

Final Report

(Bus Transportation System)

Course Code: CS254

Semester: B. Tech 4th Sem

Academic Year: 2020-21

Mr. Sharath Yaji

Course Title: DBMS Lab

Section: S2

Course Instructor: Dr. Annappa B and

Team Members:

1. LSV Sandeep Margana, 191CS226, 9652065499, mlsvsandeep@gmail.com
2. Tummala Ajay, 191CS259, 9100647787, ajaytummala99@gmail.com
3. Venkat Kumar Yenumula, 191CS263, 8688414344, venkatkumaryenumula18@gmail.com

1 Abstract

Brief Description:

Buses are the most frequently available and used for travelling. There are many types of buses made available in the market according to the class of the public. As per the needs of the public ,there are many features being provided in the buses .Such as AC coach, NON-AC coach,sleeper,semi sleeper e.tc. Apart from providing these facilities in their buses the problem here arises about the tickets i.e; everyone needs the tickets. But it's not possible. The buses which will register on our system will have the option to give their customers an option to select the seats of the bus according to their wish as many places depending on the availability. Making the system more useful unique tickets will be generated as its seats confirmed. Here we are with a system which can facilitate the users to which they can get their seats prior. As staff will decide the seat availability and here we have other users like the staff of the bus and driver with the owner who together collaborates in smooth functioning of the bus ticket booking system.

Key Features:

1. Login Facility
2. User can search details of the bookings, availability of buses, bus fare.

3. Ticket Booking
4. Admin can edit, add and delete the details of buses,fare of buses
5. Manage the information about Booking, buses, customers

Software Specifications:

- Frontend: CSS, JS, HTML, Python-Flask
- Backend: MySQL

2 Introduction

Transport management system project is developed to automate transport operation like booking order, select buses, user login information, contact etc. in a transport office. Using this system user can computerize transport office work like managing buses records, users details, admin details etc. The main job of the transporter is to provide transport service to consigner and consignee.

Transportation management system is a software application to maintain day to day transactions in transport office. Using this system user can manage transport work. He can select vehicle to transport. The seat numbers will be tentative so no need to select any buses. Customer can also book good transport order online. Proposed transport management system

Proposed system will automate all the main part of work done in existing transport systems like bookings and login facilities. It will store all the records . Using this system user can online check rates of buses available to the destination. Login and SignUp page values are associate with database, New User's details are inserted when a User creates a account. Admin can manage most of the values available in database but can edit only buses. This system provide the basic components of a shared information system to support the collaboration, bookings, availability, users.

Main modules of the project: This project has the following modules:

1. Home Page and About Page
2. Check Availability
3. Bookings
4. SignIn and SignUp
5. Contact Module
6. Admin Module
7. View and Edit Values

