 1) Easy to use GUI.
- 2) The software will be accessed online by Administrator for authorized data, by reservation counters for reservation and by passenger for search for trains from a particular source to destination.
- 3) New features as per requirements.
- 4) Customers will know the live location of the train.
- 5) Customer will be select their seat from the seat plan immediately.
- 6) Customer will be able to pay their bill via bKash.



## CHAPTER-03

### Requirements

#### 3.1 Case Diagram

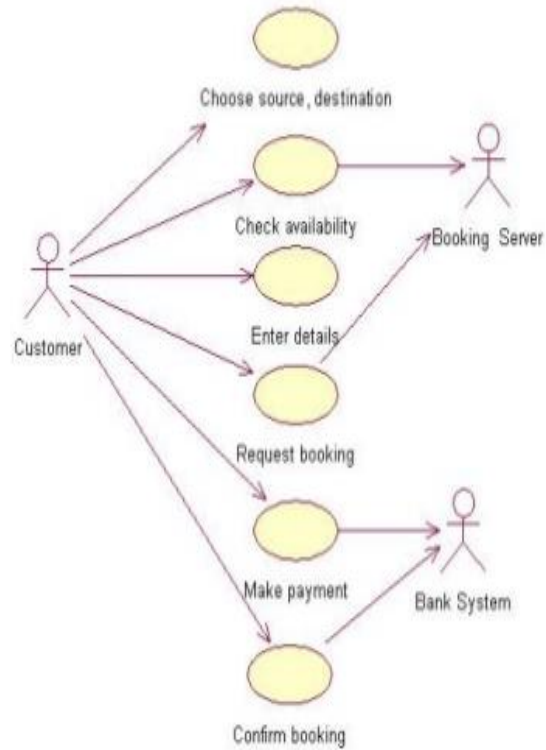


Figure-3.1: Case Diagram

## 3.2 Activity Diagram

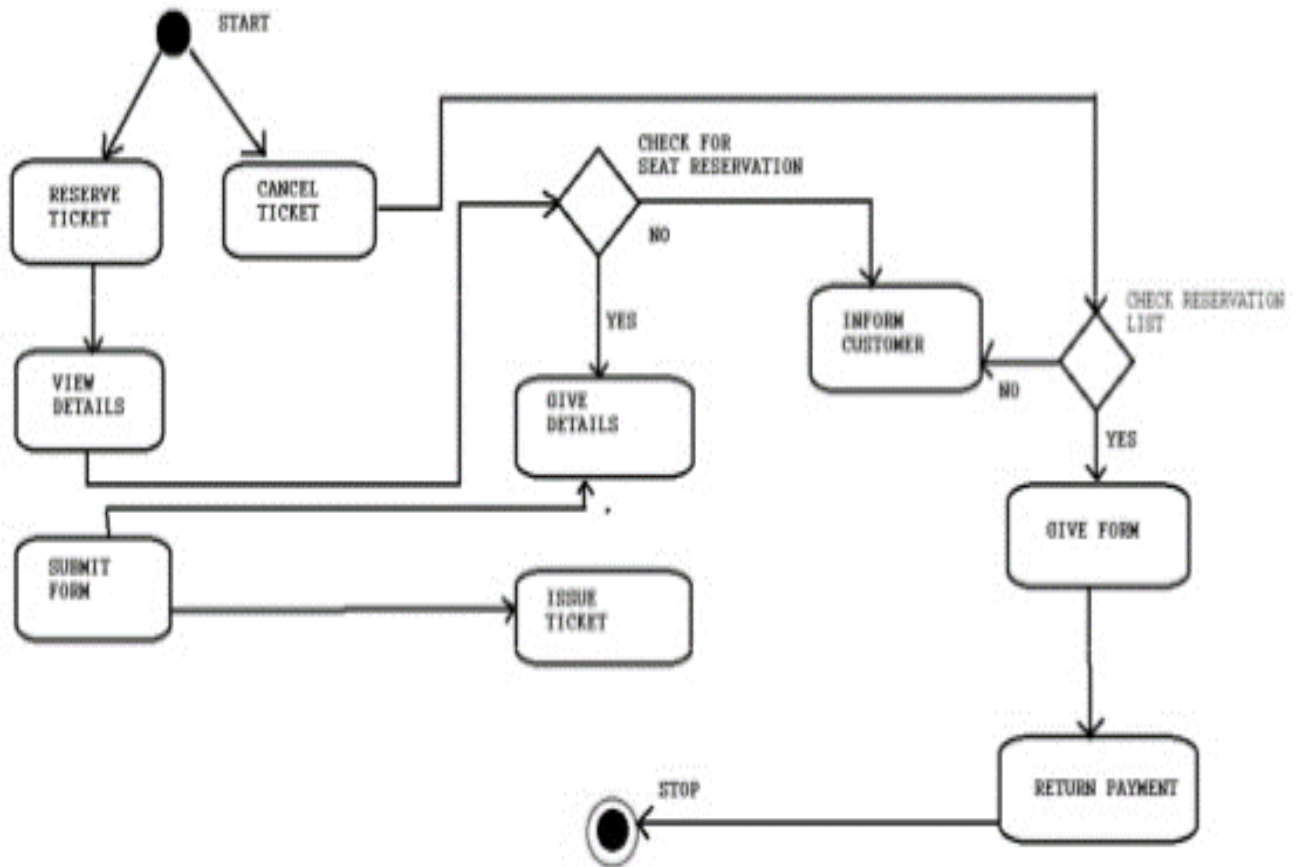


Figure-3.2: Activity Diagram

## 3.3 System Requirements

### 1. Web Server (Apache Server (XAMPP) Version: 3.2.2 and above:

Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all web servers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules. Most Word Press hosting providers use Apache as their web server software.[2]

## **2. Database Management System (MySQL):**

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing Content in a database. it is most noted for its quick processing, proven reliability, ease and flexibility of use.[3]

## **3. Server-side Language (PHP)**

PHP The most popular server-side language on the web, PHP is designed to pull and edit information in the database. Its most commonly bundled with databases written in the SQL language. PHP was designed strictly for the web and remains one of the most widely used languages around. It's easy to install and deploy, is staying competitive with lots of modern frameworks, and is the foundation for a number of content-management systems. PHP-powered sites: Word Press, Wikipedia, Facebook.[4]

## **4. HTML**

First developed by Tim Berner Lee in 1990, HTML is short for Hyper Text Markup Language HTML is used to create electronic documents (called pages) that are displayed on the World wide web. Each page contains a series of connections to other pages called hyperlinks. Every web page seen on the Internet is written using one version of HTML code or another. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style sheet are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).[5]

## **5. CSS**

CSS is a language that describes the style of an HTML document.CSS describes how HTML elements should be displayed. It is much easier to maintain the appearance of a Web site. If one uses a single CSS file updating this file allows the Web site look-and-feel to be altered easily; in contrast use of HTML formatting elements would require every file to be updated to change the appearance. CSS provides rich functionality, including defining the appearance of HTML pages when they are printed. Use of CSS provides much greater accessibility, allowing users with special needs to alter the appearance of a Web page to suit their requirements. CSS also allows Web pages to be more easily rendered by special devices, such as speaking browsers.[6]

## **6. BOOTSTRAP**

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.[7]

## **7. JavaScript**

The most common use of JavaScript is to add Dynamic HTML (DHTML). Scripts are embedded in Document Object Model (DOM) of the page.[8]

## **8. JQuery**

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.[9]

## **9. Version Control Software (GitHub)**

GitHub is a development platform inspired by the way anyone work. From open to Business, one can host and review code, manage projects, and build software alongside millions of other developers. [10]

## **10. Text Editor (Brackets)**

Brackets is a open source text editor that lets one efficiently design, develop and maintain standard based websites and applications. Bracket provides a powerful combination of visual layout tools application, development features and code editing support. [11]

## **11. Browser:**

Any browser can open and support the server and website.[12]

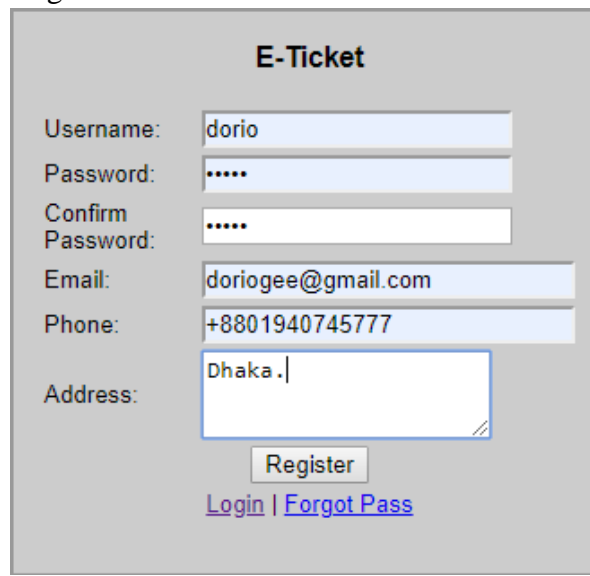
## CHAPTER-04

### Functionalities

#### 4.1 Registration and login:

##### 4.1.1: Registration

We have to get a login ID and Password to get access to the website. So we have to fill this registration form in order to get the access and able to Book tickets.



