PROJECT REPORT

BIKE RENTAL MANAGEMENT SYSTEM

Prepared By:

Kirti (IIT2020142)

Abhinandan (IIT2020119)

Shivam Harjani (IIT2020121)

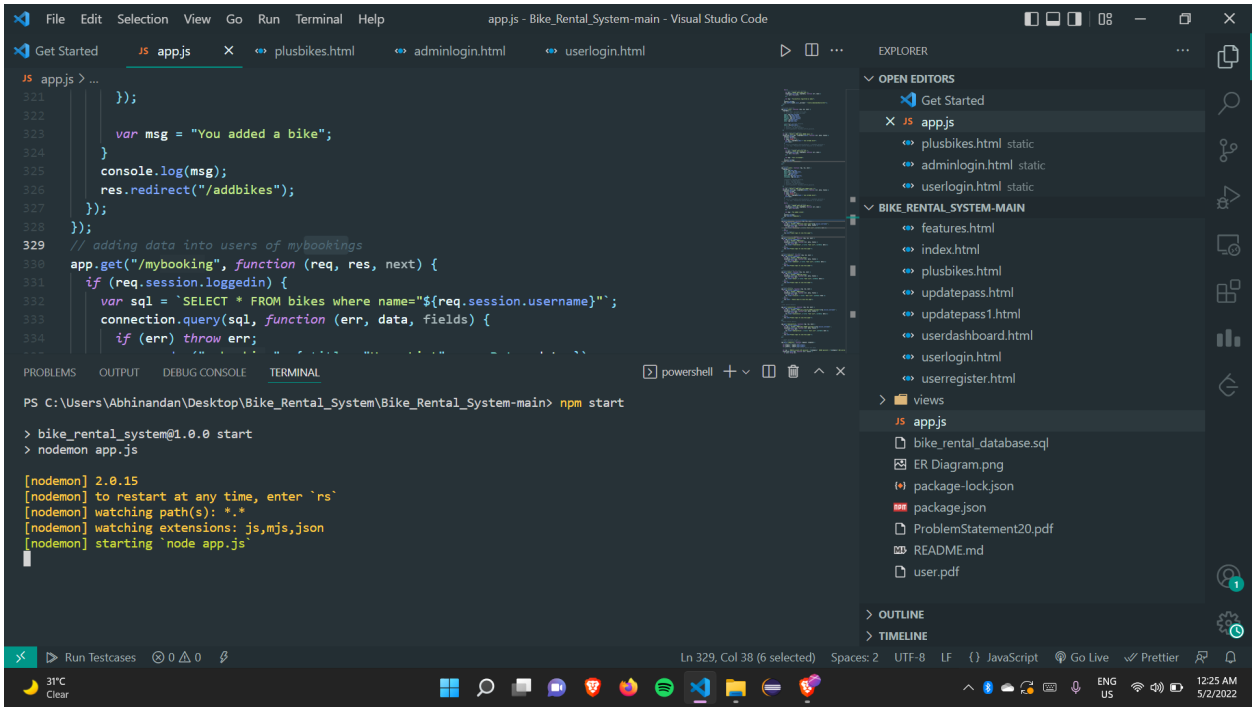
Aditi (IIT2020138)

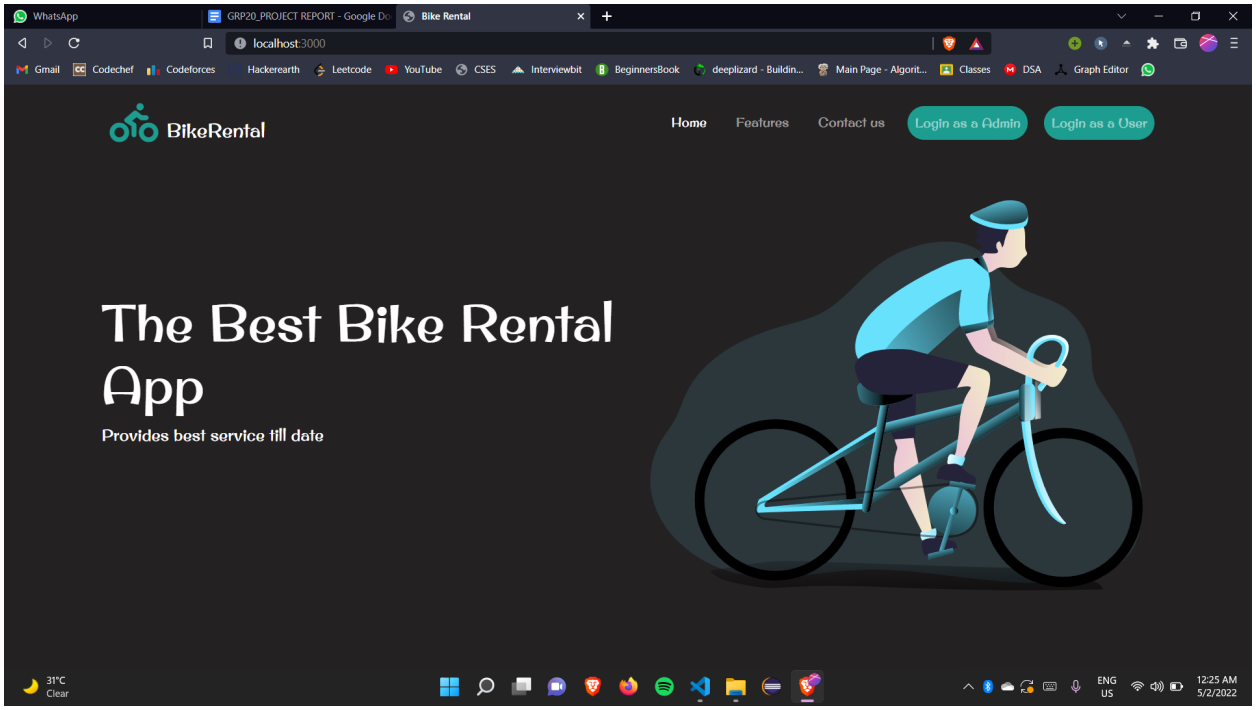
Introduction

An online bike rental system that allows customers to search for available rental bikes in a more efficient manner than the old one, saving time and energy. Admin and User are the two parts that make up this renting system. Admins can go in to add, amend, and delete information on bikes, as well as update the list of them. He or she has access to all users' bookings and may manage their bookings and transactions. Users may create an account on the website, log in, check the availability of bikes, book the bike of their preference, and pay for it.

How to run

1. Run the bike_rental_database.sql in MySql workbench
2. Change the password in app.js to your database password.
3. run command npm start in your terminal
4. Your project will run on http://localhost:3000/





Functionalities

- **Admin**

1. Login
2. Update bike information
3. Manage invoice
4. Provide transactions
5. Change of passwords
6. Update invoice
7. Access to all bookings
8. Updating details

- **User**

1. Login
2. Booking a bike

3. Update self credentials
4. Canceling a bike
5. Receive invoice
6. Receive transaction
7. Change password
8. Selecting bike as per of given type

Entities along with the schema

1. **Admin**

   Attributes:

id	PRIMARY KEY,
username	CANDIDATE KEY,
password ,	
email	

2. **User**

   Attributes:

id	PRIMARY KEY,
username	CANDIDATE KEY,
contact number,	
password,	
license	CANDIDATE KEY,
email,	
address	

3. **Bike**

   Attributes:

id	PRIMARY KEY,
name,	
cost,	
feature,	
model,	
description,	
image	

4. **Bookings**

Attributes:

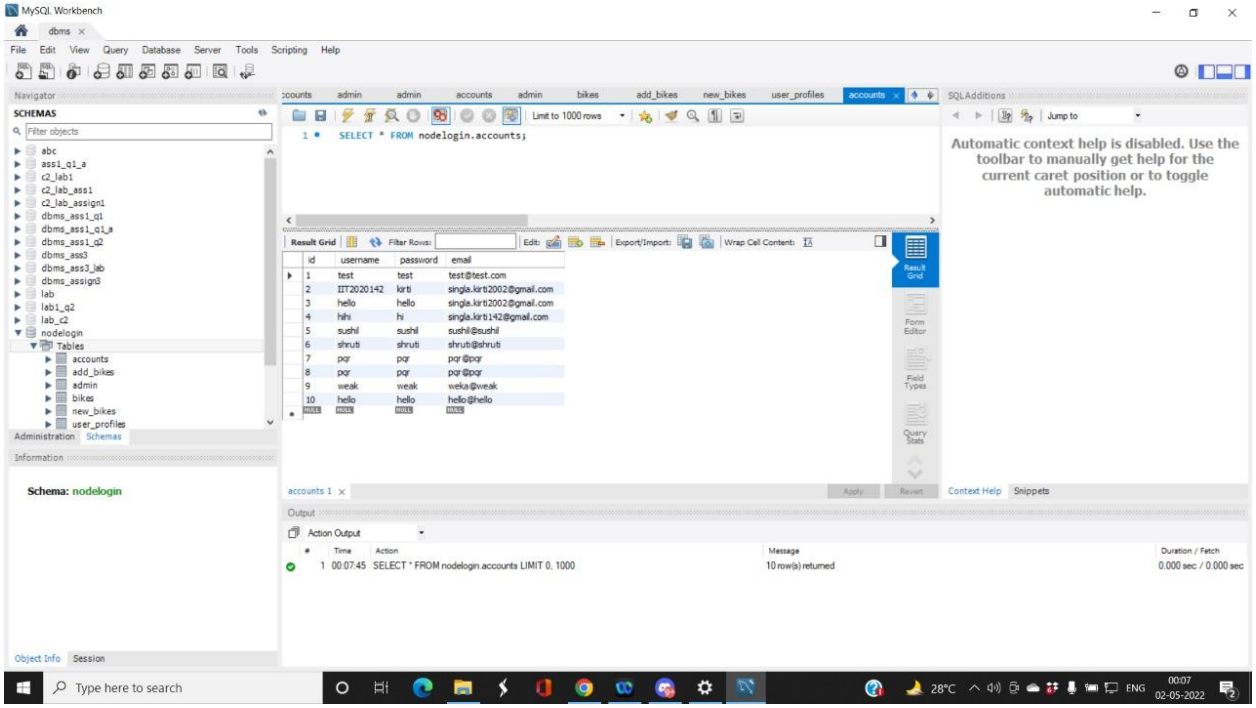
id	PRIMARY KEY,
username	FOREIGN KEY references user(username),
password,	
email	

Relationships

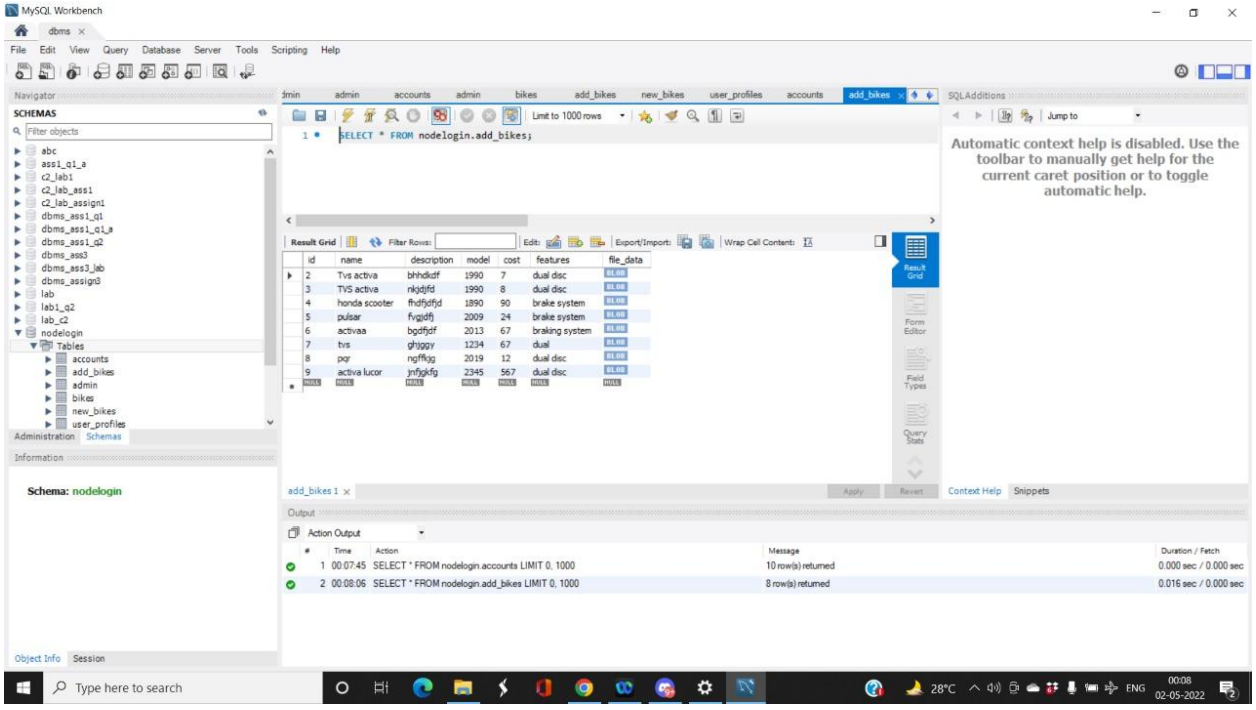
1. Admin to User
2. User to Bikes
3. User to Bookings
4. Admin to Bookings
5. Admin to Bikes
6. Bookings to Bikes