3 ER Diagram

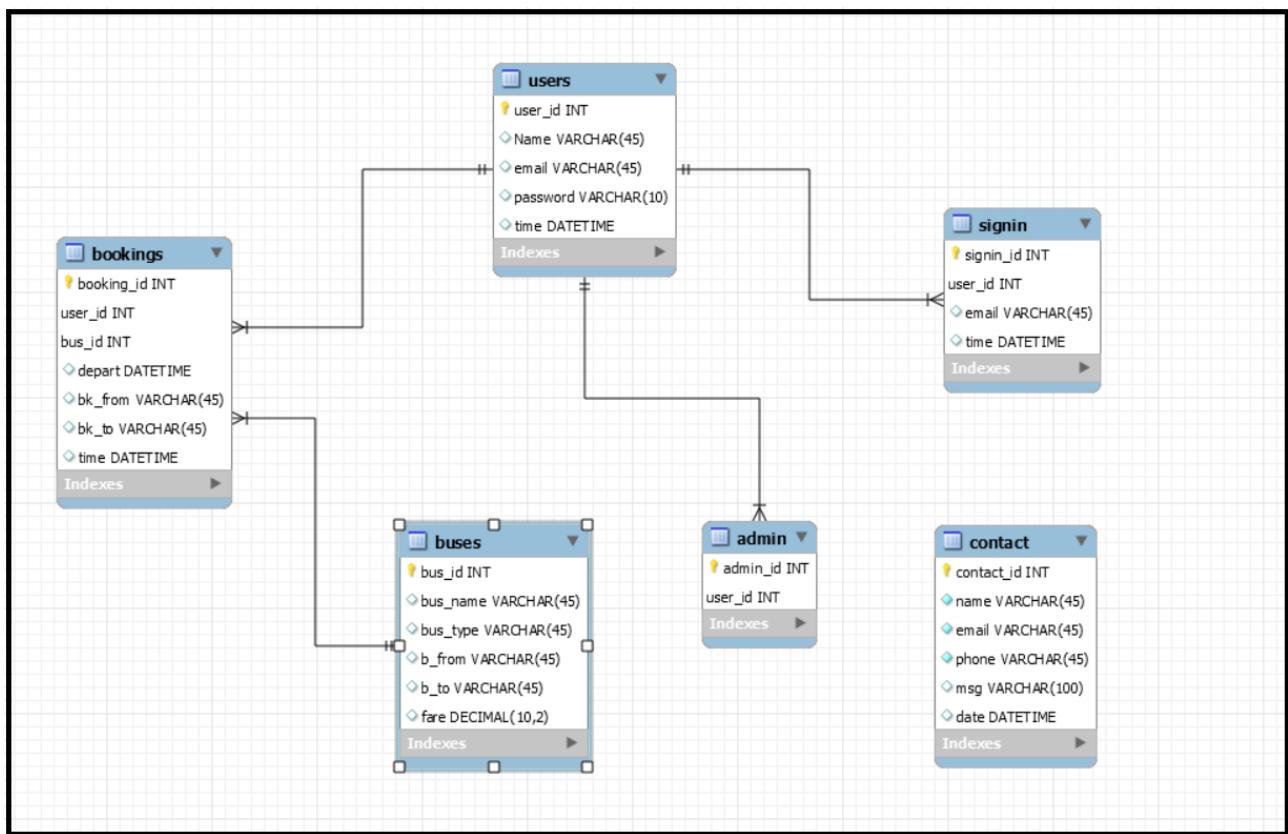


Figure 1: ER - Diagram of database

4 Source Code

Frontend: We created many pages corresponding to the modules, we provided with google drive like the static folder contains the CSS and JS files and Templates Folders contains the HTML files.

The HTML pages are: about.html, admin.html, bookings.html, contact.html, help.html, index.html, layout.html, invalid.html, login.html, post.html, signup.html, users.html.

Layout.html: This contains code of basic template of all the other pages, meaning this is the main code of jinja template inheritance.

```
<body>
  <!-- Navigation -->
  <nav class="navbar navbar-expand-lg navbar-light fixed-top" id="mainNav">
    <div class="container">
      <button class="navbar-toggler navbar-toggler-right" type="button"
        data-toggle="collapse" data-target="#navbarResponsive"
        aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
        navigation">
        Menu
        <i class="fas fa-bars"></i> </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="/">Home</a> //This links to the page index.html
            <li class="nav-item">
              <a class="nav-link" href="/about">About</a> // This links to the page about.html
            <a class="nav-link" href="/post">Book_Tickets</a> //here we can book tickets
            <li class="nav-item">
              <a class="nav-link" href="/contact">Contact</a> // links to the Contact.html
            <a class="nav-link" href="/signin">Signin</a> //links to the page sign.html
          </ul>
        </div>
      </div>
    </nav>
  {% block body %}
  {% endblock %}
```

Backend: In the backend we used MySQL, create the tables users, admin, bookings, buses with the appropriate foreign and primary keys Table creates:

```
users(user_id,name,email,password,time)
signin(signin_id,user_id,email,time)
buses(bus_id,bus_name,b_from,b_to,Fare)
bookings(booking_id,user_id,bus_id,depart,bk_form,bk_to,time)
admin(admin_id,user_id)
contact(contact_id,name,Email,phone,msg,date)
```

hit the link below to view SQL commands used:

https://drive.google.com/file/d/1xf86iJaaTiYAUkLsXiwsHx2PCGjE5F_5/view?usp=sharing