The screenshot shows a registration form titled "E-Ticket" on a grey background. The form contains the following fields and values:

- Username: dorio
- Password: \*\*\*\*\*
- Confirm Password: \*\*\*\*\*
- Email: doriogee@gmail.com
- Phone: +8801940745777
- Address: Dhaka.

Below the fields is a "Register" button and two links: "Login" and "Forgot Pass".

Figure 4.1: Registration

##### 4.1.2: Login

This is the login page.



The screenshot shows a login page titled "E-Ticket" on a purple background. The form contains the following fields and values:

- Username: dorio
- Password: \*\*\*\*\*

Below the fields is a "Login" button and two links: "Register" and "Forgot Pass".

Figure 4.2: login

### 4.1.3 Password Recovery

In case of forgetting the password, we can use this option to retrieve the password. User needs to put his/her registered Email ID and a confirmation link of recover his/her password will be sent to his/her mail.

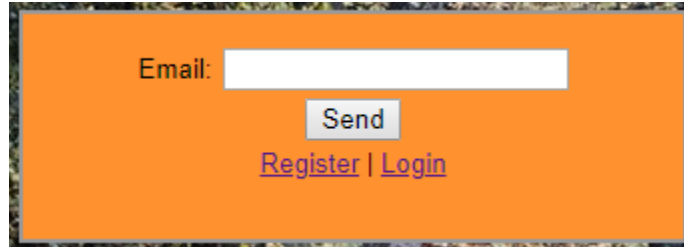
A screenshot of a password recovery form. It features an orange background. At the top left, the text "Email:" is followed by a white input field. Below the input field is a grey "Send" button. At the bottom center, there are two links: "Register" and "Login", both in purple text.

Figure 4.3: Password Recovery

## 4.2 Home Page and Booking tickets

### 4.2.1 Home page

This is the UI we get after successfully logging into the website.

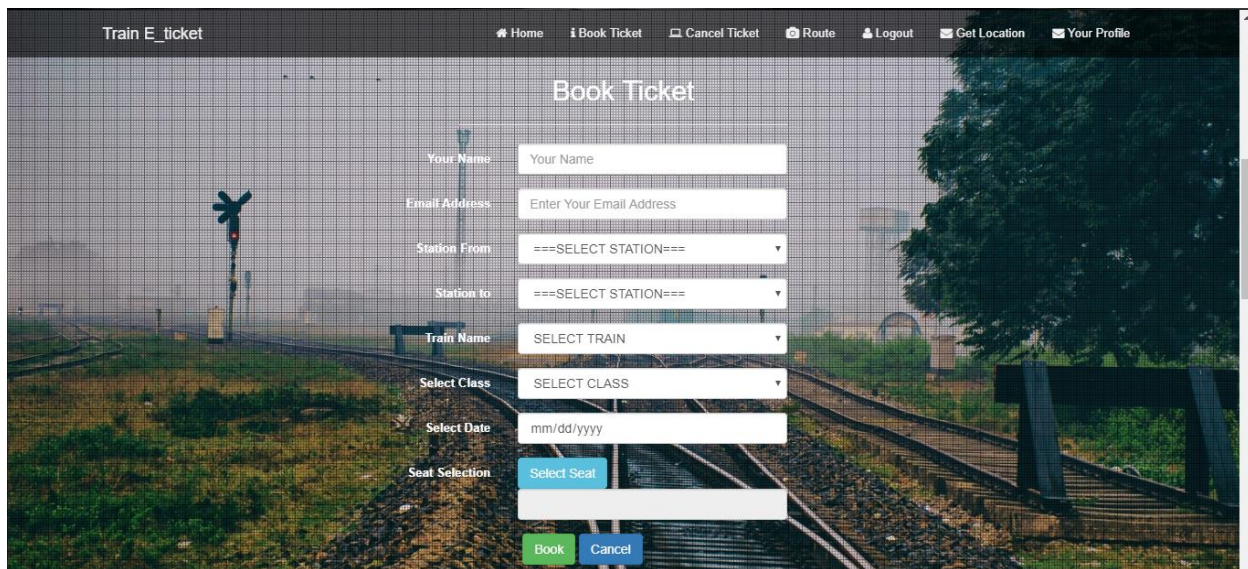
A screenshot of the "Book Ticket" form on the website. The page has a dark header with the text "Train E\_ticket" and navigation links: Home, Book Ticket, Cancel Ticket, Route, Logout, Get Location, and Your Profile. The main content area has a background image of a railway track. The form is titled "Book Ticket" and contains the following fields: "Your Name" (text input), "Email Address" (text input), "Station From" (dropdown menu), "Station to" (dropdown menu), "Train Name" (dropdown menu), "Select Class" (dropdown menu), "Select Date" (text input with format "mm/dd/yyyy"), and "Seat Selection" (a blue "Select Seat" button). At the bottom of the form are two buttons: a green "Book" button and a blue "Cancel" button.

Figure 4.4: Home Page

### 4.2.2 Book ticket

For booking tickets we have to fill up this options and click “Book” to confirm the tickets.

**Book Ticket**

Your Name: Dorio

Email Address: doriogee@gmail.com

Station From: RANGPUR

Station to: DHAKA

Train Name: Rangpur Express Eve

Select Class: ac\_chair

Select Date: 04/23/2019

Seat Selection: Select Seat  
KA 11,12,16,17

Book Cancel

Figure 4.5: Book ticket

### 4.2.3 Seat Selection

Here we have to select coach and desired seats.

**Seat Selection**

Select Coach: KA

001	002	003	004	005
006	007	008	009	010
011	012	013	014	015
016	017	018	019	020

Close

Train Name: Rangpur Express Eve

Select Class: ac\_chair

Select Date: 04/23/2019

Seat Selection: Select Seat

Book Cancel

Figure 4.6: Seat Selection

## 4.2.4 Invoice

We get this invoice after booking the tickets and a Unique ticket ID will be given which can be used later to cancel the ticket. This invoice will also be sent to the user Email.

Invoice		TicketID: HP50069	
Name	Dorio		
Email	doriogee@gmail.com		
Station	rangpur - dhaka		
Train Name	Rangpur Express Eve		
Number Of Seat	4		
Seats	KA 11,12,16,17		
Fare	1600		
Journey Date	2019-04-23		
Departure Time	08:10:00		
Payment Method	<a href="#">bKash</a>   <a href="#">Rocket</a>   <a href="#">Other</a>		

[Home](#)

Figure 4.7: Invoice

## 4.2.5 Payment Method

For paying the bill, we can use bKash, Rocket, Credit/ Debit card. As it's under progress, we will connect the payment gateway to the website as soon as we get the permission from the bank.