Tables

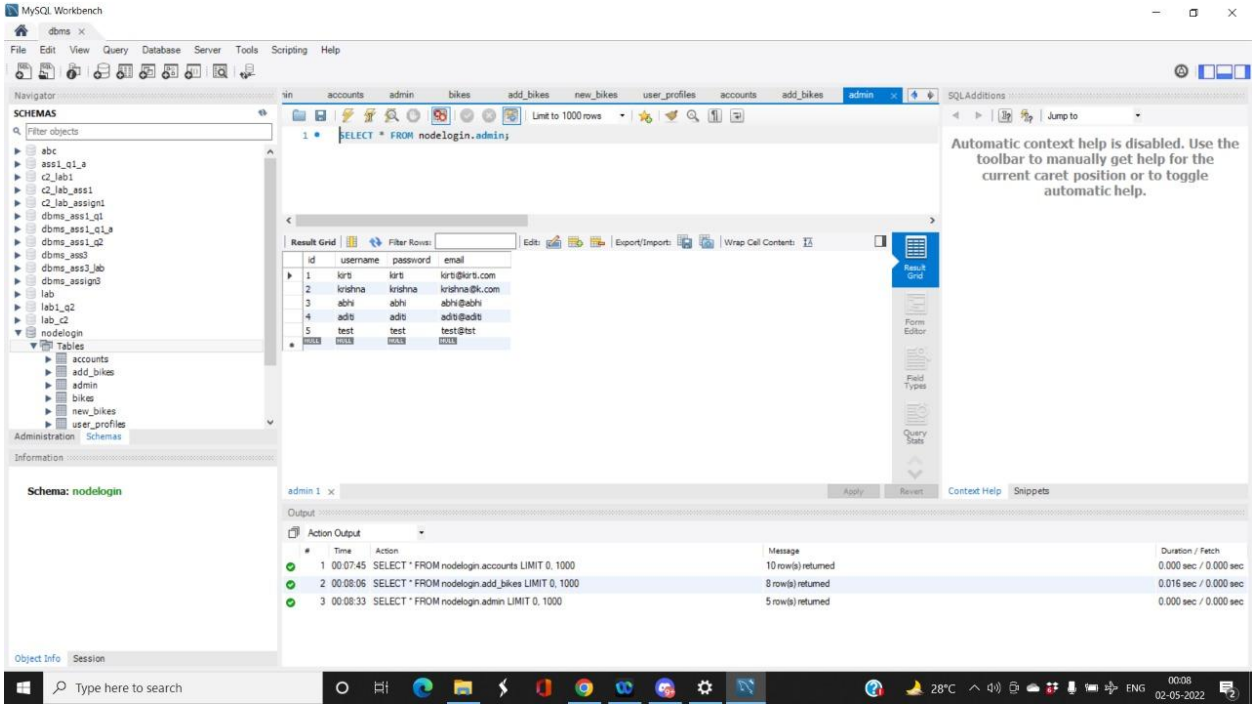
1. Admin accounts table



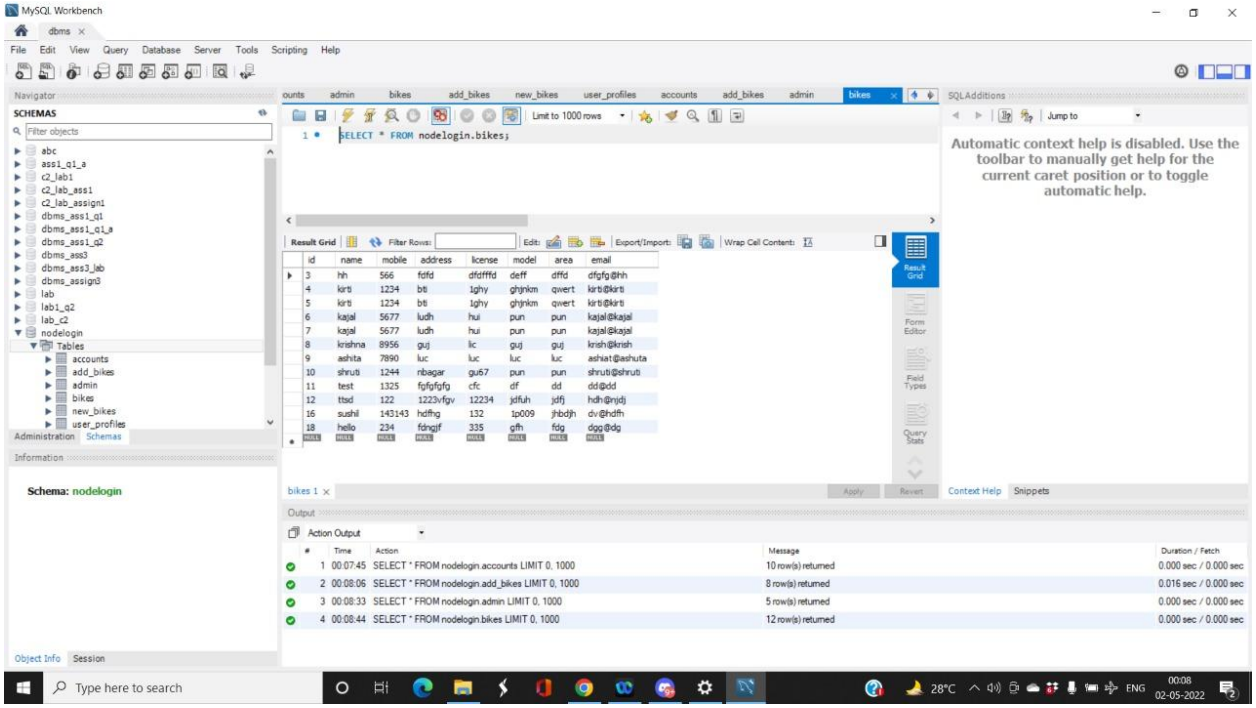
2. Bikes table



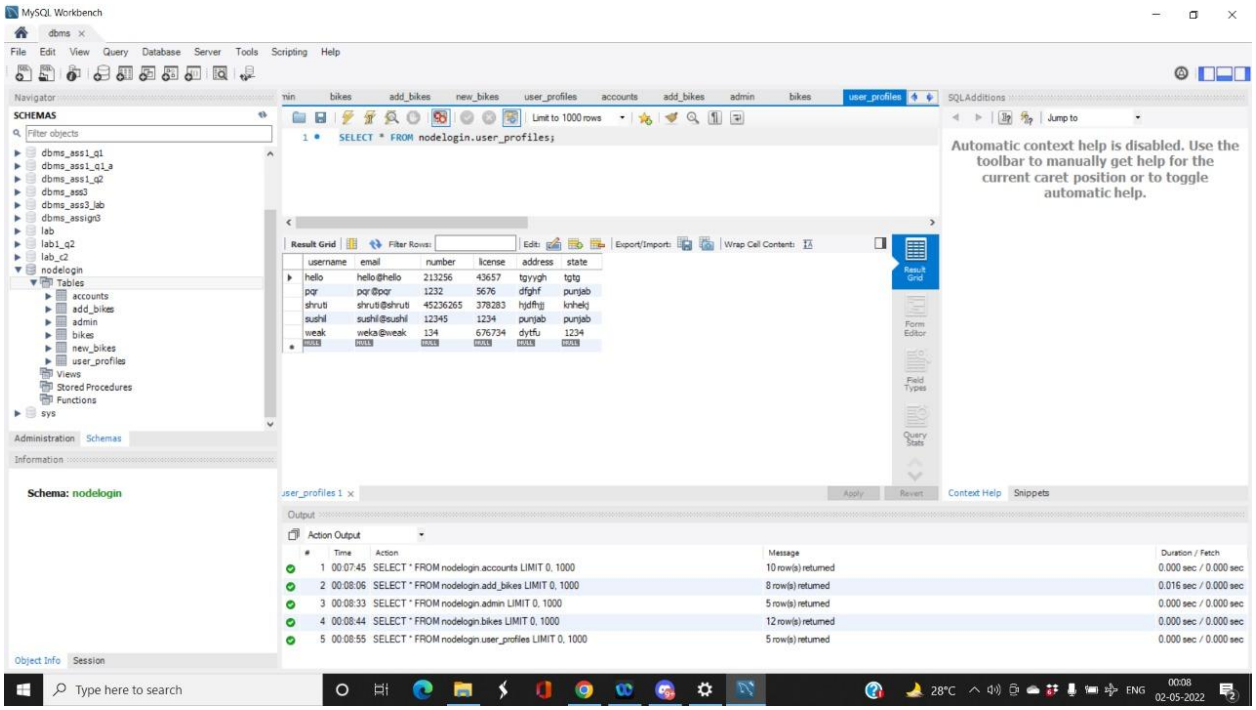
3. User accounts table



4. Bookings table



5. User profile table



As taking into account the consideration of functional dependencies between different attributes, typically between a primary key and a non primary key. The tables used in our project contain only atomic values, proving them in 1NF form. And the repeated occurrences have also been removed as all non primary keys are directly dependent on the primary key, so it's in 2NF form. And as there is no partial dependency so it is in 3NF form.

SQL Code:

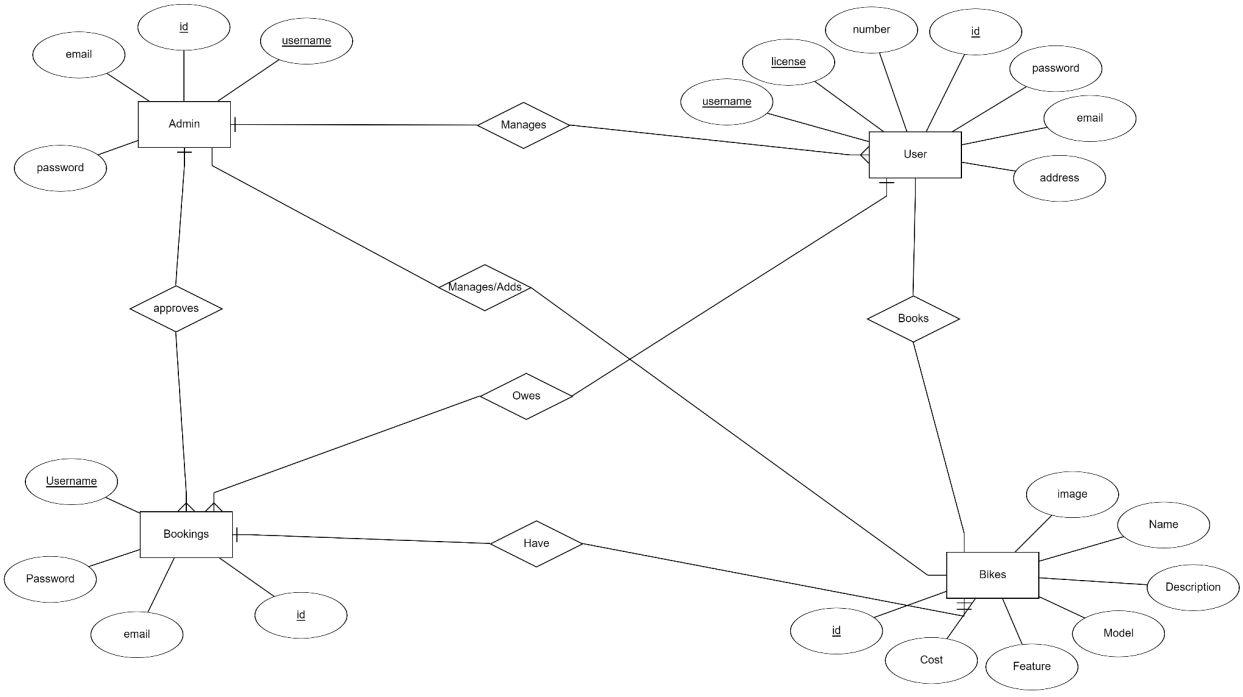
```
-- CREATE DATABASE IF NOT EXISTS `nodelogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `nodelogin`;

-- CREATE TABLE IF NOT EXISTS `accounts` (
--  `id` int(11) NOT NULL AUTO_INCREMENT,
--  `username` varchar(50) NOT NULL,
--  `password` varchar(255) NOT NULL,
```

```
--   `email` varchar(100) NOT NULL,
--   PRIMARY KEY (`id`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES
(1, 'test', 'test', 'test@test.com');
-- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'kirti123#';
-- CREATE TABLE IF NOT EXISTS `admin` (
--   `id` int(11) NOT NULL AUTO_INCREMENT,
--   `username` varchar(50) NOT NULL,
--   `password` varchar(255) NOT NULL,
--   `email` varchar(100) NOT NULL,
--   PRIMARY KEY (`id`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- INSERT INTO `admin` (`id`, `username`, `password`, `email`) VALUES (1,
'kirti', 'kirti', 'kirti@kirti.com');

-- CREATE TABLE IF NOT EXISTS `bikes` (
--   `id` int(11) NOT NULL AUTO_INCREMENT,
--   `username` varchar(50) NOT NULL,
--   `password` varchar(255) NOT NULL,
--   `email` varchar(100) NOT NULL,
--   PRIMARY KEY (`id`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- CREATE TABLE IF NOT EXISTS `bikes` (
```

```
--   `id` int(11) NOT NULL AUTO_INCREMENT,
--   `name` varchar(50) NOT NULL,
--   `mobile` int(20) NOT NULL,
--    `address` varchar(100) NOT NULL,
--    `license` varchar(100) NOT NULL,
--    `state` varchar(100) NOT NULL,
--    `area` varchar(100) NOT NULL,
--   `email` varchar(100) NOT NULL,
--   PRIMARY KEY (`id`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- SELECT * FROM bikes where name="test";

-- CREATE TABLE IF NOT EXISTS `add_bikes` (
--   `id` int(11) NOT NULL AUTO_INCREMENT,
--   `name` varchar(50) NOT NULL,
--
--    `description` varchar(200) NOT NULL,
--    `model` int(100) NOT NULL,
--    `cost` int(100) NOT NULL,
--    `features` varchar(200) NOT NULL,
--   `file_data` MEDIUMBLOB NOT NULL,
--   PRIMARY KEY (`id`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- CREATE TABLE IF NOT EXISTS `user_profiles` (
--
--   `username` varchar(50) NOT NULL,
```

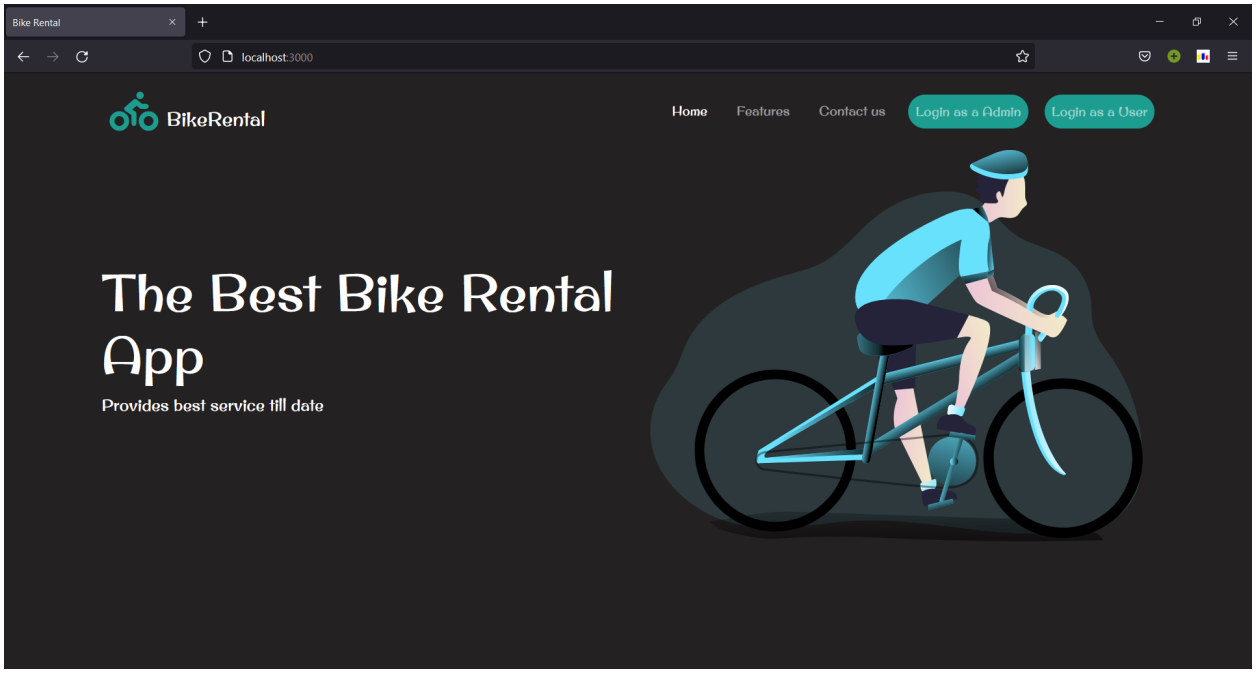
```
--    `email` varchar(200) NOT NULL,
--    `number` int(100) NOT NULL,
--    `license` int(100) NOT NULL,
--    `address` varchar(200) NOT NULL,
--    `state` varchar(200) NOT NULL,
--
--   PRIMARY KEY (`username`)
-- ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

ALTER TABLE bikes
RENAME COLUMN state to model;
```

