PROJECT REPORT

TOURISM DESKTOP APPLICATION

HOLIDAY HELPER

GROUP 15 - BINARY BRAINS

How objectives are met through the application

To provide an easy-to-use application that acts as a personal tour guide for tourists.

Tourists can use this application to search for places to visit, medical facilities and also they can rent vehicles using the system.

Help tourists plan their whole stay in the country during the period before arrival.

After login to the application users can go to make my plan section and customize a plan of their own so they can plan the stay before arrival.

Provide flexibility by re-computing an optimized schedule to accommodate changes in plans by users.

System is adapted to the changes made by the users.

Object Oriented Programming Concepts implemented in our application

We used Swing API and NetBeans IDE to develop our desktop application.

Class and object creation:

The javax.swing.JFrame class is used to extend the class. This class represents a GUI form. This code creates instances of the classes.

Constructors and methods:

Constructors have the same name as the class and don't have a return type.

- initComponents() method is responsible for initializing and setting up the GUI components, such as buttons, labels, text fields, etc., on the SignUp form.

Ex:

```
public SignUp() {
        initComponents();
 }
```

Inheritance:

Classes extend the javax.swing.JFrame class, inheriting its properties and behaviors. This is an example of inheritance, where a subclass inherits from a superclass (JFrame).

```
Ex: public class SignUp extends javax.swing.JFrame
    public class myplan extends javax.swing.JFrame
```

Encapsulation:

The code encapsulates the implementation details of classes within its own class definition. The variables, methods, and GUI components are declared within the class and have access modifiers to control their visibility and accessibility.

Ex:

```
        private javax.swing.JButton btnback;
        private javax.swing.JButton btnregister;
        public myplan()  ;
```

Polymorphism:

The code demonstrates polymorphism when invoking the invokeLater() method. It accepts an instance of the Runnable interface, which is implemented using an anonymous inner class (new Runnable() { ... }). This allows for the flexibility of passing different implementations of the Runnable interface.

Ex:

```
java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

             new SignUp().setVisible(true);

         }

});
```

Exception Handling:

The code includes try-catch blocks to handle potential exceptions like ClassNotFoundException and SQLException.

Ex:

```
catch (ClassNotFoundException ex) {
          JOptionPane.showMessageDialog(this, "Driver Not Found");
 } catch (SQLException ex) {
          JOptionPane.showMessageDialog(this, ex.getMessage());
  }
```

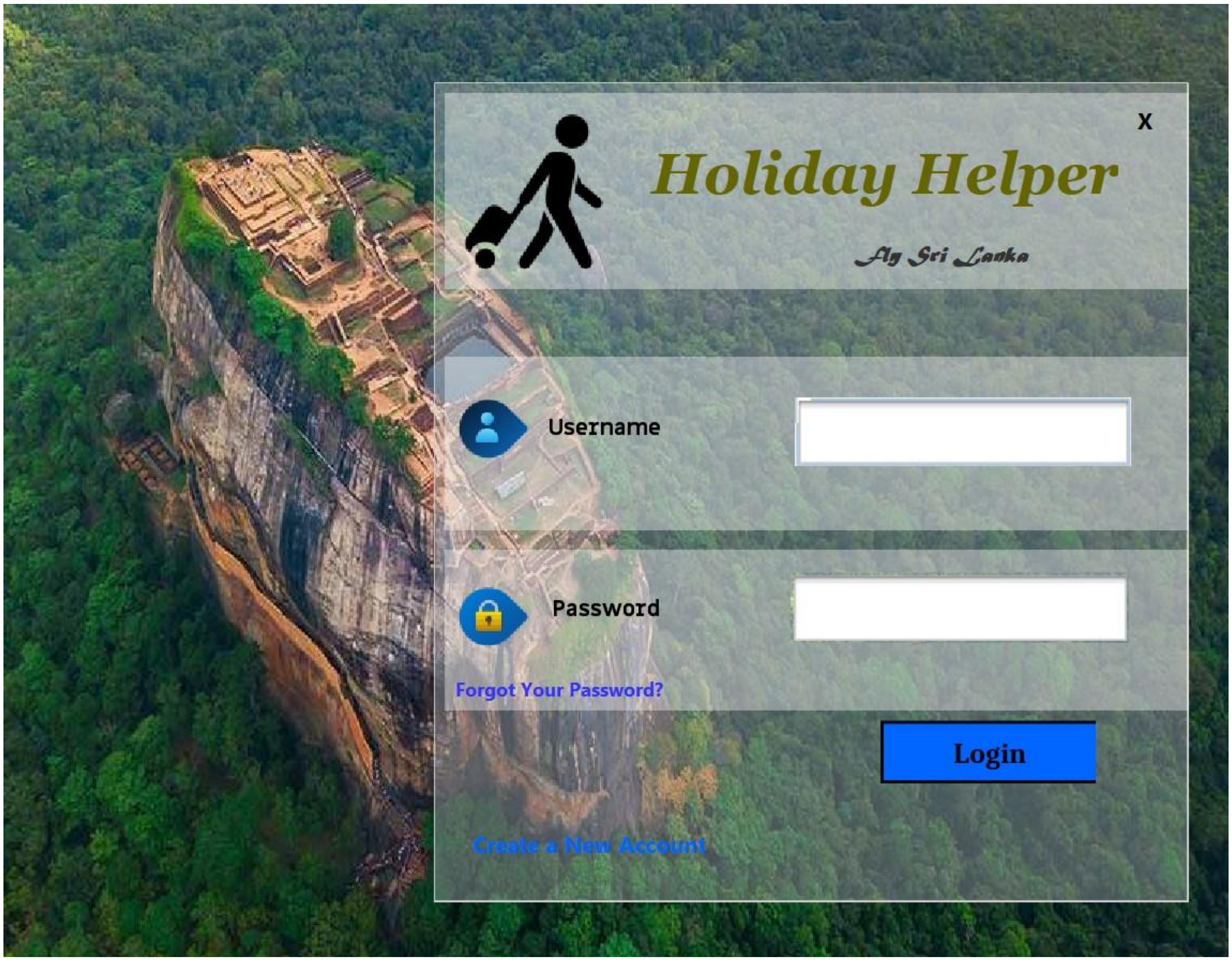
Interfaces and functionality

LOGIN PAGE

A login page is an important feature in software applications that require user authentication. It is typically the first point of interaction between the user and the system. In this introduction, I'll provide an overview of a login page and its functionality created using the NetBeans IDE in Java.

Creating a login page using NetBeans IDE in Java involves designing the user interface and implementing the necessary functionality. Here are the key steps involved when creating login page

1. Designing the User Interface: NetBeans IDE provides a drag-and-drop interface builder that allows you to create the visual elements of the login page. You can arrange components such as labels, text fields, buttons, and checkboxes to create a visually appealing and user-friendly login form.