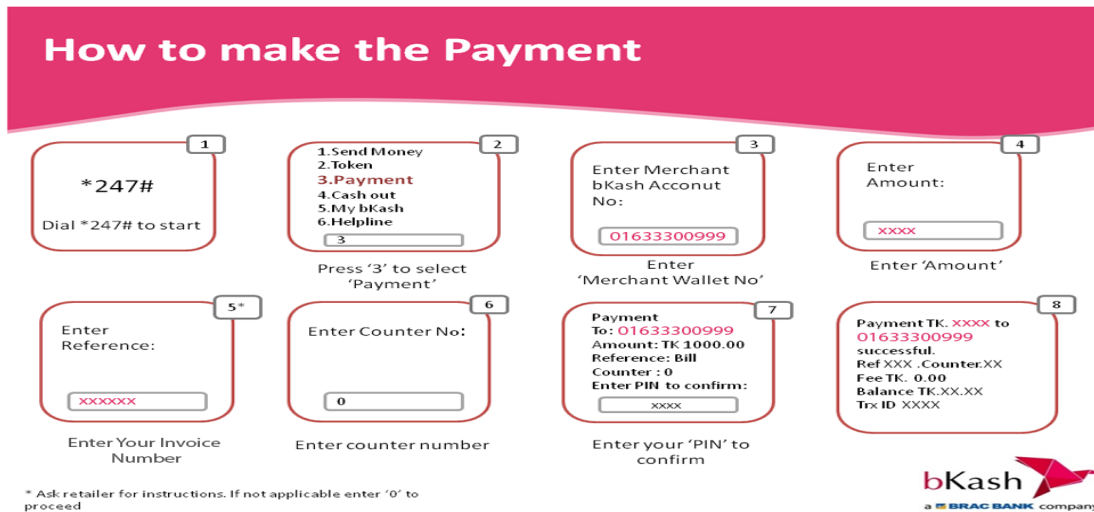


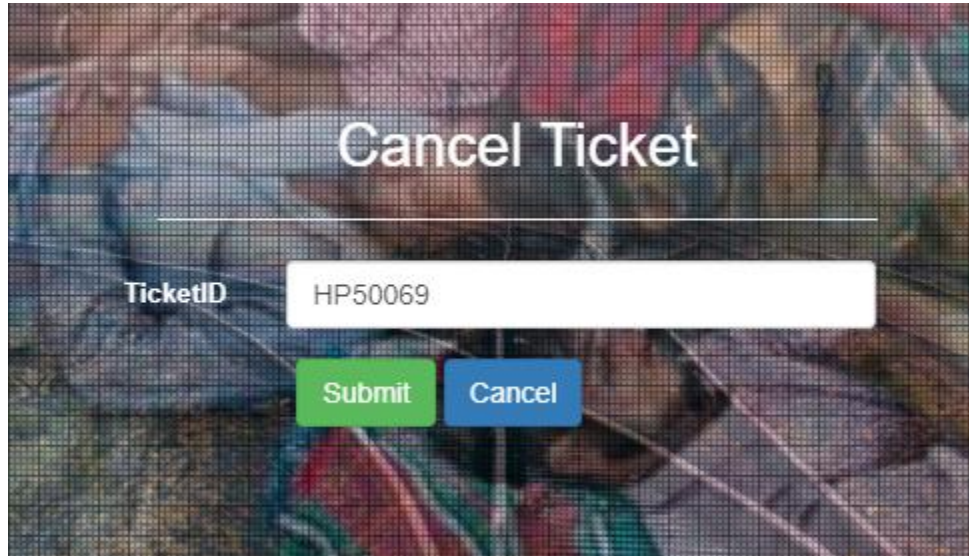
Figure 4.8: Payment using bKash.



## 4.3 Cancelling Ticket

### 4.3.1 Cancel the Ticket

In order to cancel the ticket, we have to give the unique ticket ID which we got from the booked ticket.



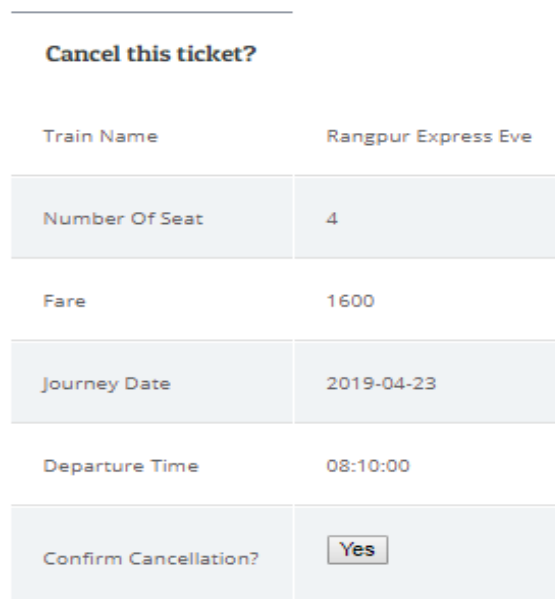
Cancel Ticket

TicketID HP50069

Submit Cancel

Figure 4.9: Cancel Ticket with unique ticket ID

### 4.3.2 Confirmation



Cancel this ticket?

Train Name	Rangpur Express Eve
Number Of Seat	4
Fare	1600
Journey Date	2019-04-23
Departure Time	08:10:00
Confirm Cancellation?	<input type="button" value="Yes"/>

Figure 4.10: Confirmation of ticket cancelling

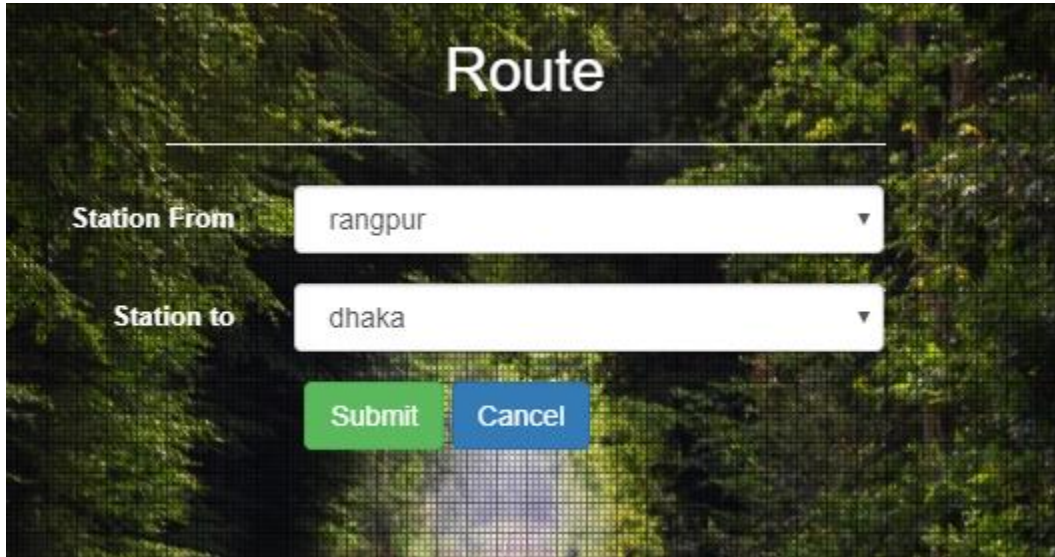
### 4.3.3 Confirmation Page:

Ticket was cancelled!

## 4.4. Route Search:

### 4.4.1 Route search

For searching the route we have to put the Station From and Station To and we will be able to see available trains ,remaining capacity, their departure time and Fare per ticket.



The screenshot shows a web form titled "Route" with a background image of a forest path. The form contains two dropdown menus. The first is labeled "Station From" and has "rangpur" selected. The second is labeled "Station to" and has "dhaka" selected. Below the dropdowns are two buttons: a green "Submit" button and a blue "Cancel" button.

Figure 4.11: Route searching

### 4.4.2 Available Trains

<b>Available Trains</b>	
Train Name	Rangpur Express Eve
Capacity	30
Station	rangpur - dhaka
Departure Time	08:10:00
Fare	400

Figure 4.12: Available trains & Details.

## 4.5 Get Location

To see the current location of listed trains we have to click “get current location”. The feature is still under development as we need to get permission from Google to use their GPS & MAP system to track down their current location.

---

<b>Get train location</b>	
Bogra Express Night	<a href="#">get current location</a>
Rangpur Express Afte	<a href="#">get current location</a>
Pabna Express Aftern	<a href="#">get current location</a>
Bogra Express Noon	<a href="#">get current location</a>
Rangpur Express Eve	<a href="#">get current location</a>
Pabna Express Mornin	<a href="#">get current location</a>

[Home](#)

Figure 4.13 : Get train location

### 4.5.1 Location of train

This is the demo of tracking the train which will be replaced by the real time location of train

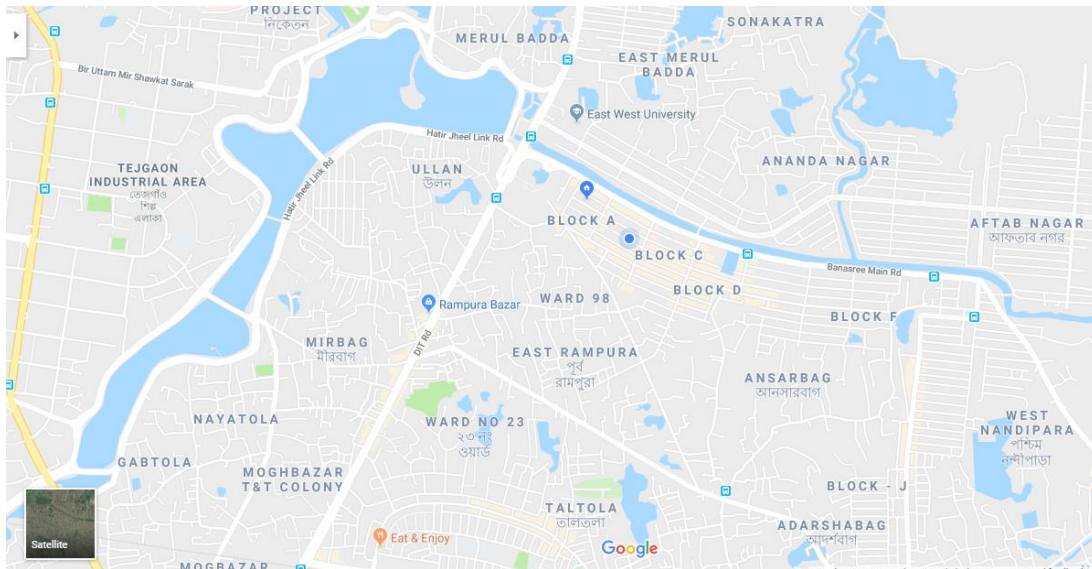


Figure 4.14 : Map view of train location

### 4.6 Ticket details

In order to see the booked ticket details we have to go through the the following processes.