Entity-Relationship Diagram

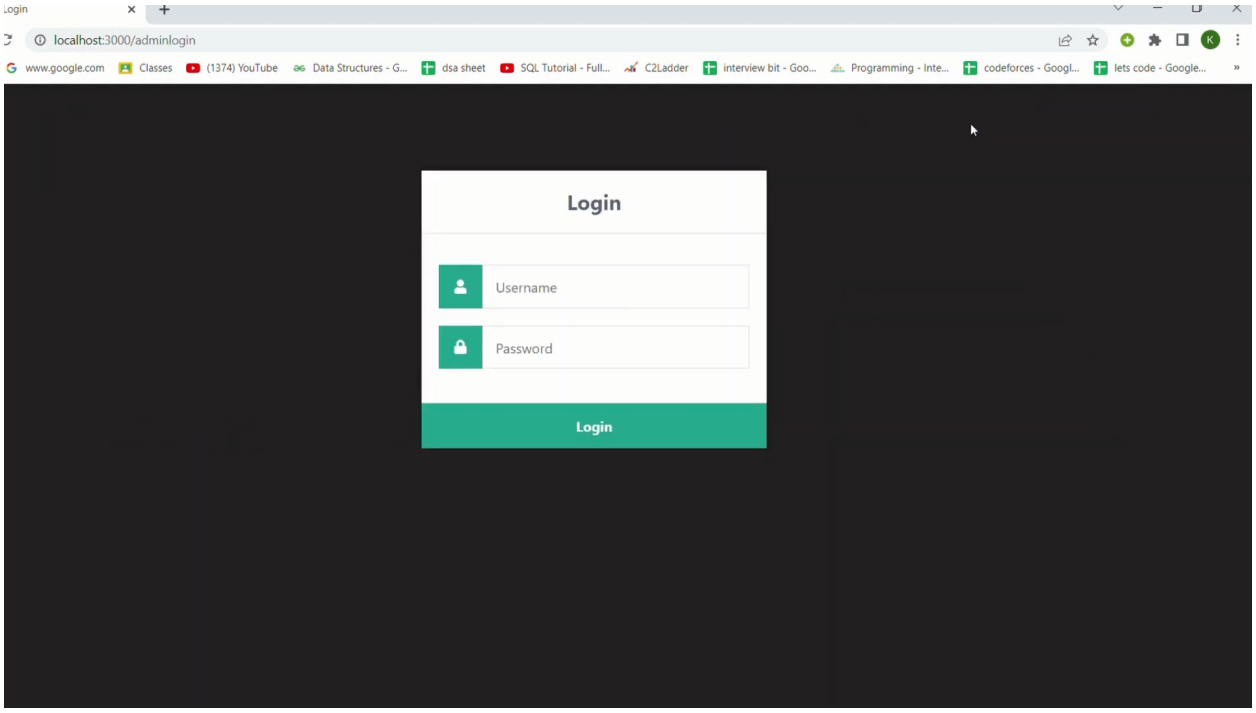


Screenshots

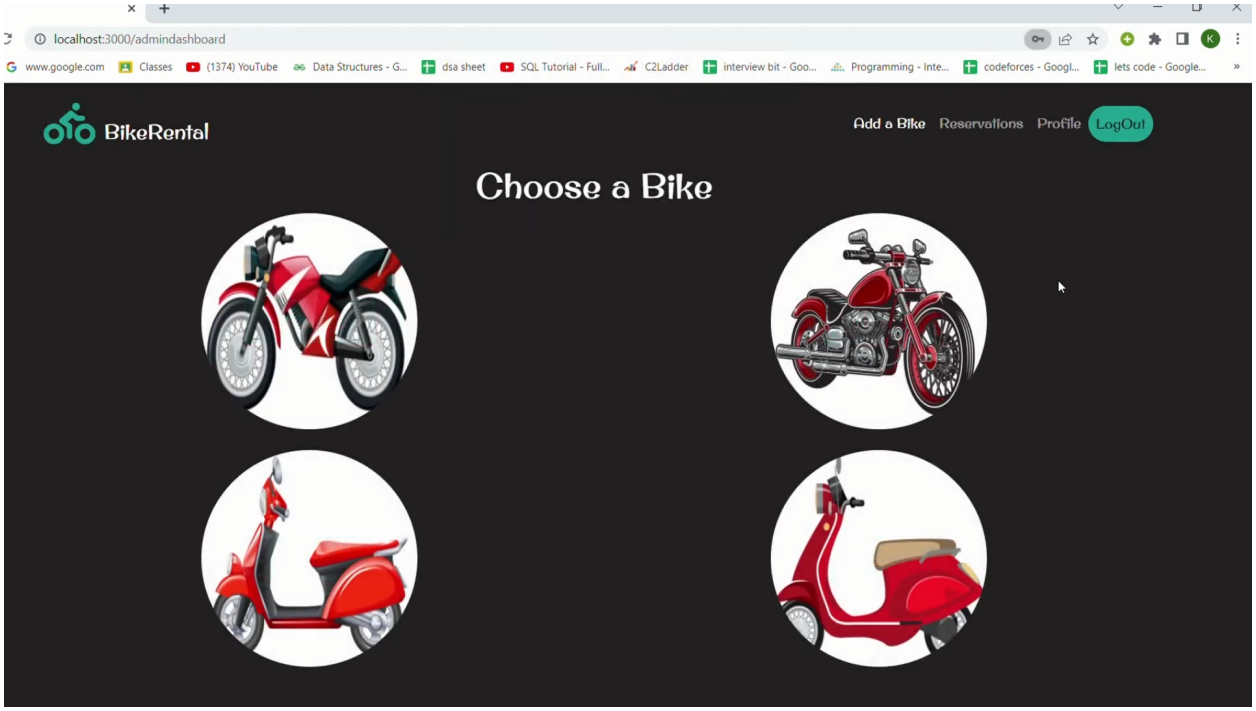
1. Home Page



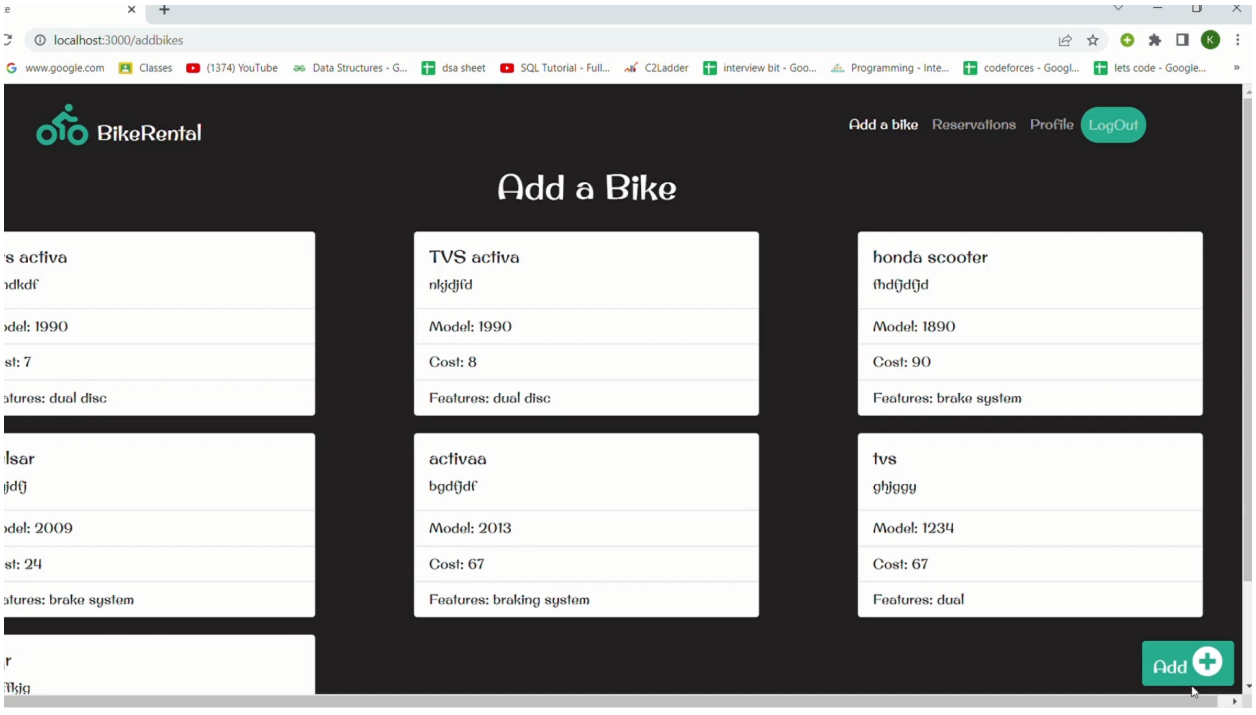