2. Validating User Input: Once the user enters their credentials, the login page needs to validate the input before proceeding. This includes checking if the username and password fields are not empty**.**

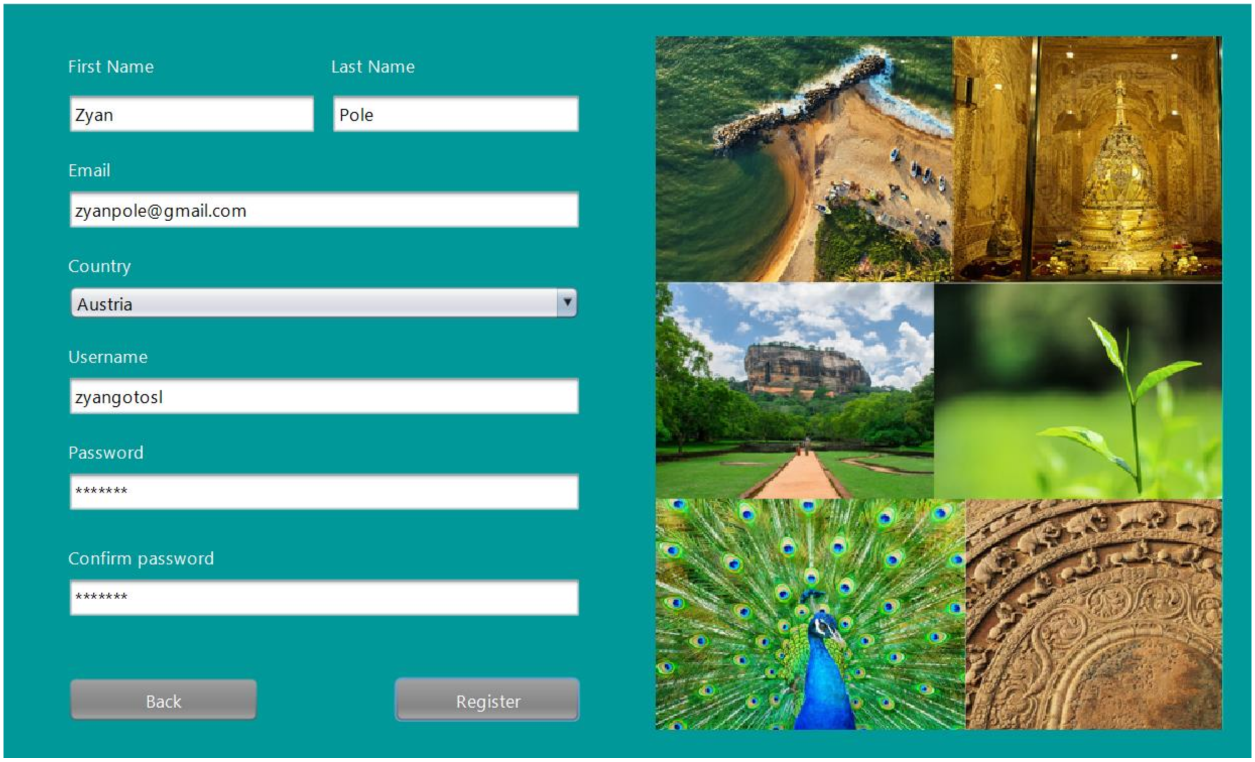
3. Authentication Process: After validating the user input, the login page interacts with the authentication mechanism to verify the entered credentials. This may involve querying a database or calling an authentication API to compare the provided username and password with the stored or authorized credentials.

4. Handling Authentication Results: Based on the authentication process, the login page responds accordingly. If the credentials are valid, the user is granted access to the application. Otherwise, an error message is displayed, indicating that the login attempt was unsuccessful or the fields are empty.

5. User Navigation: Upon successful authentication, the login page typically directs the user to the main interface of the application. It may also provide options for password recovery when you forget your password and account creation.

6. Event Handling: with the event handler we used some mouse functions on the button.For mouse click and mouse exit we used a button class and extended a button method to the button which is on the interface.

SIGNUP PAGE



First Name: Zyan

Last Name: Pole

Email: zyanpole@gmail.com

Country: Austria

Username: zyangotosl

Password: *******

Confirm password: *******

If the user has not registered before, the user can make his own account using the signup page. For that users have to enter the details to the given fields.

User Input Validation

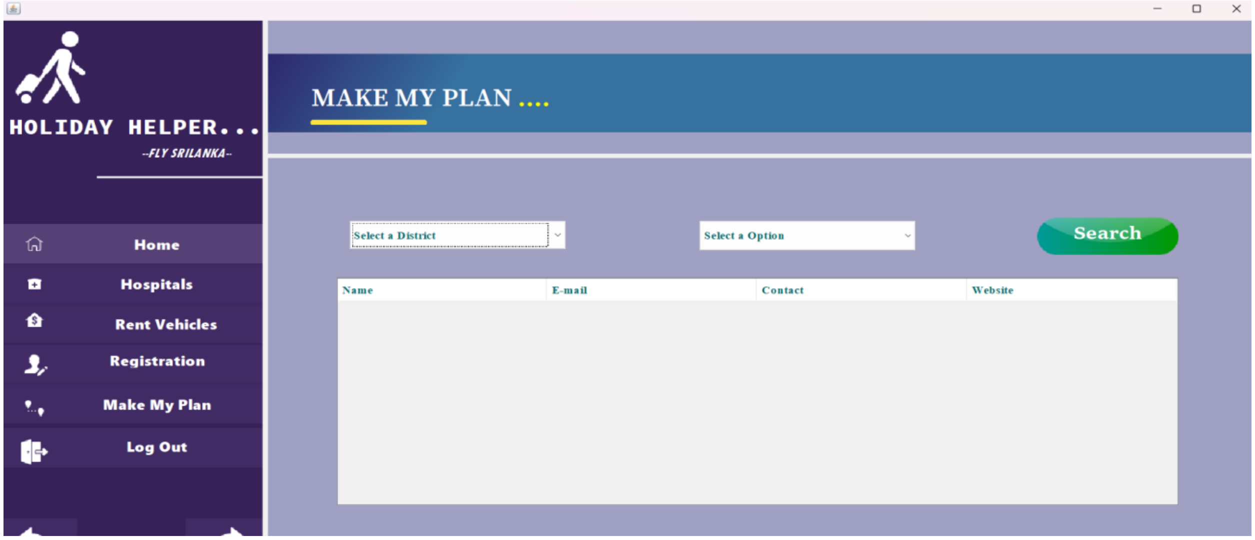
The code checks if the username and password fields are empty.

And also checks whether the password field and the confirm password field both have the same password entered.