#### 4.6.1 Entering mail address

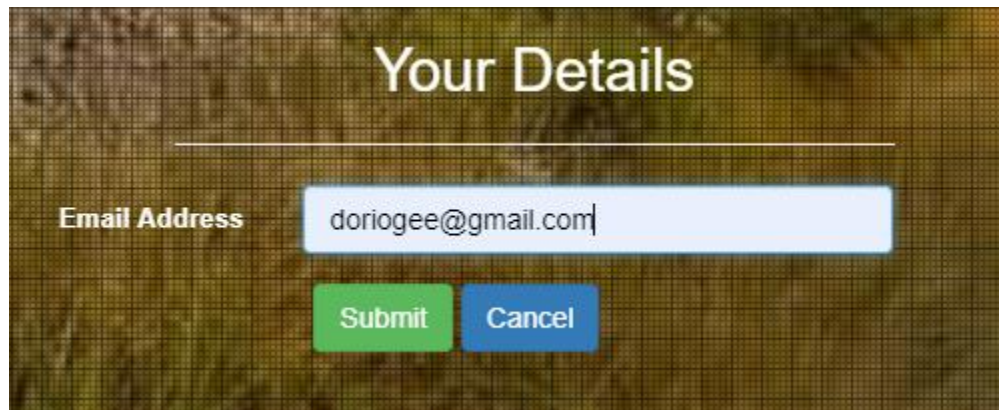


Figure 4.15 : Entering email address.

## 4.6.2 Ticket details:

### Invoice

Name	Dorio
Email	doriogee@gmail.com
Station	3 - 1
Train Name	Rangpur Express Eve
Number Of Seat	4
Fare	1600

Figure 4.16 : Details of booked ticket

## 4.7 Your Profile

Here logged in user can see his/her profile details.

<b>Your Profile</b>	
Username:	dorio
Email:	doriogee@gmail.com
Phone:	+8801940745777
Address:	Dhaka.
<a href="#">Home</a>	

Figure 4.17: Details of logged in user.

## 4.8 Logged in user

Here we can log out from the account.

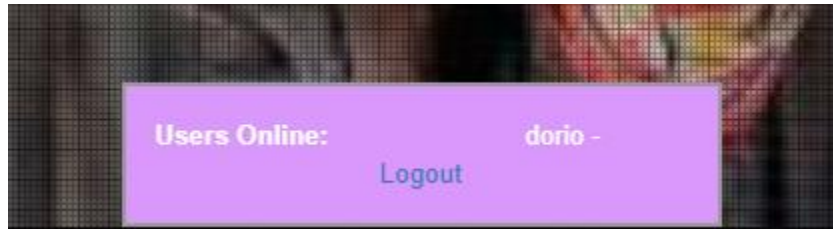


Figure 4.18: logged in user.

# CHAPTER-05

## Database Management

**Database Design:** Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. My database name is “e\_ticket” and “logintut”.

### 5.1 Database Tables

In relational databases and flat file databases, a table is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows, the cell being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows. We have total of 4 tables. They are-

- passenger
- ticket
- train
- users

### 5.2 List of Tables:

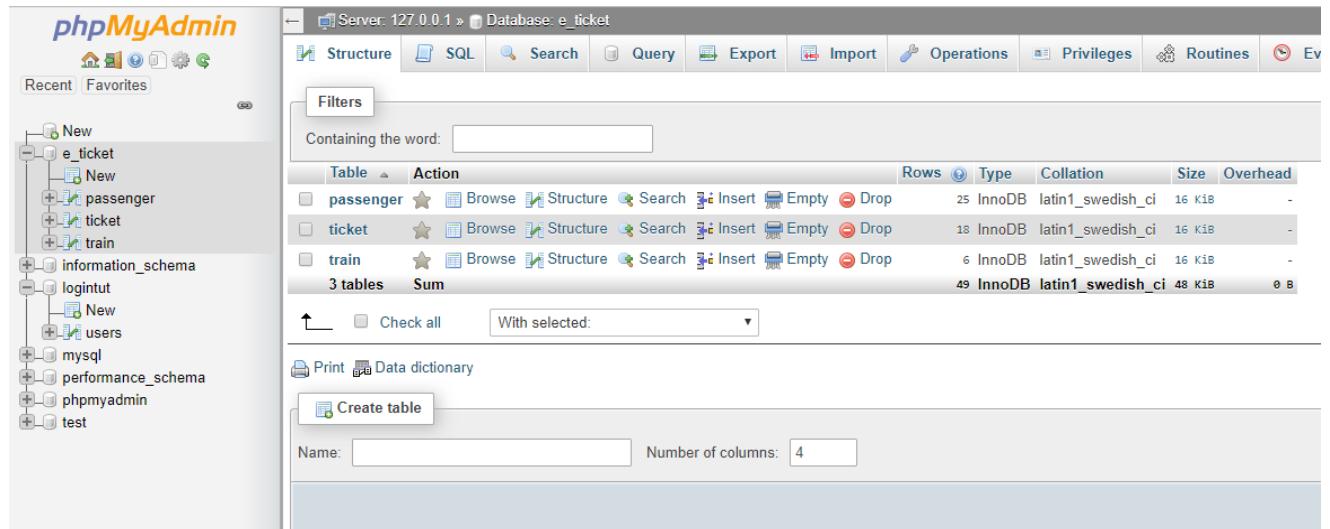


Figure 5.1: List of tables.

## 5.2.1 Passenger Table

Server: 127.0.0.1 > Database: e\_ticket > Table: passenger

Showing rows 0 - 8 (9 total, Query took 0.0116 seconds.)

```
SELECT * FROM `passenger`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options	CustomerID	name	email
<input type="checkbox"/> Edit Copy Delete	59	Md. Asik Iqbal	asikqbl@gmail.com
<input type="checkbox"/> Edit Copy Delete	60	Md. Asik Iqbal	asikqbl@gmail.com
<input type="checkbox"/> Edit Copy Delete	61	Md. Asik Iqbal	sumonxpress69@gmail.com
<input type="checkbox"/> Edit Copy Delete	62	Sinthia Tazrin	asikqbl@gmail.com
<input type="checkbox"/> Edit Copy Delete	64	Dorio	doriogee@gmail.com
<input type="checkbox"/> Edit Copy Delete	65	Dorio	doriogee@gmail.com
<input type="checkbox"/> Edit Copy Delete	66	Dorio	doriogee@gmail.com
<input type="checkbox"/> Edit Copy Delete	67	Dorio	doriogee@gmail.com

Check all | With selected: Edit Copy Delete Export

Figure 5.2: Passenger Table

## 5.2.2 Ticket Table:

Server: 127.0.0.1 > Database: e\_ticket > Table: ticket

Showing rows 0 - 4 (5 total, Query took 0.0014 seconds.)

```
SELECT * FROM `ticket`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options	ticket_no	ticketID	CustomerID	TrainName	from_place	to_place	DepartureDate	departure_time	NumberOfSeat	Total Fare
<input type="checkbox"/> Edit Copy Delete	1	RN81772	59	Rangpur Express Afte	1	3	2019-03-15	04:10:00	2	800
<input type="checkbox"/> Edit Copy Delete	2	IN88439	61	Pabna Express Momin	4	1	2019-03-12	09:10:00	2	700
<input type="checkbox"/> Edit Copy Delete	3	FB72664	64	Rangpur Express Eve	3	1	2019-04-23	08:10:00	4	1600
<input type="checkbox"/> Edit Copy Delete	4	FF45578	64	Rangpur Express Eve	3	1	2019-04-23	08:10:00	4	1600
<input type="checkbox"/> Edit Copy Delete	5	TH14545	64	Rangpur Express Eve	3	1	2019-04-23	08:10:00	4	1600

Check all | With selected: Edit Copy Delete Export

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Figure 5.3: Ticket Table



## 5.2.3 Train Table

Showing rows 0 - 5 (6 total, Query took 0.0015 seconds.)

```
SELECT * FROM `train`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	name	code	capacity	from_place	to_place	DepartureTime	Fare	loc_long	loc_lat
<input type="checkbox"/>	Bogra Express Night	12	30	dhaka	bogra	21:10:00	320	131.036	24
<input type="checkbox"/>	Rangpur Express Afte	13	30	dhaka	rangpur	04:10:00	400	131.036	24
<input type="checkbox"/>	Pabna Express Aftern	14	30	dhaka	pabna	05:10:00	350	131.036	24
<input type="checkbox"/>	Bogra Express Noon	21	30	bogra	dhaka	12:10:00	320	131.036	24
<input type="checkbox"/>	Rangpur Express Eve	31	30	rangpur	dhaka	08:10:00	400	131.036	24
<input type="checkbox"/>	Pabna Express Mornin	41	30	pabna	dhaka	09:10:00	350	131.036	24

Figure 5.4: Train Table

## 5.2.4 Users Table

Showing rows 0 - 9 (10 total, Query took 0.0015 seconds.)