Below are the MySQL commands used in Python-Flask at different contexts

1. cur.execute("SELECT DISTINCT(b_from) FROM buses")
 - To give distinct option of from places available in buses in dropdown list
2. cur.execute("SELECT DISTINCT(b_to) FROM buses")
 - To give distinct option of to places available in buses in dropdown list
3. cur.execute("SELECT * FROM buses where b_from = %s and b_to = %s", (fromu,tou))
 - To select the list bus given by user with from and to places given
4. cur.execute("SELECT bus_id FROM buses where b_from = %s and b_to = %s and bus_name = %s", (fromu,tou,name))
 - After user selects a bus to know the bus_id of it we will execute this
5. cur.execute("SELECT user_id FROM signin WHERE time = (SELECT max(time) FROM signin)")
 - To know the current user of who logged in
6. cur.execute("INSERT into bookings(user_id,bus_id,depart,bk_from,bk_to) VALUES(%s,%s,%s,%s,%s)", (userid,busid,gettextime,fromu,tou))
 - to insert the details in bookings table
7. cur.execute("SELECT email,password,user_id from users")
 - to check email and password entered by user while signing in
8. cur.execute("SELECT user_id FROM users where email = %s", [email])
 - to get the user_id that matches with the email
9. cur.execute("INSERT INTO signin(user_id,email) VALUES (%s,%s)", (userid,email))
 - after checking id and password it will be listed in signin table
10. cur.execute("SELECT name FROM users WHERE email = (SELECT email FROM signin

where time = (SELECT max(time) from signin))”)

- To display the name of the person who signed in , on the home page

11. cur.execute(”INSERT INTO users(user_id,name,email,password)

VALUES(%s,%s,%s, %s”),(user_id,name,email,password))

- When a new user tries to create a an account

12. cur.execute(”INSERT INTO signin(user_id,email) VALUES(%s, %s”),(user_id,email))

- The created users value also need to be entered in signin table at the instance

13. cur.execute(”SELECT * FROM bookings where user_id in (SELECT user_id FROM signin WHERE time = (SELECT max(time) FROM signin))”)

- This is to show the all bookings made by the user who signin in

14. cur.execute(”INSERT INTO buses(bus_name,bus_type,b_from,b_to,fare)

VALUES(%s,%s,%s,%s,%s”),(name,bustype,bfrom,to,fare))

- When admin wants to add a new bus

15. cur.execute(”DELETE from buses where bus_id = %s”,(busid,))

- When admin wants to remove a bus

GitHub_Link: Below GitHub link comprises all codes related to our DATABASE MANAGEMENT SYSTEMS mini project (BUS TRANSPORT SYSTEM)

https://github.com/venkatkumar1810/Bus_transportation_System.git

DriveLink: This contains drive link of zip folder that contains the source code and files

https://drive.google.com/file/d/1_dO0MJ27EbjX6Jx0XahxqDajWT67D1D6/view?usp=sharing

5 Results

Book_Now
Book Your tickets Now! Enjoy The Journey.

Depart On:

From:

To:

Details of the buses, that we run and maintain:

Name	Type	From	To	Fare
VANNILA	NON-AC-SEATER	VIJAYAWADA	ARAKU	1500.00

Select the Bus, You wish to Travel:

Copyright © Mini Project By Venkat - Ajay - Sandeep - 2021

Figure 2: Booking Page

Bookings

Here you go, These are your booking details:

Depart On	From	To
2021-03-06 00:00:00	BANGALORE	KURNOOL
2021-03-27 00:00:00	VIJAYAWADA	ARAKU

Admin Page:

Details of the all the bookings:

Booking_id	User_id	Bus_id	Departure Date	From	To	Booking Time
201	101	303	2021-03-01 00:00:00	MANGALORE	BANGALORE	2021-03-06 16:47:25
203	102	301	2021-03-05 00:00:00	MANGALORE	BANGALORE	2021-03-08 11:04:50
210	103	307	2021-03-06 00:00:00	BANGALORE	KURNOOL	2021-03-08 11:25:36
213	104	309	2021-03-09 00:00:00	MANGALORE	GOA	2021-03-08 18:39:25
220	103	310	2021-03-27 00:00:00	VIJAYAWADA	ARAKU	2021-03-25 12:08:12

Database (Workbench)

booking_id	user_id	bus_id	depart	bk_from	bk_to	time
201	101	303	2021-03-01 00:00:00	MANGALORE	BANGALORE	2021-03-06 16:47:25
203	102	301	2021-03-05 00:00:00	MANGALORE	BANGALORE	2021-03-08 11:04:50
210	103	307	2021-03-06 00:00:00	BANGALORE	KURNOOL	2021-03-08 11:25:36
213	104	309	2021-03-09 00:00:00	MANGALORE	GOA	2021-03-08 18:39:25
220	103	310	2021-03-27 00:00:00	VIJAYAWADA	ARAKU	2021-03-25 12:08:12