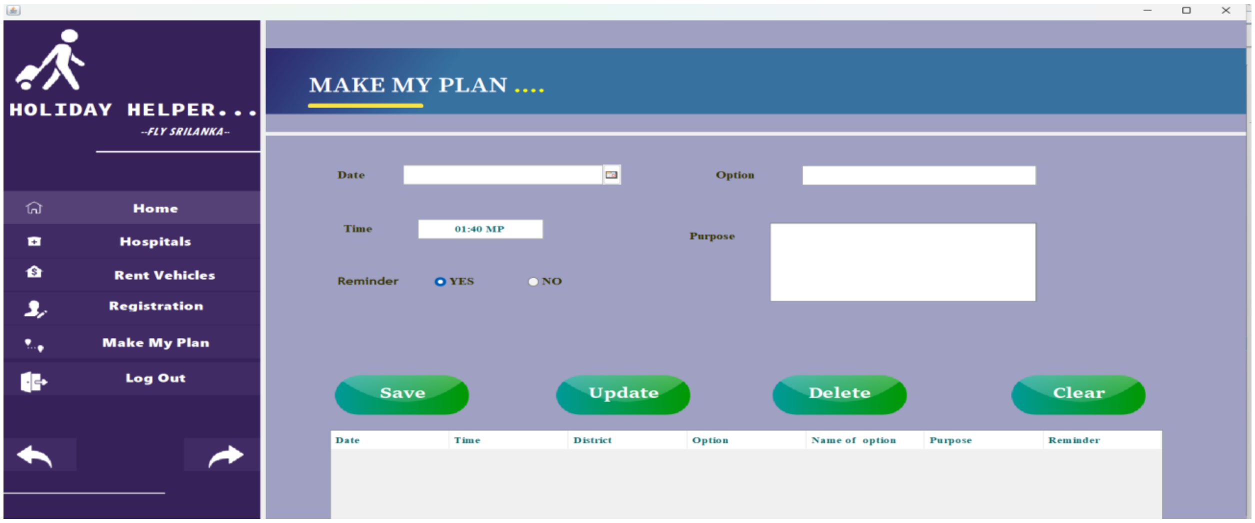
If any of these validations fail, the system will display error messages.

If all the details are filled out successfully, the record will be sent to the database, and a confirmation message will be displayed.

MAKE MY PLAN



Name	E-mail	Contact	Website



Date

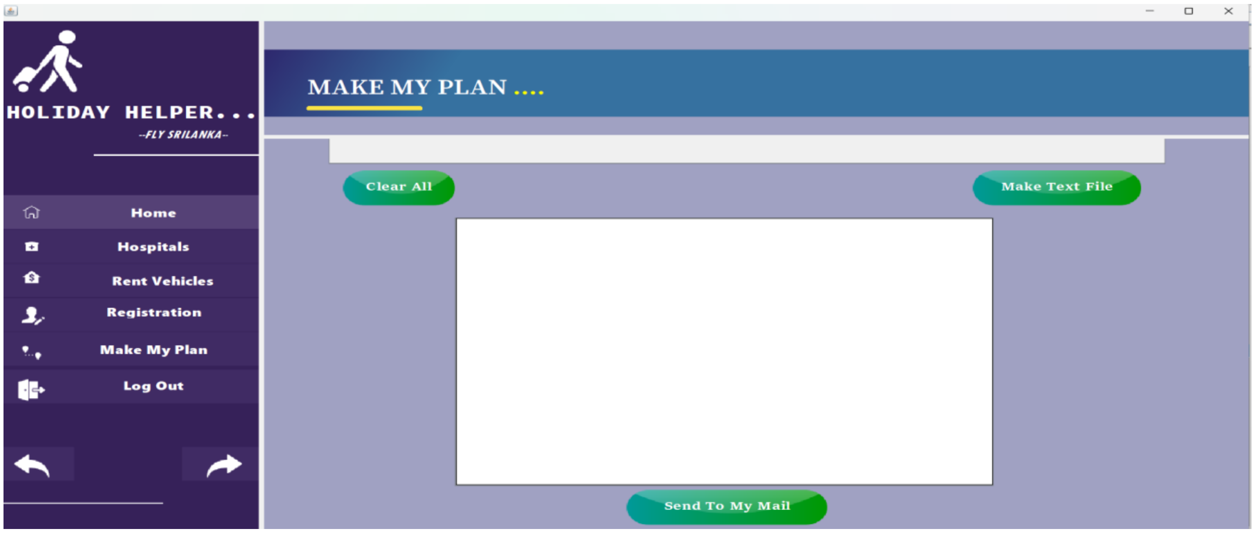
Option

Time 01:40 MP

Purpose

Reminder ● YES ○ NO

Date	Time	District	Option	Name of option	Purpose	Reminder



❖ The user interface of the Make my plan consists of the following components and layout:

- District Selection: Provide a JComboBox that allows users to select the district they want to visit.

- Option Selection: Display the available places within the selected district using a JComboBox. Users can choose the Options they want to visit.

- Search Button: When the user clicks the search button, retrieve data from the database and display it in a table. If there are any errors, show an error message to the user.

- Website Browsing: When the user clicks on a row in the table, open the corresponding website to get more details and paste the name of the place in an option text field. This text field should be read-only and cannot be directly edited by the user.

- Date and Time Selection: Utilize a JCalendar component or a Swing TimePicker library to allow users to select the date and time of their visit.

- Purpose and Reminder: Provide input fields where users can enter the purpose of their visit and set reminders for their planned activities.

- Day Plan Table: Create a separate table to display all the details of the trip plan. Users can edit, update, save, delete, or click on a row to populate the corresponding fields for further editing.

- Text Area and Text File Creation: Convert the details shown in the table into a text format and display it in a text area. Additionally, provide a button which is called as Make Text File to generate a text file containing the trip plan.

- Email Integration: Implement a separate interface where users can enter their email address, username, and password to send their day plan to their email.

- JButton Inheritance: Inherit JButton to extend its functionality, such as adding gradient effects or custom behaviors.

❖ Integration and Libraries:

- Use Java MySQL Connector library to connect the application to the MySQL database for data storage.