```
SELECT * FROM `users`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	username	password	mobile	address	online	email	active	rtime
<input type="checkbox"/>	2	asikxd	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	1555528344	asikqbl@gmail.com	0	1552063382
<input type="checkbox"/>	3	abir1234	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	0	asikxdqbl@gmail.com	0	1552063932
<input type="checkbox"/>	4	tridib	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	0	sinthia@gmail.com	0	1552066687
<input type="checkbox"/>	5	roki	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	0	ziaulhossain1@gmail.com	0	1552067125
<input type="checkbox"/>	6	nipun	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	0	sumonxpress69@gmail.com	0	1552067528
<input type="checkbox"/>	7	12345	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	1552140021	12345@test.com	0	1552139243
<input type="checkbox"/>	8	roki123	827ccb0eea8a706c4c34a16891f84e7b	NULL	NULL	1552140868	asikqvbl@gmail.com	0	1552140783
<input type="checkbox"/>	9	Saka	e10adc3949ba59abbe56e057f20f883e	NULL	NULL	1552149700	asikxdwqsbl@gmail.com	0	1552149102
<input type="checkbox"/>	10	caesar	22eab8e77ab79102b3fcb8057354cce6	NULL	NULL	1552199360	dorigee@gmail.com	0	1552199028
<input type="checkbox"/>	12	orpita	df64b8ce3e8f0d1f0694982c5a3111b9	01752743786	Banasree	1552337922	asiksdksdk@gmail.com	0	1552336144

Figure 5.5: Users Table

# CHAPTER-06

## Specification of code

### 6.1 Index.php

```
index.php x
1 <?php
2
3     include('login/config.php');
4     include "login/functions.php";
5     session_start();
6     $logged = (!isset($_SESSION['uid'])) ? false : true;
7     if($logged)
8     {
9         $time = date('U')+50;
10        $update = mysqli_query($con,"UPDATE `users` SET `online` = '". $time."' WHERE `id` = '".$_SESSION['uid']."'");
11    }
12    else
13    {
14        header('location: http://localhost/E_TICKET-master/login/login.php');
15    }
16    ?>
17 <!DOCTYPE html>
18 <html lang="en">
19 <head>
20     <meta charset="utf-8">
21     <meta name="viewport" content="width=device-width, initial-scale=1.0">
22     <meta name="description" content="">
23     <meta name="author" content="">
24
25     <title>Welcome to The E-Ticket</title>
26     <!-- Bootstrap core CSS -->
27     <link href="css/bootstrap.min.css" rel="stylesheet">
28     <!-- Add custom CSS here -->
29     <link href="css/slidefolio.css" rel="stylesheet">
30     <!-- Font Awesome -->
31     <link href="font-awesome/css/font-awesome.min.css" rel="stylesheet">
32     <link href="css/style.dg.css" rel="stylesheet">
33     <link href="login/style1.css" rel="stylesheet">
34 </head>
35 <body>
36     <!-- Header Area -->
37     <div id="top" class="header">
38
39     </div>
40     <!-- /Header Area -->
41     <div id="nav">
42     <!-- Navigation -->
43     <nav class="navbar navbar-new" role="navigation">
44     <div class="container">
45     <!-- Brand and toggle get grouped for better mobile display -->
46     <div class="navbar-header">
47     <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#mobilemenu">
48     <span class="sr-only">Toggle navigation</span>
49     <span class="icon-bar"></span>
50     <span class="icon-bar"></span>
51     <span class="icon-bar"></span>
52     </button>
53     <a href="#" class="navbar-brand">Train E_ticket</a>
54     </div>
55     <div class="collapse navbar-collapse" id="mobilemenu">
56
57     <ul class="nav navbar-nav navbar-right text-center">
```

```

58 <li><a href="#top"><i class="service-icon fa fa-home"></i>&nbsp;Home</a></li>
59 <li><a href="#book_ticket"><i class="service-icon fa fa-info"></i>&nbsp;Book Ticket</a></li>
60 <li><a href="#cancel_ticket"><i class="service-icon fa fa-laptop"></i>&nbsp;Cancel Ticket</a></li>
61 <li><a href="#route"><i class="service-icon fa fa-camera"></i>&nbsp;Route</a></li>
62 <?php if($logged)
63 {
64 <li><a href="http://localhost/E_TICKET-master/login/logout.php"><i class="service-icon fa fa-user"></i>&nbsp;Logout</a></li>;
65 }
66 else{
67 <li><a href="http://localhost/E_TICKET-master/login/login.php"><i class="service-icon fa fa-user"></i>&nbsp;Login</a></li>;
68 }?>
69 <li><a href="getlocation.php"><i class="service-icon fa fa-envelope"></i>&nbsp;Get Location</a></li>
70 <li><a href="login/profile.php?user=?&$_SESSION['uid']?><i class="service-icon fa fa-envelope"></i>&nbsp;Your Profile</a></li>
71 </ul>
72 </div><!-- /.navbar-collapse -->
73 </div>
74 </nav>
75 <!-- /Navigation -->
76 </div>
77 <!-- About -->
78 <div id="book_ticket" class = "style">
79 <div class="container">
80 <div class="row">
81 <div class="col-md-4 col-md-offset-4 text-center">
82 <h2>Book Ticket</h2>
83 <hr>
84 </div>
85 <div class="col-md-5 col-md-offset-3">
86 <!-- contact form starts -->
87 <form action="contact.php" method="post" id="apply-form" class="form-horizontal">
88 <fieldset>
89 <div class="form-group">
90 <label class="col-sm-4 control-label" for="name">Your Name</label>
91 <div class="col-sm-8">
92 <input type="text" placeholder="Your Name" class="form-control" name="name" id="name" required>
93 </div>
94 </div>
95 <div class="form-group">
96 <label class="col-sm-4 control-label" for="email">Email Address</label>
97 <div class="col-sm-8">
98 <input type="text" placeholder="Enter Your Email Address" class="form-control" name="email" id="email" required>
99 </div>
100 </div>
101 <div class="form-group">
102 <label class="col-sm-4 control-label" for="subject">Station From</label>
103 <div class="col-sm-8">
104 <select name="station from" id="st-from" class="form-control" required="required" >
105 <option value="0" label=" ===SELECT STATION===> SELECT STATION</option>
106 <option value="1" label="DHAKA">Dhaka</option>
107 <option value="2" label="BOGRA">Bogra</option>
108 <option value="3" label="RANGPUR">Rangpur</option>
109 <option value="4" label="PABNA">Pabna</option>
110 </select>
111 </div>
112 </div>
113 </div>
114 </div>

```

```

115 <div class="form-group">
116 <label class="col-sm-4 control-label" for="message">Station to</label>
117 <div class="col-sm-8">
118 <select name="station to" id="st-to" class="form-control" tabindex="2" required="required">
119 <option value="0" label=" ==SELECT STATION=="> SELECT STATION</option>
120 <option value="1" label="DHAKA">Dhaka</option>
121 <option value="2" label="BOGRA">Bogra</option>
122 <option value="3" label="RANGPUR">Rangpur</option>
123 <option value="4" label="PABNA">Pabna</option>
124 </select>
125 </div>
126 </div>
127 </div>
128
129 <div class="form-group">
130 <label class="col-sm-4 control-label" for="name">Train Name</label>
131 <div class="col-sm-8">
132 <select name="train_name" id="select_train" class="form-control" tabindex="2" required="required">
133 <option value="" label=" SELECT TRAIN"> Train Name </option>
134 </select>
135 </div>
136 </div>
137 </div>
138 <div class="form-group">
139 <label class="col-sm-4 control-label" for="name">Select Class</label>
140 <div class="col-sm-8">
141 <select name="class_name" id="select_class" class="form-control" tabindex="2" required="required">
142 <option value="0" label=" SELECT CLASS"> Class Name </option>
143 <option value="1" label="ac_chair"> AC Chair </option>
144 <option value="2" label="s_chair"> S Chair </option>
145 <option value="3" label="chair"> Chair </option>
146 </select>
147 </div>
148 </div>
149 </div>
150
151 <div class="form-group">
152 <label class="col-sm-4 control-label" for="name">Select Date</label>
153 <div class="col-sm-8">
154 <input type="date" name="journeydate" class="form-control">
155 </div>
156 </div>
157 </div>
158
159 <div class="form-group">
160 <label class="col-sm-4 control-label" for="message">Seat Selection</label>
161 <div class="col-sm-8">
162 <button type="button" class="btn btn-info" data-toggle="modal" data-target="#select-seat">Select Seat</button>
163 <input type="text" id="selected-seat" class="form-control" name="selected_seat" readonly>
164 <input type="hidden" name="seatcount" id="seatcount" value="">
165 </div>
166 </div>
167 <div style="color:#000000" class="modal fade" tabindex="-2" id="select-seat" role="dialog" aria-labelledby="select-seat" aria-hidden="true">
168 <div class="modal-dialog">
169 <div class="modal-content">
170 <div class="modal-header">
171 <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></button>