Figure 3: Details reflected in Bookings, Admin pages and In Database

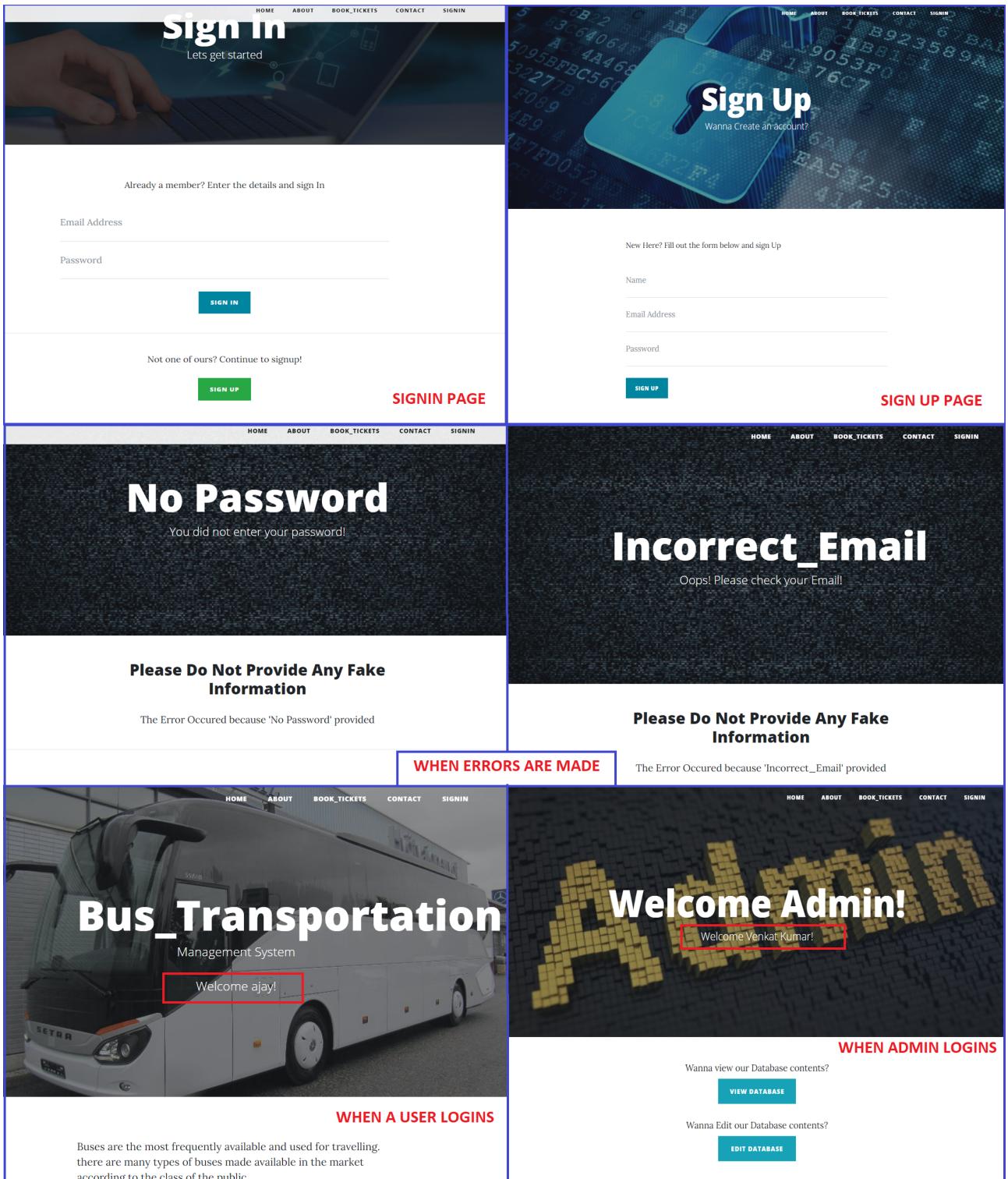


Figure 4: SignIn page module and its features

6 References:

1. <https://www.slideshare.net/ShamimAhmed36/bus-management-system-72297632>
2. <https://youtu.be/1SnPKhCdlsU>
3. <https://www.lovelycoding.org/bus-management-system/>

******* END *******