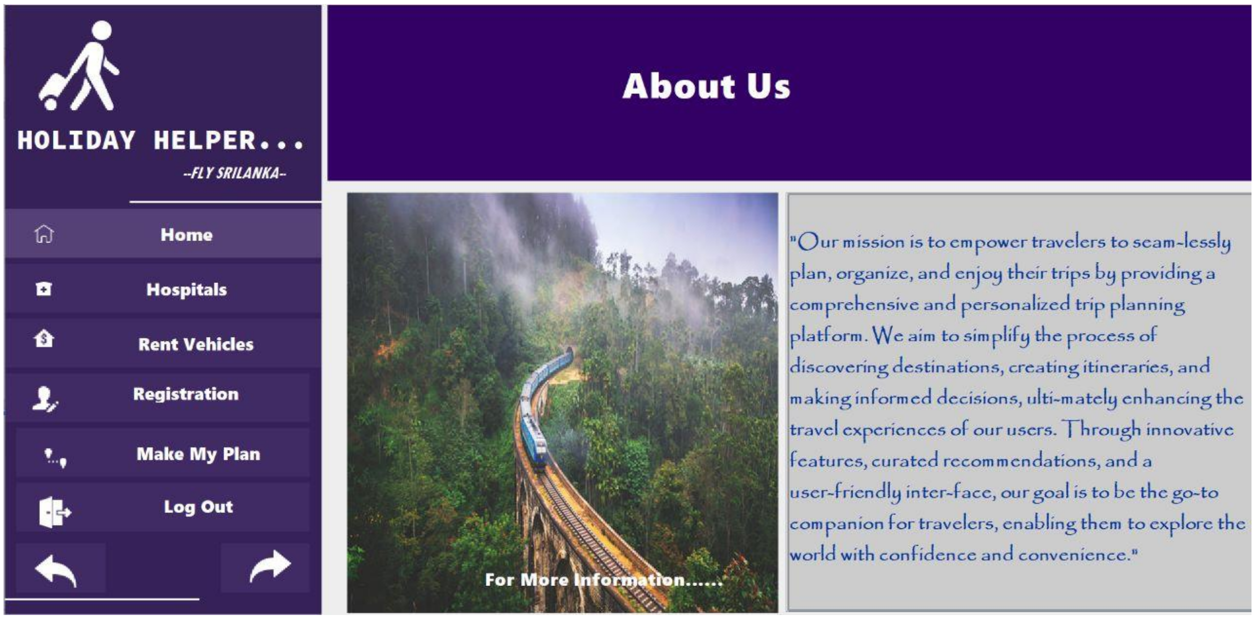
- Import and utilize the JCalendar component for date selection.

- Utilize the Swing TimePicker library for time selection.

- Implement email functionality using libraries such as JavaMail and the Activation Framework.

- Import and utilize the Timing Framework library to handle timing and animation effects