```

```

172     <h4 class="modal-title" id="select-seat">Seat Selection</h4>
173 </div>
174 <div class="ss-modal-body">
175     <form name="ss_form" id="ss_form">
176         <label>Select Coach</label>
177         <select name="coach" id="coach">
178             <option value="KA">KA</option>
179             <option value="KHA">KHA</option>
180             <option value="GA">GA</option>
181             <option value="GHA">GHA</option>
182         </select>
183         <div class="seat">
184             <table>
185                 <tr><td><input type="checkbox" name="seatno" value="1">001</td>
186                 <td><input type="checkbox" name="seatno" value="2">002</td><td class="blank"></td>
187                 <td><input type="checkbox" name="seatno" value="3">003</td>
188                 <td><input type="checkbox" name="seatno" value="4">004</td>
189                 <td><input type="checkbox" name="seatno" value="5">005</td></tr>
190                 <tr>
191                 <td><input type="checkbox" name="seatno" value="6">006</td>
192                 <td><input type="checkbox" name="seatno" value="7">007</td><td class="blank"></td>
193                 <td><input type="checkbox" name="seatno" value="8">008</td>
194                 <td><input type="checkbox" name="seatno" value="9">009</td>
195                 <td><input type="checkbox" name="seatno" value="10">010</td>
196                 </tr>
197                 <tr>
198                 <td><input type="checkbox" name="seatno" value="11">011</td>
199                 <td><input type="checkbox" name="seatno" value="12">012</td><td class="blank"></td>
200                 <td><input type="checkbox" name="seatno" value="13">013</td>
201                 <td><input type="checkbox" name="seatno" value="14">014</td>
202                 <td><input type="checkbox" name="seatno" value="15">015</td>
203                 </tr>
204                 <tr>
205                 <td><input type="checkbox" name="seatno" value="16">016</td>
206                 <td><input type="checkbox" name="seatno" value="17">017</td><td class="blank"></td>
207                 <td><input type="checkbox" name="seatno" value="18">018</td>
208                 <td><input type="checkbox" name="seatno" value="19">019</td>
209                 <td><input type="checkbox" name="seatno" value="20">020</td>
210                 </tr>
211             </table>
212         </div>
213     </form>
214
215 </div>
216 <div class="modal-footer"><div id="response"></div>
217     <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
218 </div>
219 </div>
220 </div>
221 </div>
222 <div class="col-sm-offset-4 col-sm-8">
223     <button type="submit" class="btn btn-success">Book</button>
224     <button type="reset" class="btn btn-primary">Cancel</button>
225 </div>
226 </fieldset>
227 </form>

```

```

229         <!-- contact form ends -->
230     </div>
231 </div>
232 </div>
233 </div>
234 <!-- /About -->
235 <!-- Services -->
236 <div id="cancel_ticket" class = "style">
237     <div class="container">
238         <div class="row">
239             <div class="col-md-4 col-md-offset-4 text-center">
240
241                 <h2>Cancel Ticket</h2>
242                 <hr>
243             </div>
244             <div class="col-md-5 col-md-offset-3">
245                 <!-- contact form starts -->
246                 <form action="database_delete_reservation.php" method="post" id="contact-form" class="form-horizontal">
247                     <fieldset>
248
249                         <div class="form-group">
250                             <label class="col-sm-4 control-label" for="email">TicketID</label>
251                             <div class="col-sm-8">
252                                 <input type="text" placeholder="Enter Your TicketID" class="form-control" name="tid" id="tid" required>
253                             </div>
254                         </div>
255
256                         <div class="col-sm-offset-4 col-sm-8">
257                             <button type="submit0" class="btn btn-success">Submit</button>
258                             <button type="reset" class="btn btn-primary">Cancel</button>
259                         </div>
260                     </fieldset>
261                 </form>
262
263                 <!-- contact form ends -->
264             </div>
265         </div>
266     </div>
267 </div>
268 <!-- /Services -->
269
270 <!-- Portfolio -->
271 <div id="route" class = "style">
272     <div class="container">
273         <div class="row">
274             <div class="col-md-4 col-md-offset-4 text-center">
275                 <h2>Route</h2>
276                 <hr>
277             </div>
278             <div class="col-md-5 col-md-offset-3">
279                 <!-- contact form starts -->
280                 <form action="Search_route.php" method="post" id="contact-form" class="form-horizontal">
281                     <fieldset>
282
283                         <div class="form-group">
284                             <label class="col-sm-4 control-label" for="subject">Station From</label>
285                             <div class="col-sm-8">
286                                 <select name="station_from" id="station_from" class="form-control" required="required" >

```

```

286         <option value="0" label=" ===SELECT STATION===> SELECT STATION</option>
287         <option value="dhaka" label="dhaka">Dhaka</option>
288         <option value="bogra" label="bogra">Bogra</option>
289         <option value="rangpur" label="rangpur">Rangpur</option>
290         <option value="pabna" label="pabna">Pabna</option>
291
292     </select>
293
294 </div>
295 </div>
296 <div class="form-group">
297     <label class="col-sm-4 control-label" for="message">Station to</label>
298     <div class="col-sm-8">
299         <select name="station_to" id="station_to" class="form-control" tabindex="2" required="required" >
300             <option value="0" label=" ===SELECT STATION===> SELECT STATION</option>
301             <option value="dhaka" label="dhaka">Dhaka</option>
302             <option value="bogra" label="bogra">Bogra</option>
303             <option value="rangpur" label="rangpur">Rangpur</option>
304             <option value="pabna" label="pagna">Pabna</option>
305
306         </select>
307     </div>
308 </div>
309
310 <div class="col-sm-offset-4 col-sm-8">
311     <button type="submit1" class="btn btn-success">Submit</button>
312     <button type="reset" class="btn btn-primary">Cancel</button>
313 </div>
314 </fieldset>
315 </form>
316
317
318
319 <!-- contact form ends -->
320 </div>
321 </div>
322 </div>
323 </div>
324 <!-- /Portfolio -->
325 <!-- Contact -->
326 <div id="details" class = "style">
327     <div class="container">
328         <div class="row">
329             <div class="col-md-4 col-md-offset-4 text-center">
330                 <h2>Your Details</h2>
331                 <hr>
332             </div>
333             <div class="col-md-5 col-md-offset-3">
334                 <!-- contact form starts -->
335                 <form action="DatabaseSearch.php" method="post" id="contact-form" class="form-horizontal">
336                     <fieldset>
337
338                         <div class="form-group">
339                             <label class="col-sm-4 control-label" for="email">Email Address</label>
340                             <div class="col-sm-8">
341                                 <input type="text" placeholder="Enter Your Email Address" class="form-control" name="email" id="email" required>
342                             </div>

```

```

343     </div>
344
345     <div class="col-sm-offset-4 col-sm-8">
346         <button type="submit2" class="btn btn-success">Submit</button>
347         <button type="reset" class="btn btn-primary">Cancel</button>
348     </div>
349 </fieldset>
350 </form>
351
352 <!-- contact form ends -->
353 </div>
354 </div>
355 </div>
356 </div>
357 <!-- /Contact -->
358 <!-- Footer -->
359 <div id="border">
360
361 <table cellpadding="2" cellspacing="0" border="0" width="100%">
362
363 <tr>
364
365 <td><b>Users Online:</b></td>
366
367 <td>
368
369 <?php
370
371
372 $res = mysqli_query($con,"SELECT * FROM `users` WHERE `online` > '".date('U')."'");
373
374
375 while($row = mysqli_fetch_assoc($res)){
376
377
378     echo $row['username']." - ";
379
380 }
381
382 ?>
383
384 </td>
385
386 </tr>
387
388 <tr>
389
390 <td colspan="2" align="center"><a href="login/logout.php">Logout</a></td>
391
392 </tr>
393
394 </table>
395
396 </div>
397 <!-- /Footer -->
398 <!-- Bootstrap core JavaScript -->
399 <!-- Placed at the end of the document so the pages load faster -->

```



```

400 <script src="js/jquery.js"></script>
401 <script src="js/jquery-scrolltofixed-min.js"></script>
402 <script src="js/jquery.vegas.js"></script>
403 <script src="js/jquery.mixitup.min.js"></script>
404 <script src="js/jquery.validate.min.js"></script>
405 <script src="js/script.js"></script>
406 <script src="js/bootstrap.js"></script>
407
408 <!-- Slideshow Background -->
409 <script>
410 $.vegas('slideshow', {
411   delay:5000,
412   backgrounds:[
413     { src: './img/t6.jpg', fade:2000 },
414     { src: './img/t2.jpg', fade:2000 },
415     { src: './img/t4.jpg', fade:2000 },
416     { src: './img/t11.jpg', fade:2000 },
417     { src: './img/t1.jpg', fade:2000 },
418     { src: './img/t3.jpg', fade:2000 },
419   ]
420 })('overlay', {
421   src: './img/overlay.png'
422 });
423 </script>
424 <!-- /Slideshow Background -->
425
426 <!-- Mixitup : Grid -->
427 <script>
428   $(function(){
429     $('#Grid').mixitup();
430   });
431 </script>
432 <!-- /Mixitup : Grid -->
433
434 <!-- Custom JavaScript for Smooth Scrolling - Put in a custom JavaScript file to clean this up -->
435 <script>
436   $(function() {
437     $('a[href*=#]:not([href=#])').click(function() {
438       if (location.pathname.replace(/^\//, '') == this.pathname.replace(/^\//, '')
439         || location.hostname == this.hostname) {
440         var target = $(this.hash);
441         target = target.length ? target : $('[name=' + this.hash.slice(1) +']');
442         if (target.length) {
443           $('html,body').animate({
444             scrollTop: target.offset().top
445           }, 1000);
446           return false;
447         }
448       }
449     });
450   });
451 </script>
452 <!-- Navbar -->
453 <script type="text/javascript">
454   $(document).ready(function() {
455     $('#nav').scrollToFixed();
456   });
457 </script>
458 <script type="text/javascript" src="js/main.dg.js"></script>
459 <!-- /Navbar-->
460
461 </body>
462
463 </html>