2. Admin Login



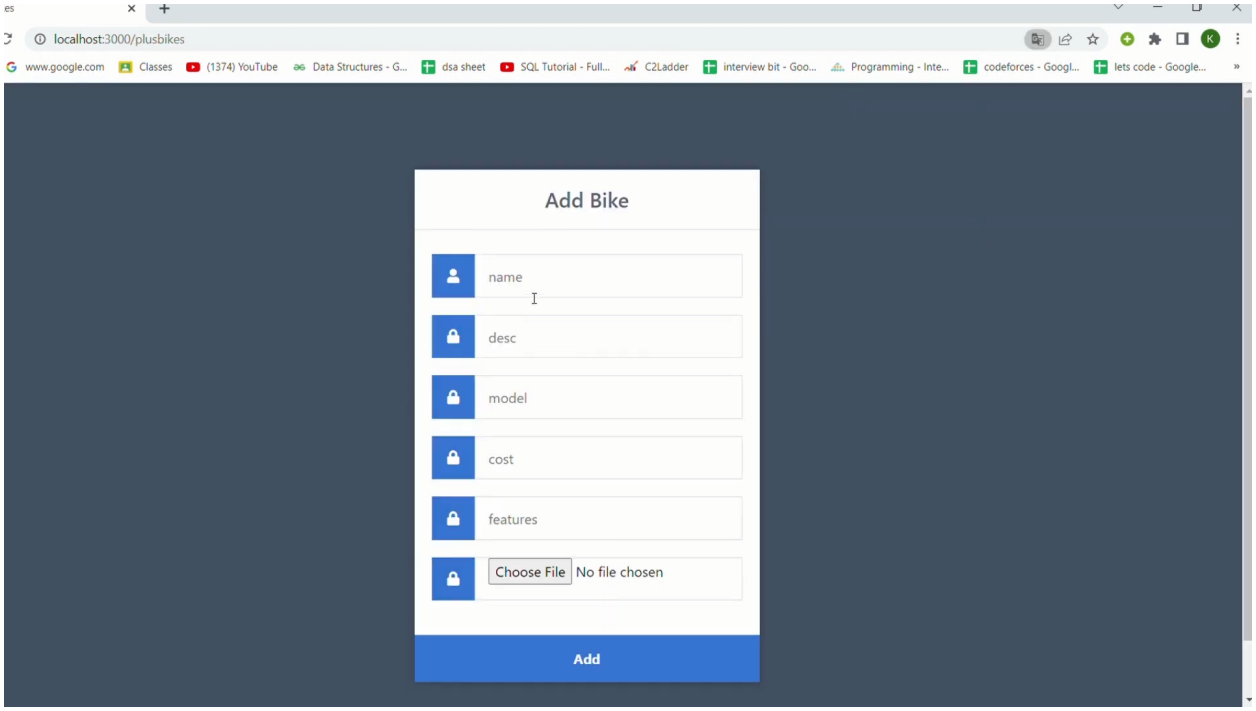
3. Admin Dashboard

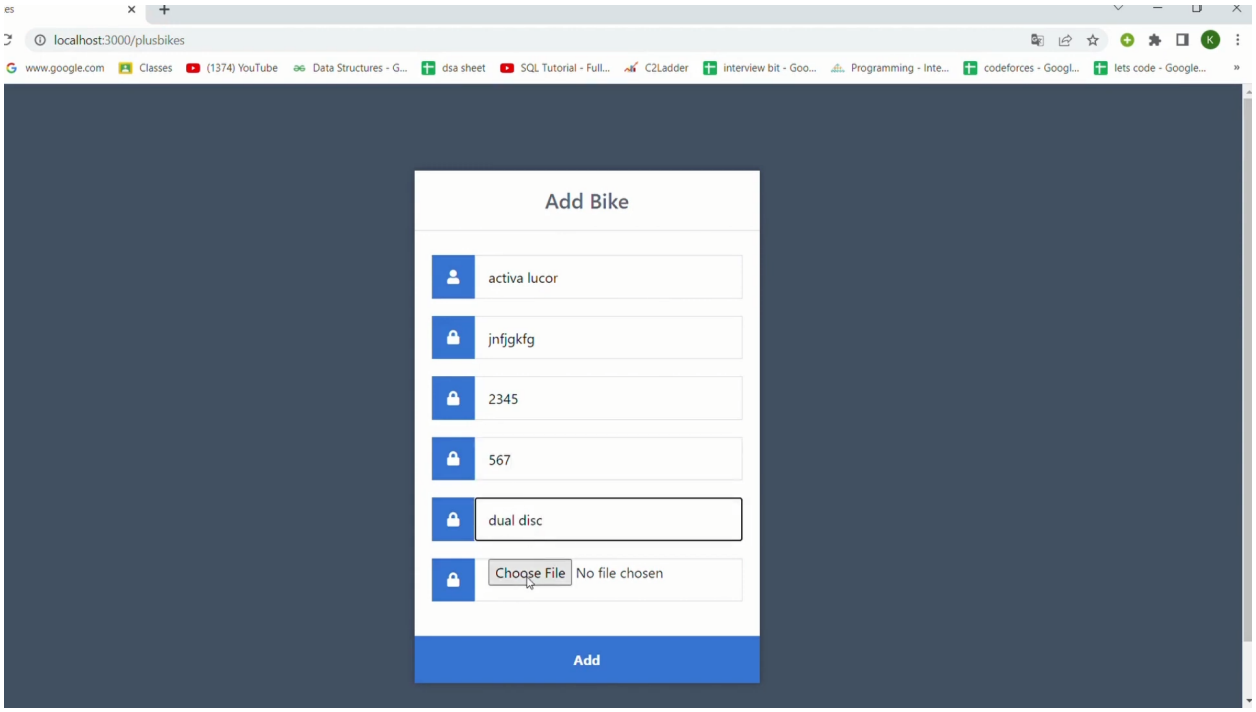


4. Available Bikes

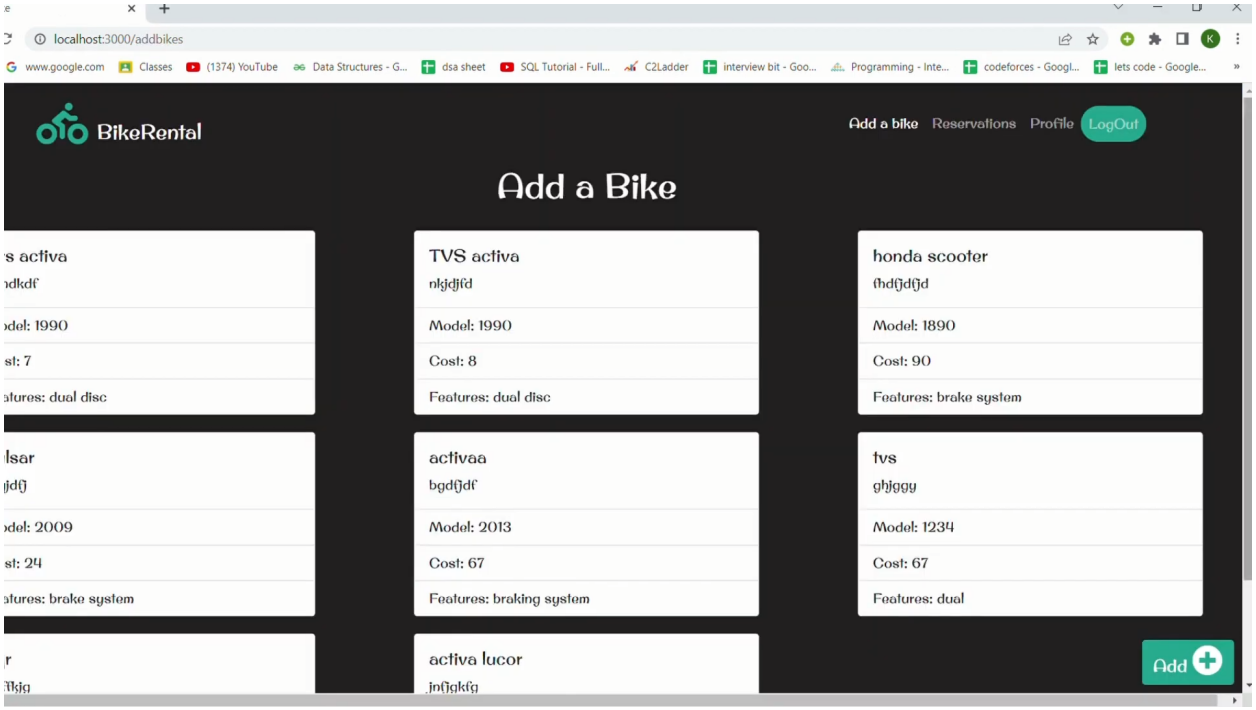


5. Adding a Bike