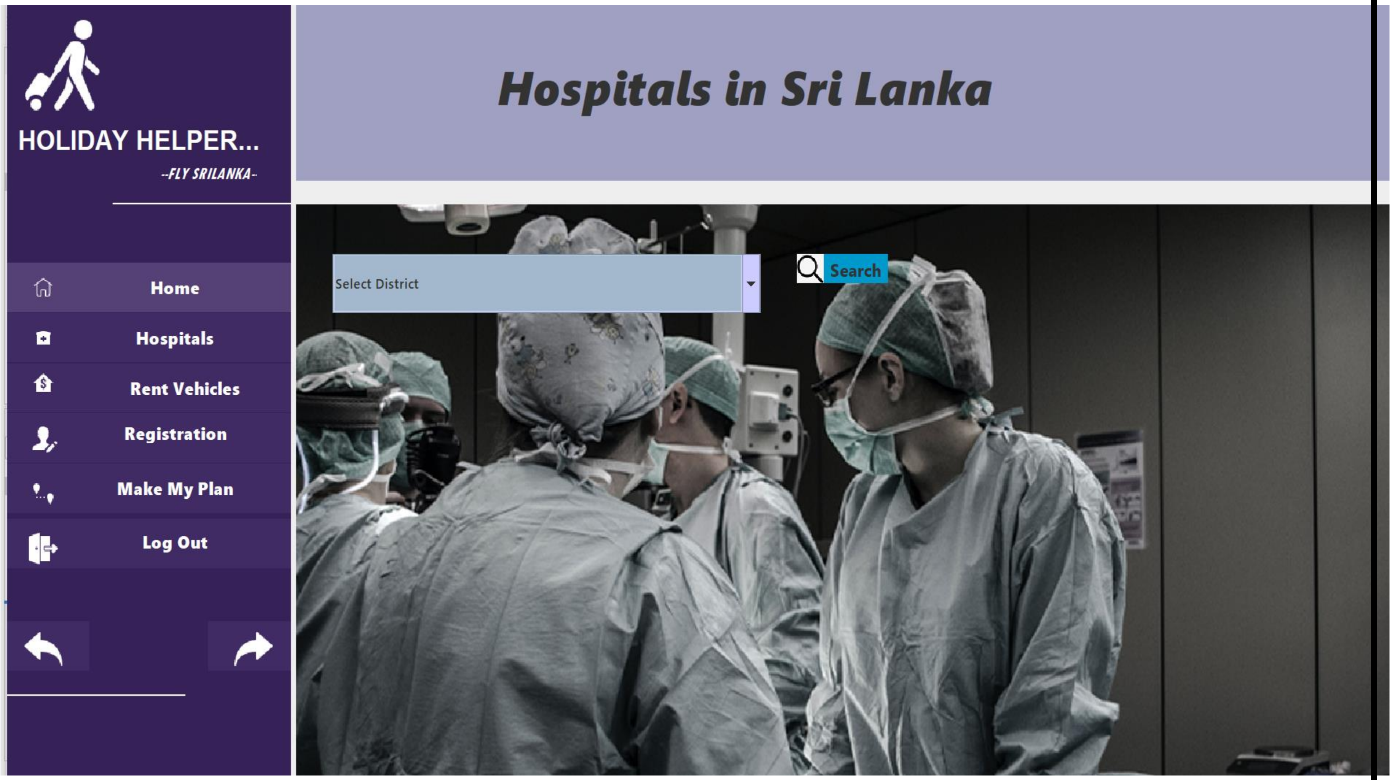
ABOUT US

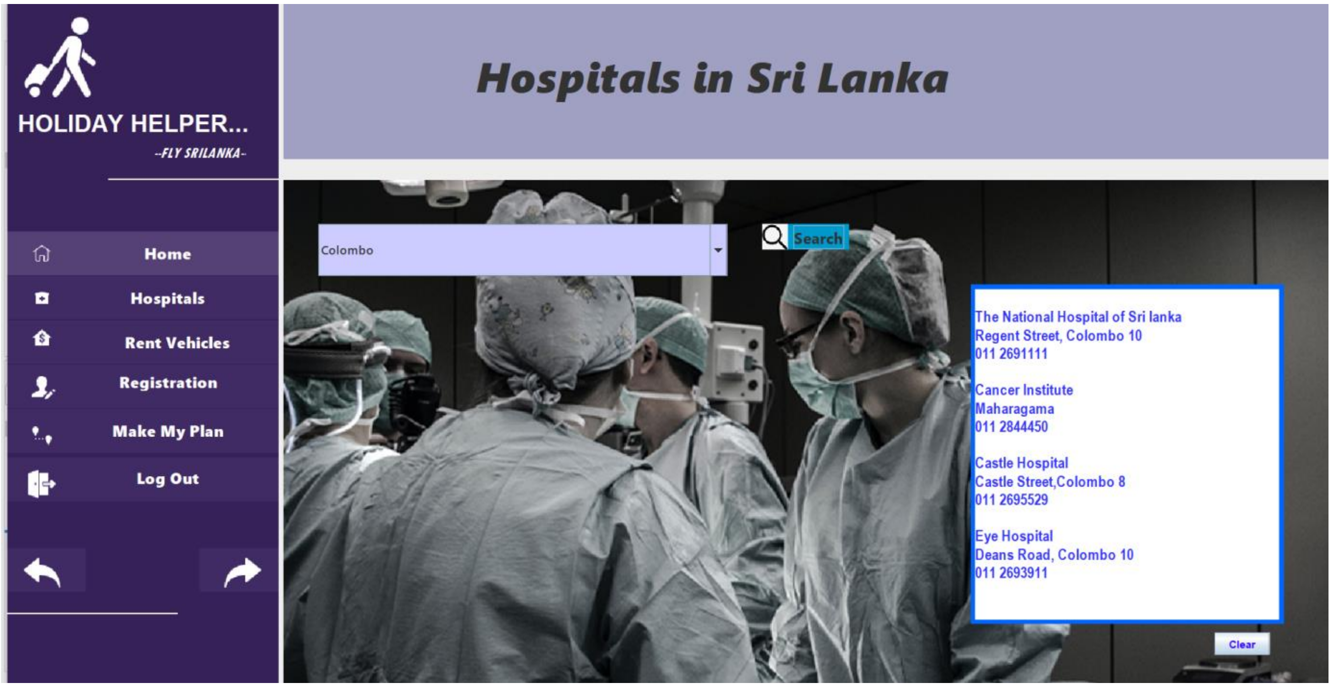


"Our mission is to empower travelers to seam-lessly plan, organize, and enjoy their trips by providing a comprehensive and personalized trip planning platform. We aim to simplify the process of discovering destinations, creating itineraries, and making informed decisions, ulti-mately enhancing the travel experiences of our users. Through innovative features, curated recommendations, and a user-friendly inter-face, our goal is to be the go-to companion for travelers, enabling them to explore the world with confidence and convenience."

- By using this ' About Us ' page the user can get some idea about our mission.
- We use the ' event handler ' function. The user can recognize that there is a link to get more information by using this, because if the user's mouse pointer enters the text section, the text section that says " For More information…… " will change color to light blue.
- And we use the ' click function ' for that text block. This click function helps the user to get more information from the Sri Lanka Tourism Board.

MEDICAL


HOLIDAY HELPER...
--FLY SRILANKA--
Home
Hospitals
Rent Vehicles
Registration
Make My Plan
Log Out
Hospitals in Sri Lanka
Select District
Search



Colombo

The National Hospital of Sri lanka
Regent Street, Colombo 10
011 2691111

Cancer Institute
Maharagama
011 2844450

Castle Hospital
Castle Street,Colombo 8
011 2695529

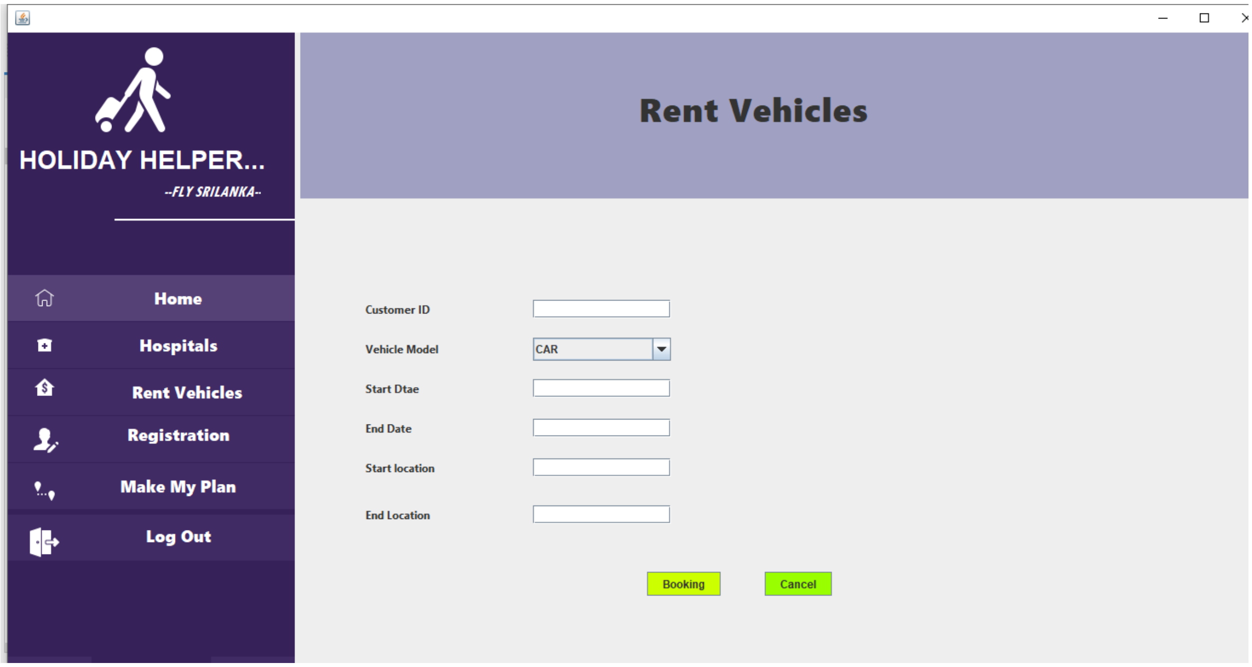
Eye Hospital
Deans Road, Colombo 10
011 2693911

We designed a medical information interface using Jframe. We used Jpanels, Jcombobox, Jlabel, Jbutton, Jscrollpane, Jtextarea.

If anyone wants to know details about hospitals in Sri Lanka. They can click on the district JComboBox and see the drop down list of all the districts of Sri Lanka and after selecting one district and click on the search button. Then all the details appear in the text area.

If you want to clear the text area you can click the clear button shown below the text area.

RENT VEHICLES



Customer ID

Vehicle Model CAR

Start Dtae

End Date

Start location

End Location

User can insert data and booking vehicle. First, customer should insert customer ID. After select vehicle type, insert start and end date, insert start and end location. Fill all of things and click booking button. After data goes to the database and booking vehicle. Click booking button after customer can see “Successful Booking” message.

Customer clicks the booking button without filling in any information, the customer can see a failed message.

When a customer clicks the cancel button, the rental vehicle screen is closed.

Java Packages imported

1) **java.sql.Connection package** - provides classes and interfaces that allow Java applications to connect to and interact with a relational database.