```

## 6.2 Search route

```
Search_route.php x
1 <html>
2 <head>
3 <link rel="stylesheet" href="css/styles_search.css">
4 </head>
5 </html>
6 <?php
7
8 $FromStation=$_POST["station_from"];
9 $ToStation=$_POST["station_to"];
10
11 $con=mysqli_connect("localhost","root","");
12     if($con==false)
13         die("could not connect");
14     if(!mysqli_select_db($con,"e_ticket"))
15         echo "could not connect_db";
16
17 $sql= "select * from train where from_place='$FromStation' and to_place='$ToStation' ";
18 if(!($result1= mysqli_query($con,$sql)))
19     {echo "problem loading";}
20 $row1 = mysqli_fetch_array($result1);
21
22
23 echo "
24
25 <html>
26 <header>
27     <link href='css/style.css' rel='stylesheet'>
28     <script async defer
29     src='https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap'>
30     </script>
31 </header>
32
33 <table>
34     <thead>
35     <tr>
36         <th><h2>Available Trains</h2></th>
37     </tr>
38     </thead>
39     <tbody>
40     <tr>
41         <td>Train Name</td>
42         <td>".$row1['name']. "</td>
43
44     </tr>
45     <tr>
46         <td>Capacity</td>
47         <td>".$row1['capacity']. "</td>
48
49     </tr>
50     <tr>
51         <td>Station </td>
52         <td>".$row1['from_place']. " - ".$row1['to_place']. "</td>
53
54     </tr>
55     <tr>
56         <td>Departure Time</td>
57         <td>".$row1['DepartureTime']. "</td>
```

```
58
59     </tr>
60     <tr>
61         <td>Fare</td>
62         <td>".$rowl['Fare']. "</td>
63
64     </tr>
65     <tr>
66         <td colspan='2'>Current location</td>
67     </tr>
68     <tr>
69         <td colspan='2'> <div id='map'></div>
70
71         </td>
72     </tr>
73
74 </tbody>
75 </table>
76 <script>
77     function initMap() {
78         var uluru = {lat: ".$rowl['loc_lat'].", lng: ".$rowl['loc_long']. "};
79
80         var map = new google.maps.Map(
81             document.getElementById('map'), {zoom: 4, center: uluru});
82
83         var marker = new google.maps.Marker({position: uluru, map: map});
84     }
85 </script>
86 </html>";
87 <?>
```

## 6.3 Get location

```
getlocation.php
1 <?php
2
3     include('login/config.php');
4     include "login/functions.php";
5 session_start();
6 $logged = (!isset($_SESSION['uid'])) ? false : true;
7 if($logged)
8 {
9     $time = date('U')+50;
10    $update = mysqli_query($con,"UPDATE `users` SET `online` = '". $time."' WHERE `id` = '".$_SESSION['uid']."'");
11 }
12 else
13 {
14     header('location: http://localhost/E_TICKET-master/login/login.php');
15 }
16
17 $con=mysqli_connect("localhost","root","");
18 if(!mysqli_select_db($con,"e_ticket"))
19     echo "could not connect_db";
20
21 $getTrains = "SELECT * FROM `train`";
22 $result = mysqli_query($con,$getTrains);
23 $data = mysqli_fetch_all($result);
24
25 if(isset($_GET['get']))
26 {
27     $code = $_GET['get'];
28     $getCode = "SELECT * FROM `train` WHERE `code` = '$code'";
29     $res = mysqli_query($con,$getCode);
30     $dt = mysqli_fetch_array($res);
31 }
32
33 ?>
34
35 <html>
36 <header>
37 <link href='css/style.css' rel='stylesheet'>
38 <script async defer
39 src='https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap'>
40 </script>
41 </header>
42 <body>
43 <table>
44 <thead>
45 <tr>
46 <th>Get train location</th>
47 </tr>
48 </thead>
49 <tbody>
50
51 <?php
52     if(isset($_GET['get']))
53     {
54         ?>
55         <tr>
56 <td colspan='2'>Current location of <b><?=$dt['name']?></b></td>
57 </tr>
```

```

58     <tr>
59         <td colspan='2' <div id='map'></div>
60     </td>
61 </tr>
62 </tr>
63 <tr>
64 <td><i>Other Trains</i></td>
65 </tr>
66     <?php
67     </tr>
68     foreach($data as $d)
69     {
70         <?>
71         <tr>
72             <td><?=$d[0]></td>
73             <td><a href="?get=?=$d[1]?">get current location</a></td>
74         </tr>
75     <?php
76     }
77 <?>
78 <tr>
79     <td colspan="2"><center><a href="http://localhost/E_TICKET-master">Home</a></center></td>
80 </tr>
81 </tbody>
82 </table>
83 <?php
84     if(isset($_GET['get']))
85     {
86         echo "<script>
87         function initMap() {
88             var uluru = {lat: ".$dt['loc_lat'].", lng: ".$dt['loc_long']."};
89
90             var map = new google.maps.Map(
91                 document.getElementById('map'), {zoom: 4, center: uluru});
92
93             var marker = new google.maps.Marker({position: uluru, map: map});
94         }
95         </script>";
96     }
97 <?>
98 </body>
99 </html>

```

## 6.4 Delete reservation:

```
database_delete_reservation.php
1 <html>
2 <head>
3 <link rel="stylesheet" href="css/styles_search.css">
4 </head>
5 </html>
6 <?php
7
8 $con=mysqli_connect("localhost","root","");
9 if($con==false)
10 die("could not connect");
11 if(!(mysqli_select_db($con,"e_ticket")))
12 echo "could not connect_db";
13
14 if(isset($_GET['confirm']))
15 {
16     $tid = $_GET['TID'];
17     $sql= "delete from ticket where ticketID='$tid' ";
18     if(!(mysqli_query($con,$sql)))
19         {echo "problem2 loading";}
20
21     echo 'Ticket was cancelled!';
22 }
23 else if(isset($_POST['tid']))
24 {
25     $tid=$_POST["tid"];
26
27
28
29
30
31     /* */
32     $query = "SELECT * FROM `ticket` WHERE `ticketID` = '$tid'";
33     $result = mysqli_query($con,$query);
34     $data = mysqli_fetch_array($result);
35     echo "
36 <html>
37 <header>
38 <link href='css/style.css' rel='stylesheet'>
39 </header>
40 <table>
41 <thead>
42 <tr>
43 <th>Cancel this ticket?</th>
44 </tr>
45 </thead>
46 <tbody>
47 <tr>
48 <td>Train Name</td>
49 <td>".$data['TrainName']. "</td>
50
51 </tr>
52 <tr>
53 <td>Number Of Seat</td>
54 <td>".$data['NumberOfSeat']. "</td>
55
56 </tr>
57
58 <td>Fare</td>
59 <td>".$data['Total Fare']. "</td>
60
61 </tr>
62 <tr>
63 <td>Journey Date</td>
64 <td>".$data['DepartureDate']. "</td>
65
66 </tr>
67 <tr>
68 <td>Departure Time</td>
69 <td>".$data['departure_time']. "</td>
70
71 </tr>
72 <tr>
73 <td>Confirm Cancellation?</td>
74 <td><form action=".$_SERVER['PHP_SELF']. " method='get'><input type='submit' name='confirm' value='Yes'><input type='hidden' name='TID' value=".$tid."></form></td>
75 </tr>
76 </tbody>
77 </table>
78 </html>";
79
80 ] ?>
```