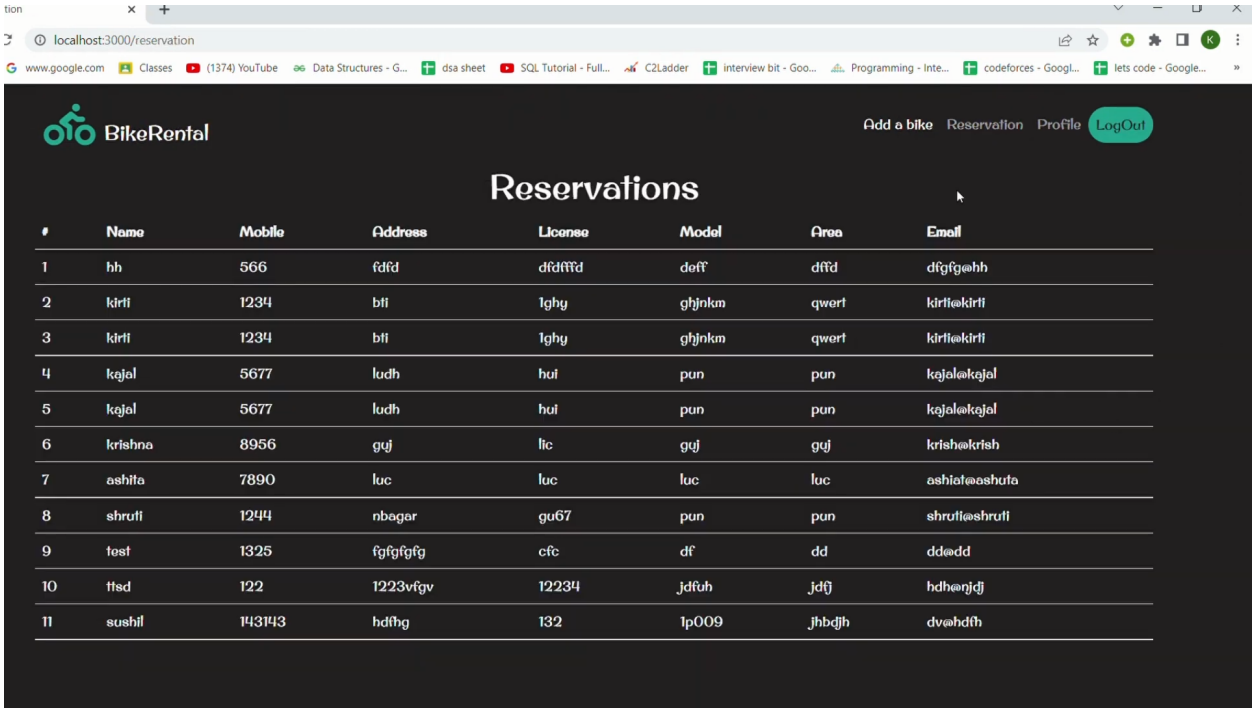




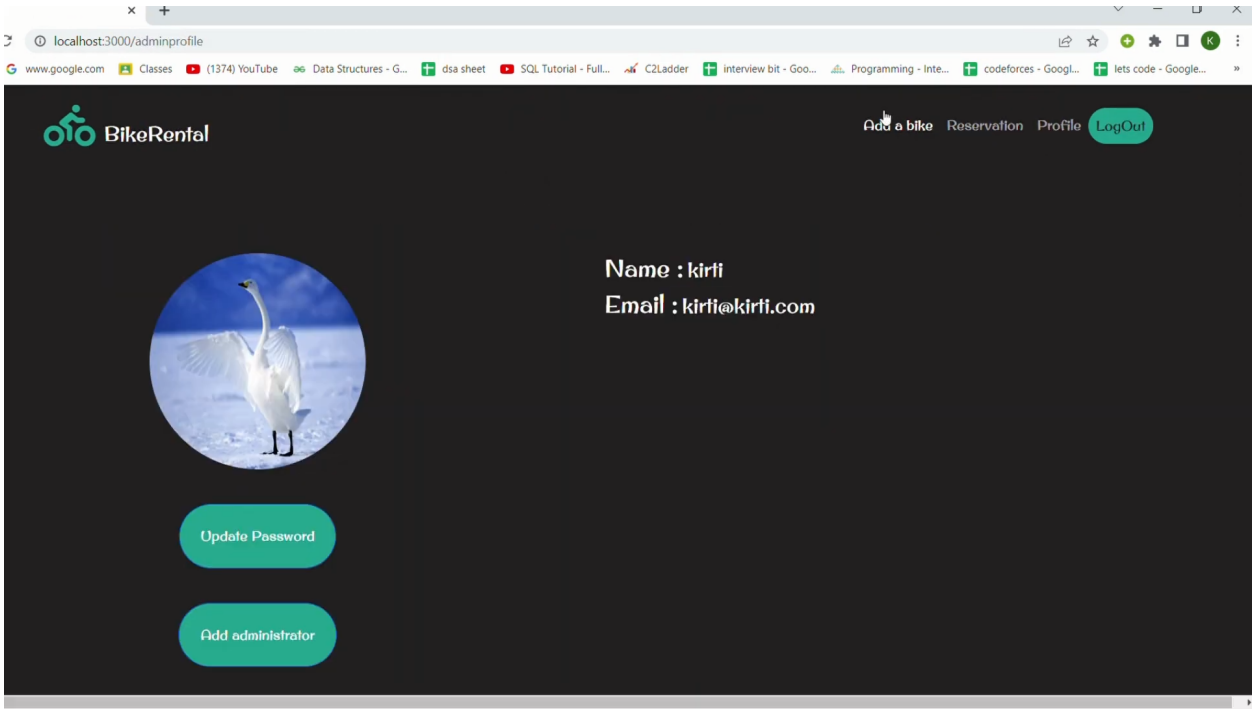
6. Added Bike



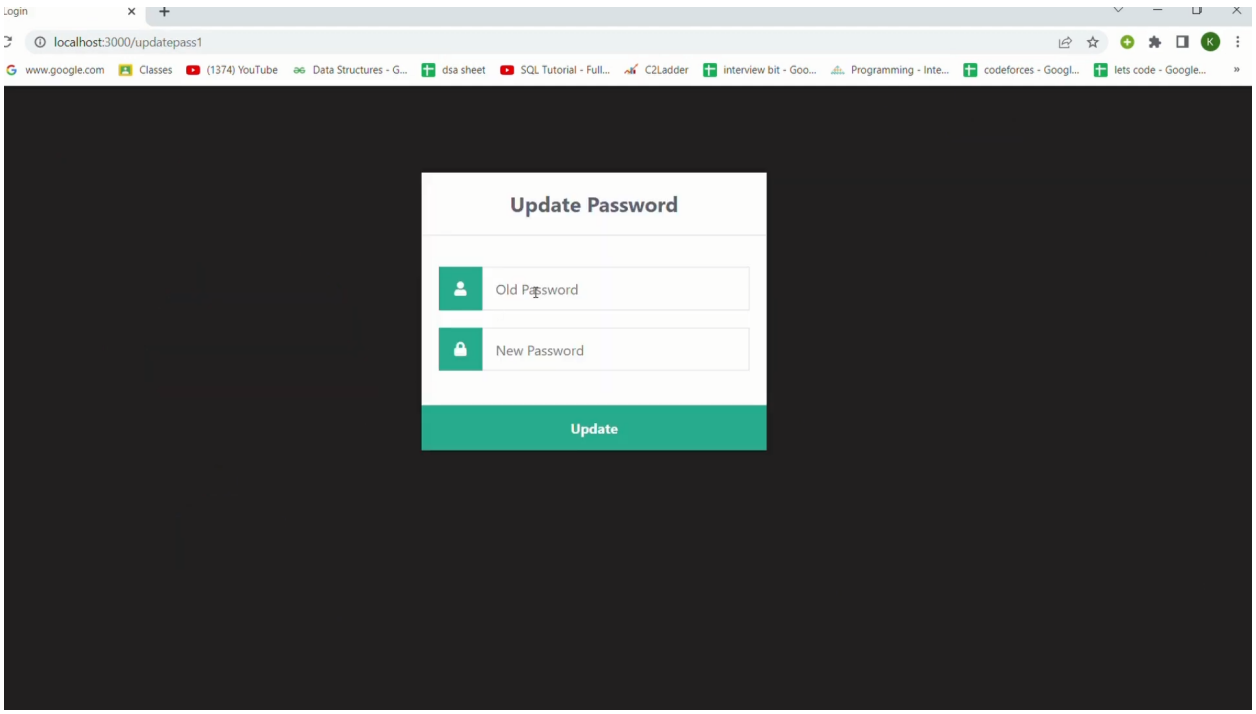
7. List of all bookings available to admin