2) **java.sql.DriverManager package** - responsible for managing JDBC drivers and establishing database connections.

3) **java.sql.ResultSet package** - used to represent a set of results from a database query.

4) **java.sql.SQLException package** - used to handle exceptions that occur during database operations.

5) **java.sql.Statement package** - used to execute SQL statements and interact with a database. It provides methods to execute SQL queries, update statements, and retrieve result sets.

6) **javax.swing.JPanel package** - container component in Java Swing that is used to organize and group other components.

Database Connectivity

DbConnection.java class is created and drive_class, url , username, password are declared as private variables. We used *getters and setters* to access these *private variables.*

Loading the JDBC Driver:

Class.forName("...");

is used to load the MySQL JDBC driver class. This step is required to register the driver with the DriverManager before establishing a database connection.

Establishing a Connection:

DriverManager.getConnection(...) ;

is used to establish a connection to the MySQL database. It takes the database URL (jdbc:mysql://localhost:3306/userinfo), username (root), and password ("") as parameters. This method returns a Connection object that represents the connection to the database.

Creating a Statement:

connection.createStatement(...)

is used to create a Statement object. The Statement object allows you to execute SQL statements and interact with the database.

Executing an SQL Statement:

stm.executeUpdate(sql)

stm.executeQuery()

are used to execute an SQL statement.

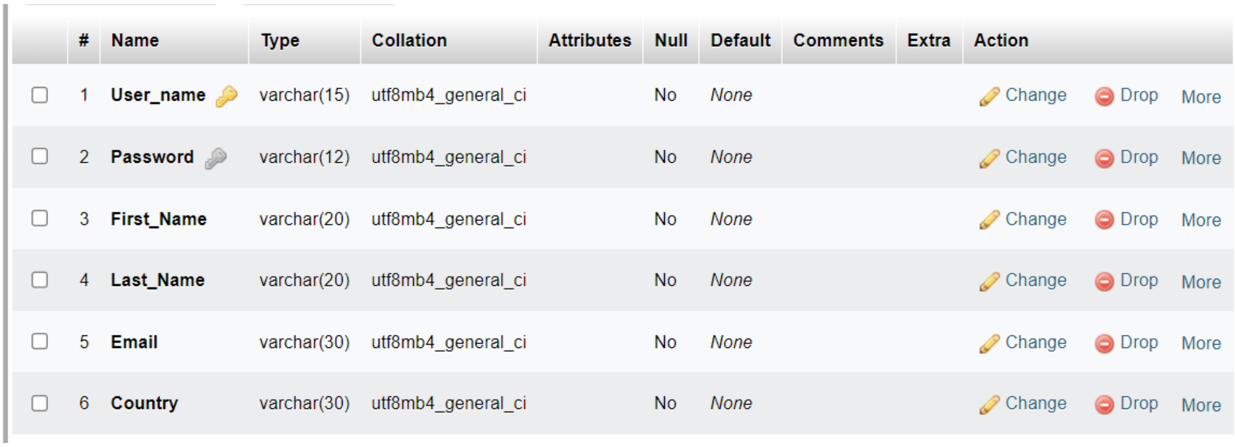
Database and tables structure

We used the xampp web server and phpmyadmin database management tool to create this database.

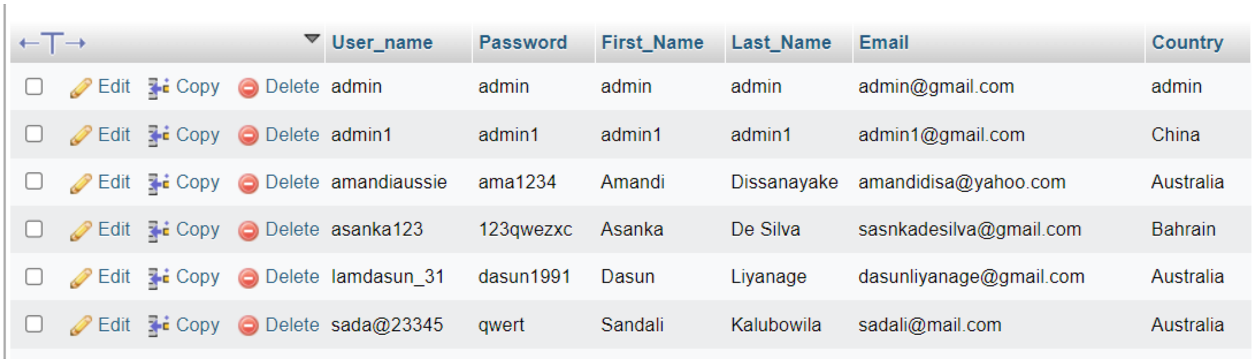
Database consists of four tables

login table structure

User_name field is the primary key field.



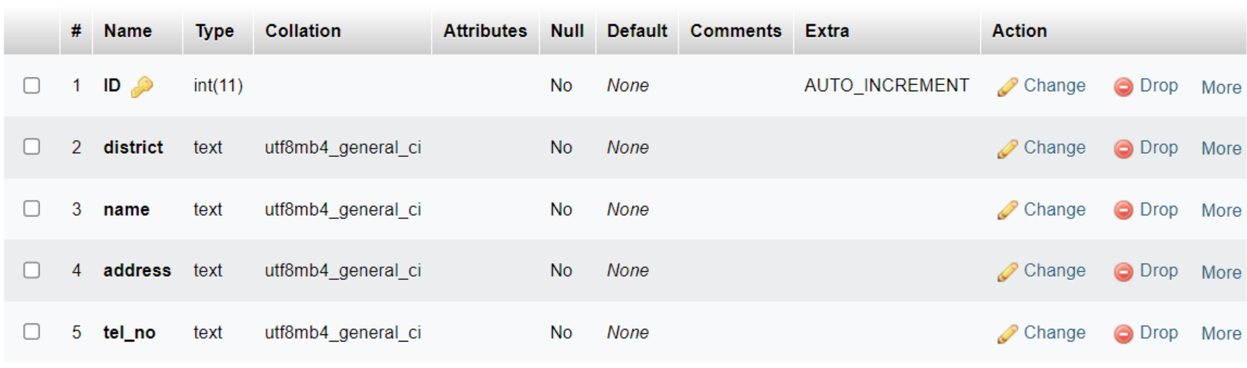
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	**User_name**	varchar(15)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	2	**Password**	varchar(12)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	3	**First_Name**	varchar(20)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	4	**Last_Name**	varchar(20)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	5	**Email**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	6	**Country**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More