## 6.5 Contact.php

```
contact.php
1 <?php
2
3 function sendtomail($mail,$name,$f,$t,$date,$time,$seats,$fare)
4 {
5     $subject = 'Ticket purchase confirmation!';
6     $message = '<p>Hello, '.$name.'!<br>Your ticket purchase process was completed. Details of your journey:<br>
7     <b>Journey From:</b> '.$f.'<br>
8     <b>Journey To:</b> '.$t.'<br>
9     <b>Journey Date:</b> '.$date.'<br>
10    <b>Journey time:</b> '.$time.'<br>
11    <b>Seats:</b> '.$seats.'<br>
12    <b>Total Fare</b>: '.$fare.'<br>
13    '
14    $headers = "From: localhost < info@localhost >\n";
15    $headers .= "X-Sender: localhost < info@localhost >\n";
16    $headers .= 'X-Mailer: PHP/' . phpversion();
17    $headers .= "X-Priority: 1\n";
18    $headers .= "Return-Path: info@localhost\n";
19    $headers .= "MIME-Version: 1.0\r\n";
20    $headers .= "Content-Type: text/html; charset=iso-8859-1\n";
21
22    mail($mail, $subject, $message, $headers);
23 }
24
25 function getTicketID()
26 {
27     $strs = array("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z");
28     $dghts = rand(10000,99999);
29     $str1 = rand(0,25);
30     $str2 = rand(0,25);
31     $code = $strs[$str1].$strs[$str2].$dghts;
32     return $code;
33 }
34
35 $uname= $_POST["name"];
36 $email= $_POST["email"];
37 $fromPlace= $_POST["station_from"];
38 $toPlace= $_POST["station_to"];
39 $trainName= $_POST['train_name'];
40 $seats= $_POST["selected_seat"];
41 $numberOfSeat = $_POST['seatcount'];
42 $jDate = $_POST['journeydate'];
43 $ticketID = getTicketID();
44
45 $con=mysqli_connect("localhost","root","");
46 if(!(mysqli_select_db($con,"e_ticket")))
47     echo "could not connect_db";
48
49
50 // $sql0 = "select name from train where from_place = '$fromPlace' and to_place = '$toPlace'";
51
52     $sql= "insert into passenger values('','$uname','$email')";
53 if(!($result= mysqli_query($con,$sql)))
54     echo "problem loading1";
55
56 $sql1= "select CustomerID from passenger where email= '$email'";
57 if(!($result= mysqli_query($con,$sql1)))
```

```

58     echo "problem loading2";
59     $row = mysqli_fetch_array($result);
60     $CustomerID= $row['CustomerID'];
61     $jStruc = $FromPlace.$ToPlace;
62     $sql1= "select from_place,to_place from train where code= '$jStruc'";
63     if(!($result= mysqli_query($con,$sql1)))
64         echo "problem loading2";
65     $row = mysqli_fetch_array($result);
66     $st1= $row['from_place'];
67     $st2 = $row['to_place'];
68
69     $sql2= "select name,DepartureTime,Fare from train where code= '$TrainName'";
70     if(!($result= mysqli_query($con,$sql2)))
71         echo "problem loading3";
72     $row = mysqli_fetch_array($result);
73     $TotalFare= $row['Fare']*$NumberOfSeat;
74     $DepartureTime= $row['DepartureTime'];
75     $tName = $row['name'];
76
77     $sql= "insert into ticket values('','$ticketID','$CustomerID','$tName','$FromPlace','$ToPlace','$jDate','$DepartureTime','$NumberOfSeat','$TotalFare')";
78     if(!($result= mysqli_query($con,$sql)))
79         echo "problem loading4";
80
81     sendtomail($Email,$Uname,$st1,$st2,$jDate,$DepartureTime,$Seats,$TotalFare);
82 echo "
83
84 <html>
85 <header>
86     <link href='css/style.css' rel='stylesheet'>
87 </header>
88
89 <table>
90     <thead>
91         <tr>
92             <th>Invoice</th>
93             <th>TicketID: ".$ticketID."</th>
94         </tr>
95     </thead>
96     <tbody>
97         <tr>
98             <td>Name</td>
99             <td>".$Uname."</td>
100
101         </tr>
102         <tr>
103             <td>Email</td>
104             <td>".$Email."</td>
105
106         </tr>
107         <tr>
108             <td>Station</td>
109             <td>".$st1." - ".$st2."</td>
110
111         </tr>
112         <tr>
113             <td>Train Name</td>
114             <td>".$tName."</td>
115
116         </tr>
117         <tr>
118             <td>Number Of Seat</td>
119             <td>".$NumberOfSeat."</td>
120
121         </tr>
122         <tr>
123             <td>Seats</td>
124             <td>".$Seats."</td>
125
126         </tr>
127         <tr>
128             <td>Fare</td>
129             <td>".$TotalFare."</td>
130
131         </tr>
132         <tr>
133             <td>Journey Date</td>
134             <td>".$jDate."</td>
135
136         </tr>
137         <tr>
138             <td>Departure Time</td>
139             <td>".$DepartureTime."</td>
140
141         </tr>
142         <tr>
143             <td>Payment Method</td>
144             <td><a href='payment.php' target='_blank'>bKash</a> | <a href='payment.php' target='_blank'>Rocket | <a href='payment.php' target='_blank'>Other</a></td>
145
146         </tr>
147         <tr>
148             <td colspan='2'><center><a href='http://localhost/E_TICKET-master'>Home</a></center></td>
149
150     </tbody>
151 </table>
152 </html>"
153
154

```



## 6.6 Database search

```
DatabaseSearch.php x
1 <html>
2 <head>
3 <link rel="stylesheet" href="css/styles_search.css">
4 </head>
5 </html>
6 <?php
7
8 $email=$_POST["email"];
9
10 $con=mysqli_connect("localhost","root","");
11     if($con==false)
12         die("could not connect");
13     if(! (mysqli_select_db($con,"e_ticket")))
14         echo "could not connect_db";
15
16 $sql= "select * from passenger where email='$email' ";
17 if(! ($result1= mysqli_query ($con,"$sql")))
18     {echo "problem loading";}
19 $row1 = mysqli_fetch_array ($result1);
20 $customerID= $row1['CustomerID'];
21
22
23 $sql= "select * from ticket where CustomerID='$customerID' ";
24
25 if(! ($result= mysqli_query ($con,"$sql")))
26     {echo "problem loading";}
27
28 $row = mysqli_fetch_array ($result);
29
30 echo "
31
32 <html>
33 <header>
34     <link href='css/style.css' rel='stylesheet'>
35 </header>
36
37 <table>
38     <thead>
39         <tr>
40             <th><h2>Invoice</h2></th>
41         </tr>
42     </thead>
43     <tbody>
44         <tr>
45             <td>Name</td>
46             <td>". $row1['name'] . "</td>
47
48         </tr>
49         <tr>
50             <td>Email</td>
51             <td>". $row1['email'] . "</td>
52
53         </tr>
54         <tr>
55             <td>Station</td>
56             <td>". $row['from_place'] . " - ". $row['to_place'] . "</td>
```

```

58     </tr>
59     <tr>
60         <td>Train Name</td>
61         <td>". $row['TrainName'] ."</td>
62     </tr>
63     <tr>
64         <td>Number Of Seat</td>
65         <td>". $row['NumberOfSeat'] ."</td>
66     </tr>
67     <tr>
68         <td>Fare</td>
69         <td>". $row['Total Fare'] ."</td>
70     </tr>
71 </tbody>
72 </table>
73 </html>";
74 <?>

```

## 6.7 DB config:

```

config.php x
1
2
3 <?php
4 $mysql_hostname = "localhost";
5 $mysql_user = "root";
6 $mysql_password = "";
7 $mysql_database = "e_ticket";
8 $prefix = "";
9 $con = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password) or die("Could not connect database");
10 mysqli_select_db($con, $mysql_database) or die("Could not select database");
11 <?>

```

## **CHAPTER-07**

### **Conclusion**

#### **7.1 Future Aspects**

- ✓ We could try to do the betterment of the user interface.
- ✓ The mobile and email verification will be added.
- ✓ The Location Tracking process will be added.
- ✓ This project may be used professionally by creating a domain host for the web.
- ✓ The terms and conditions feature can be updated depending on the user demand and new technology.
- ✓ The history of the user can be saved on this project but there is no account system for them. Account system can be added for the long-term user for better reliability (Under processing).

#### **7.2 Conclusions**

The train is still the most cheap and reliable way for travelling in Bangladesh. People can easily travel cities by avoiding any kinds of traffic jams. Beside these, train journey is comfortable due to the availabilities of sanitation, freedom of moving around and enjoying the nature. But, in our country. Let alone this advantage; In our country, we cannot even get the ticket in a proper way. In existing online ticket buying system has some drawbacks such as: Non-friendly User interface; Seat selection is not available right away, we have to come back again to select the seats; The most popular bKash payment method is not available. However, in our website, we tried to overcome these drawbacks and we added some new features like “Getting real time location of train” as train schedule is barely maintained. By using this feature, people can arrange their time schedule to reach the train station. We have also added the bKash payment system and a Handsome looking user friendly interface which is very easy to use and understand.

## *Appendix*

### **Acronyms**

CSS=Cascading Style sheet

DBMS= Database Management System

DHTML= Dynamic HTML

DOM= Dynamic Object Model

HTML=Hypertext Markup Language

JS=Java Script

PHP=Personal Home Page (recursive acronym=Hypertext Preprocessor)

RDBMS= Relational database management system

SPAM= [not an acronym] Slang or Unsolicited Commercial Mail

SQL= Standard Query Language

XAMP= Operating system(X) for APACHE, MySQL, PHP and PERL

## *Reference:*

- [1] <https://www.smartdraw.com/entity-relationship-diagram/>
- [2] <https://en.wikipedia.org/wiki/XAMPP>
- [3] <https://searchoracle.techtarget.com/definition/MySQL>
- [4] <https://www.w3schools.com/php/>
- [5] <https://www.w3schools.com/html/>
- [6] <https://www.w3schools.com/css/>
- [7] [https://www.w3schools.com/whatis/whatis\\_bootstrap.asp](https://www.w3schools.com/whatis/whatis_bootstrap.asp)
- [8] <https://en.wikipedia.org/wiki/JavaScript>
- [9] <https://en.wikipedia.org/wiki/JQuery>
- [10] <https://github.com/>
- [11] <http://brackets.io/>
- [12] [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)