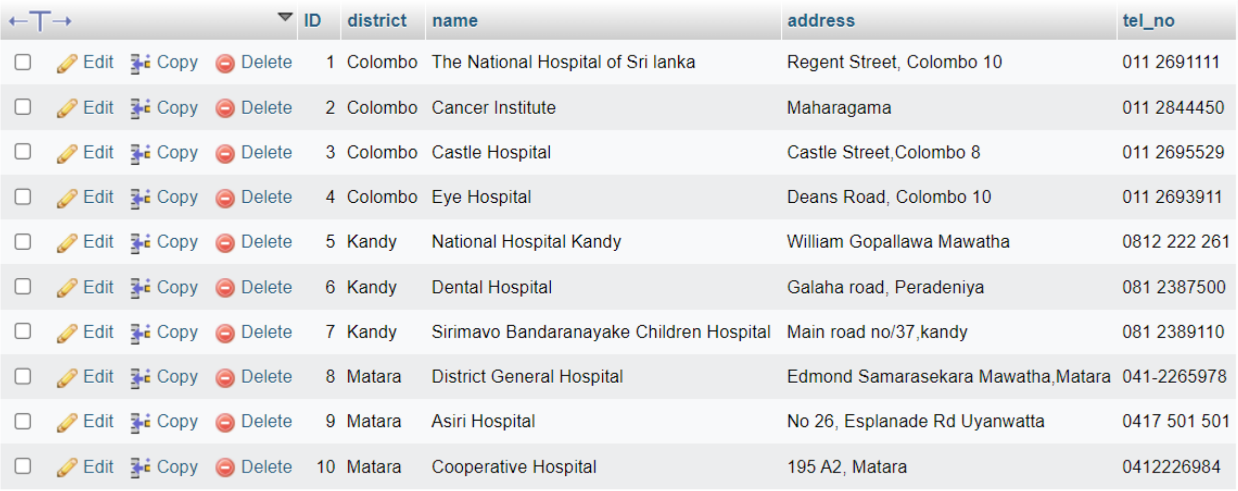
←T→				User_name	Password	First_Name	Last_Name	Email	Country
□	Edit	Copy	Delete	admin	admin	admin	admin	admin@gmail.com	admin
□	Edit	Copy	Delete	admin1	admin1	admin1	admin1	admin1@gmail.com	China
□	Edit	Copy	Delete	amandiaussie	ama1234	Amandi	Dissanayake	amandidisa@yahoo.com	Australia
□	Edit	Copy	Delete	asanka123	123qwezxc	Asanka	De Silva	sasnkadesilva@gmail.com	Bahrain
□	Edit	Copy	Delete	lamdasun_31	dasun1991	Dasun	Liyanage	dasunliyanage@gmail.com	Australia
□	Edit	Copy	Delete	sada@23345	qwert	Sandali	Kalubowila	sadali@mail.com	Australia

medicaltb table structure

The ID field is the primary key and it is set to auto increment.



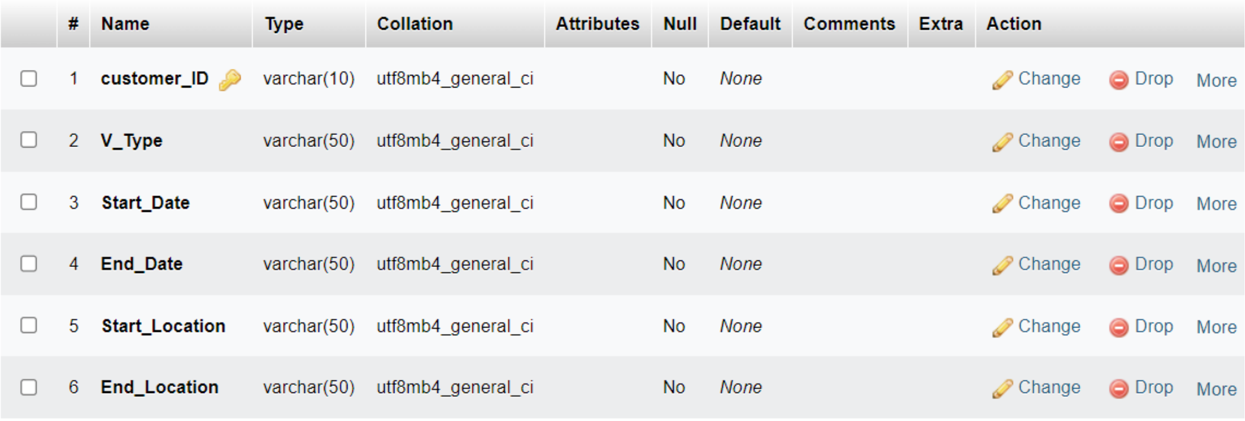
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	**ID**	int(11)			No	*None*		AUTO_INCREMENT	Change	Drop	More
□	2	**district**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	3	**name**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	4	**address**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	5	**tel_no**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More



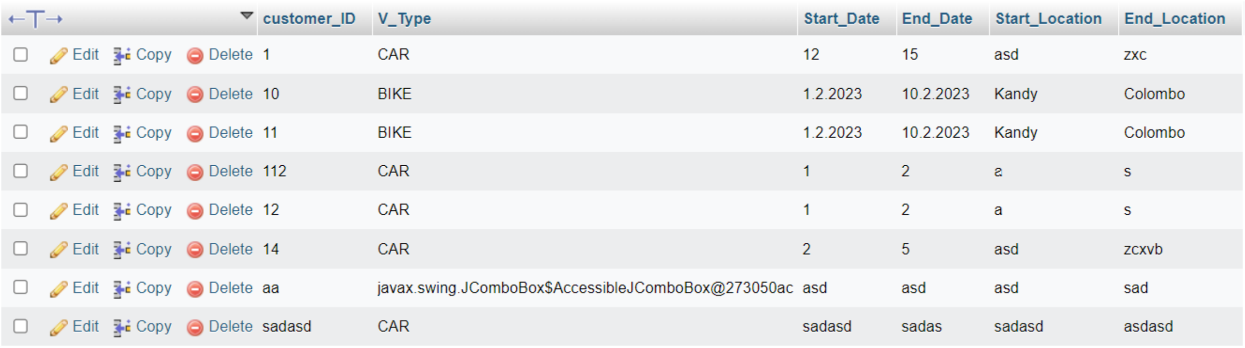
←T→	ID	district	name	address	tel_no
□ Edit Copy Delete	1	Colombo	The National Hospital of Sri lanka	Regent Street, Colombo 10	011 2691111
□ Edit Copy Delete	2	Colombo	Cancer Institute	Maharagama	011 2844450
□ Edit Copy Delete	3	Colombo	Castle Hospital	Castle Street,Colombo 8	011 2695529
□ Edit Copy Delete	4	Colombo	Eye Hospital	Deans Road, Colombo 10	011 2693911
□ Edit Copy Delete	5	Kandy	National Hospital Kandy	William Gopallawa Mawatha	0812 222 261
□ Edit Copy Delete	6	Kandy	Dental Hospital	Galaha road, Peradeniya	081 2387500
□ Edit Copy Delete	7	Kandy	Sirimavo Bandaranayake Children Hospital	Main road no/37,kandy	081 2389110
□ Edit Copy Delete	8	Matara	District General Hospital	Edmond Samarasekara Mawatha,Matara	041-2265978
□ Edit Copy Delete	9	Matara	Asiri Hospital	No 26, Esplanade Rd Uyanwatta	0417 501 501
□ Edit Copy Delete	10	Matara	Cooperative Hospital	195 A2, Matara	0412226984

vehicle table structure

customer_ID is the primary key field.



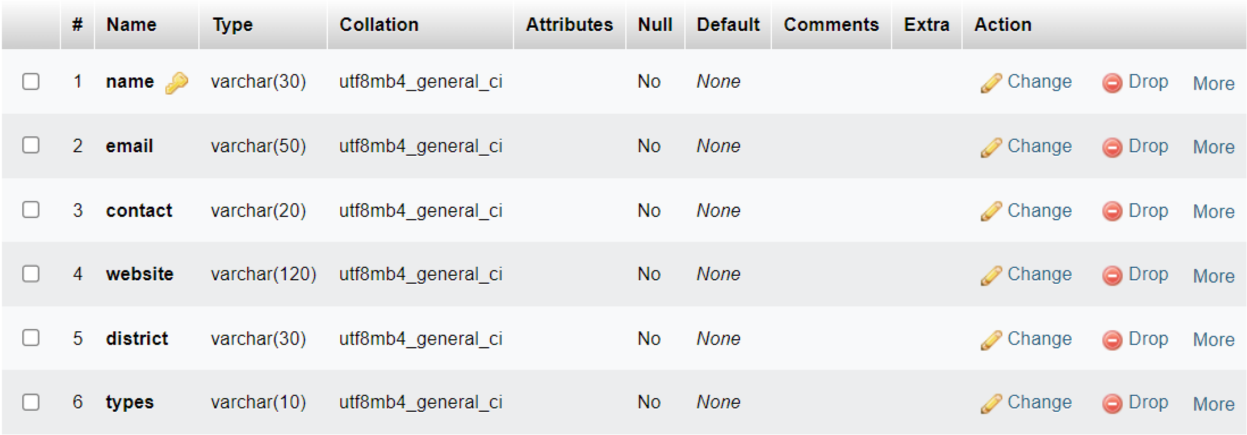
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**customer_ID**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	**V_Type**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**Start_Date**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**End_Date**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	**Start_Location**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	6	**End_Location**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More



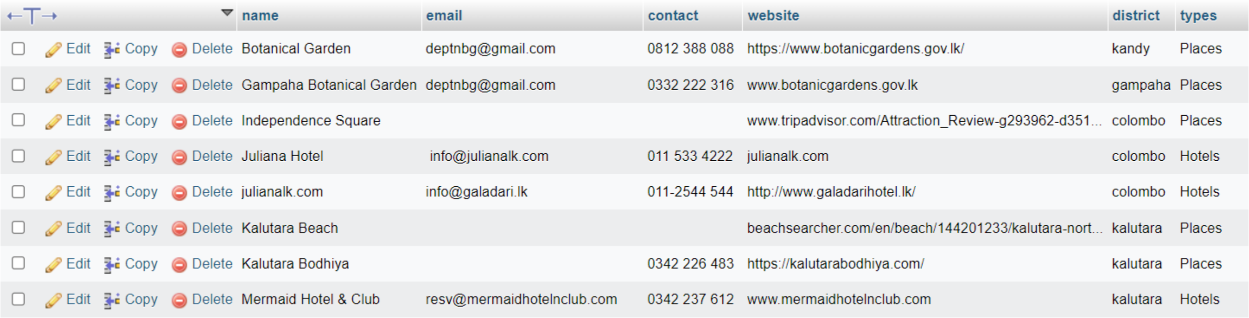
←T→	customer_ID	V_Type	Start_Date	End_Date	Start_Location	End_Location
□ Edit Copy Delete	1	CAR	12	15	asd	zxc
□ Edit Copy Delete	10	BIKE	1.2.2023	10.2.2023	Kandy	Colombo
□ Edit Copy Delete	11	BIKE	1.2.2023	10.2.2023	Kandy	Colombo
□ Edit Copy Delete	112	CAR	1	2	a	s
□ Edit Copy Delete	12	CAR	1	2	a	s
□ Edit Copy Delete	14	CAR	2	5	asd	zcxvb
□ Edit Copy Delete	aa	javax.swing.JComboBox$AccessibleJComboBox@273050ac	asd	asd	asd	sad
□ Edit Copy Delete	sadasd	CAR	sadasd	sadas	sadasd	asdasd

websites table structure

The name field is the primary key.



☐	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	**name** 🔑	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	2	**email**	varchar(50)	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	3	**contact**	varchar(20)	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	4	**website**	varchar(120)	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	5	**district**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
☐	6	**types**	varchar(10)	utf8mb4_general_ci		No	*None*			Change	Drop	More



←T→				name	email	contact	website	district	types
☐	Edit	Copy	Delete	Botanical Garden	deptnbg@gmail.com	0812 388 088	https://www.botanicgardens.gov.lk/	kandy	Places
☐	Edit	Copy	Delete	Gampaha Botanical Garden	deptnbg@gmail.com	0332 222 316	www.botanicgardens.gov.lk	gampaha	Places
☐	Edit	Copy	Delete	Independence Square			www.tripadvisor.com/Attraction_Review-g293962-d351...	colombo	Places
☐	Edit	Copy	Delete	Juliana Hotel	info@julianalk.com	011 533 4222	julianalk.com	colombo	Hotels
☐	Edit	Copy	Delete	julianalk.com	info@galadari.lk	011-2544 544	http://www.galadarihotel.lk/	colombo	Hotels
☐	Edit	Copy	Delete	Kalutara Beach			beachsearcher.com/en/beach/144201233/kalutara-nort...	kalutara	Places
☐	Edit	Copy	Delete	Kalutara Bodhiya		0342 226 483	https://kalutarabodhiya.com/	kalutara	Places
☐	Edit	Copy	Delete	Mermaid Hotel & Club	resv@mermaidhotelnclub.com	0342 237 612	www.mermaidhotelnclub.com	kalutara	Hotels

GROUP MEMBERS

R. Mohamed Sabith	-	SC/2020/11703
K.H.M.K D.R Herath	-	SC/2020/11719
Senanayake A.I.M	-	SC/2020/11693
A.F.M. Riyas	-	SC/2020/11690
R.A.D.S Ranasinghe	-	SC/2020/11698
G.A.D.S.K De Alwis	-	SC/2020/11680

END