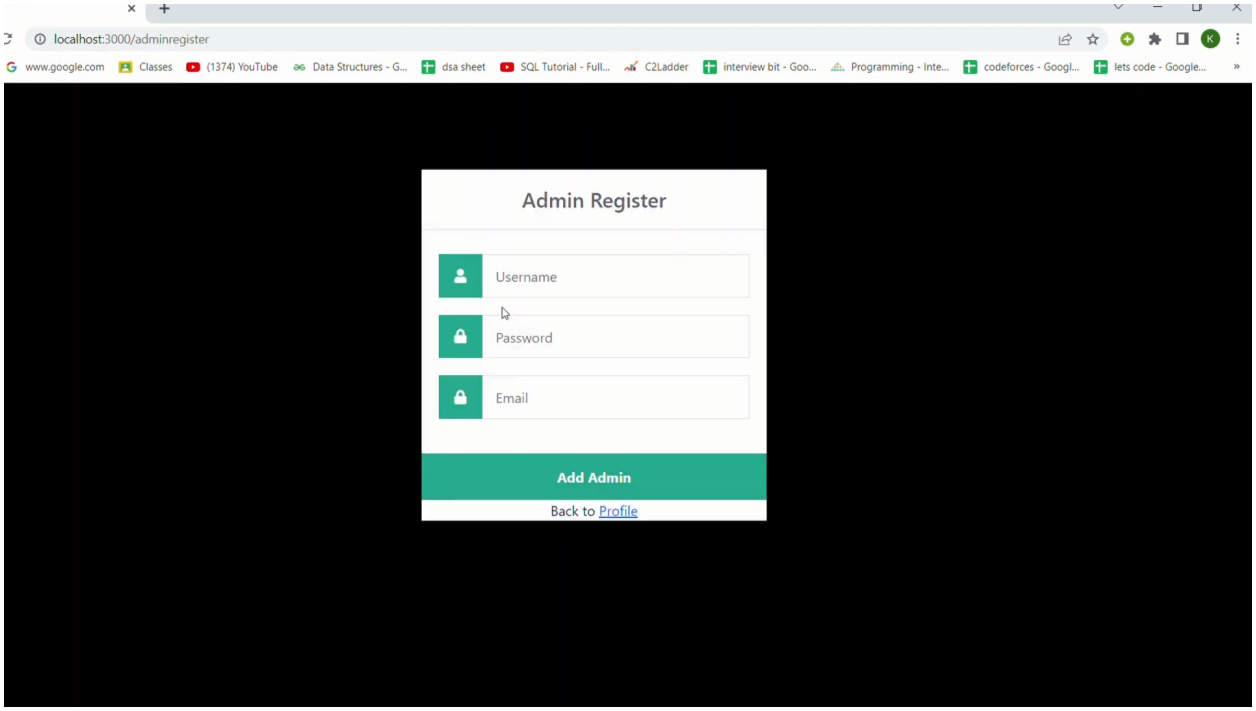
8. Admin Profile



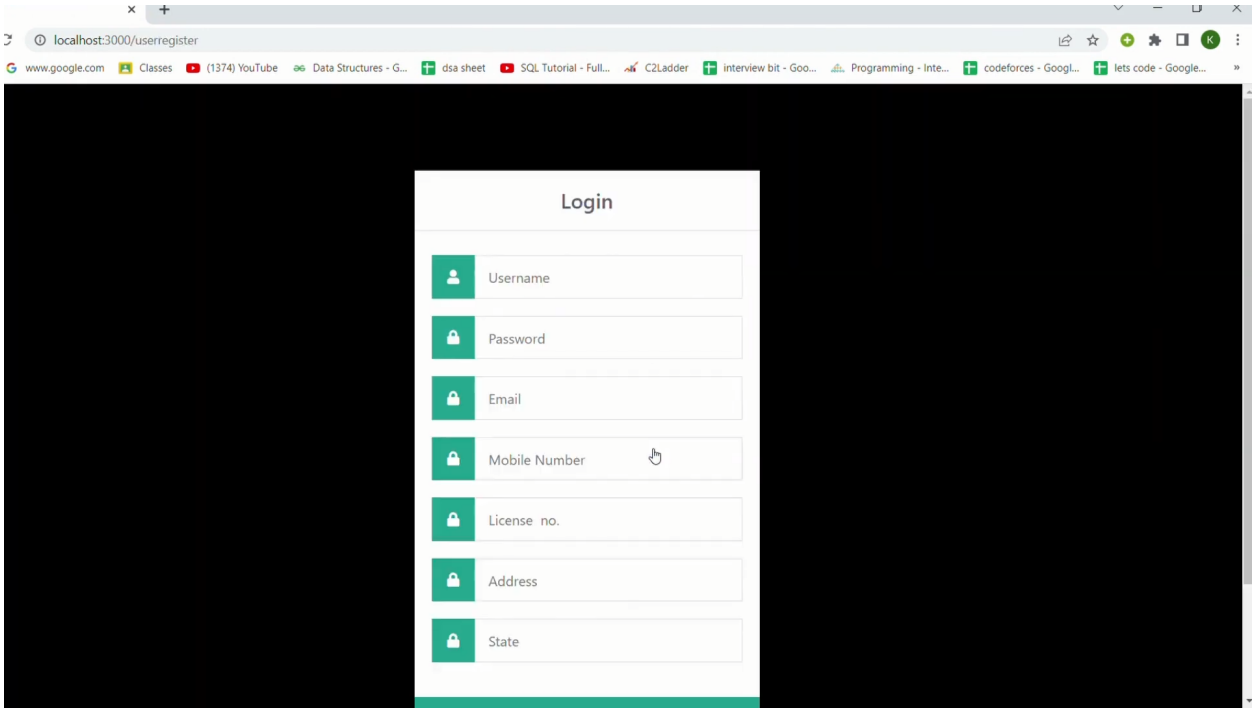
9. Password Update for Admin



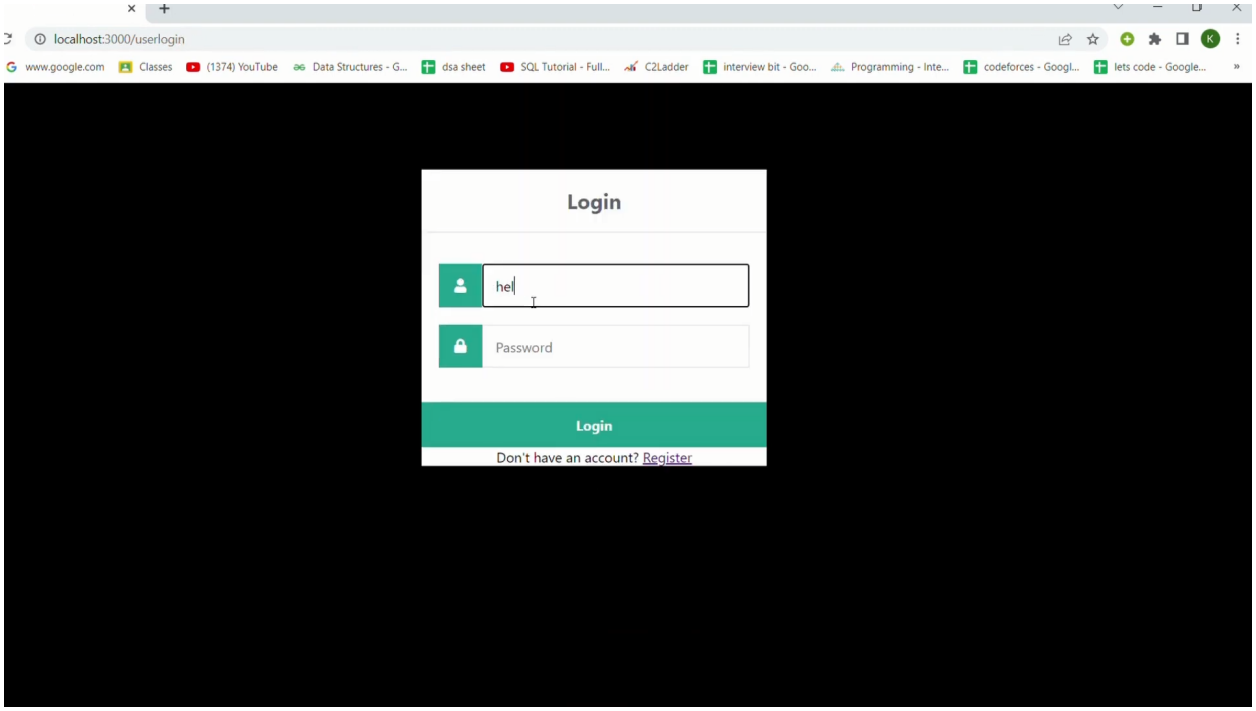
10. Admin Registration



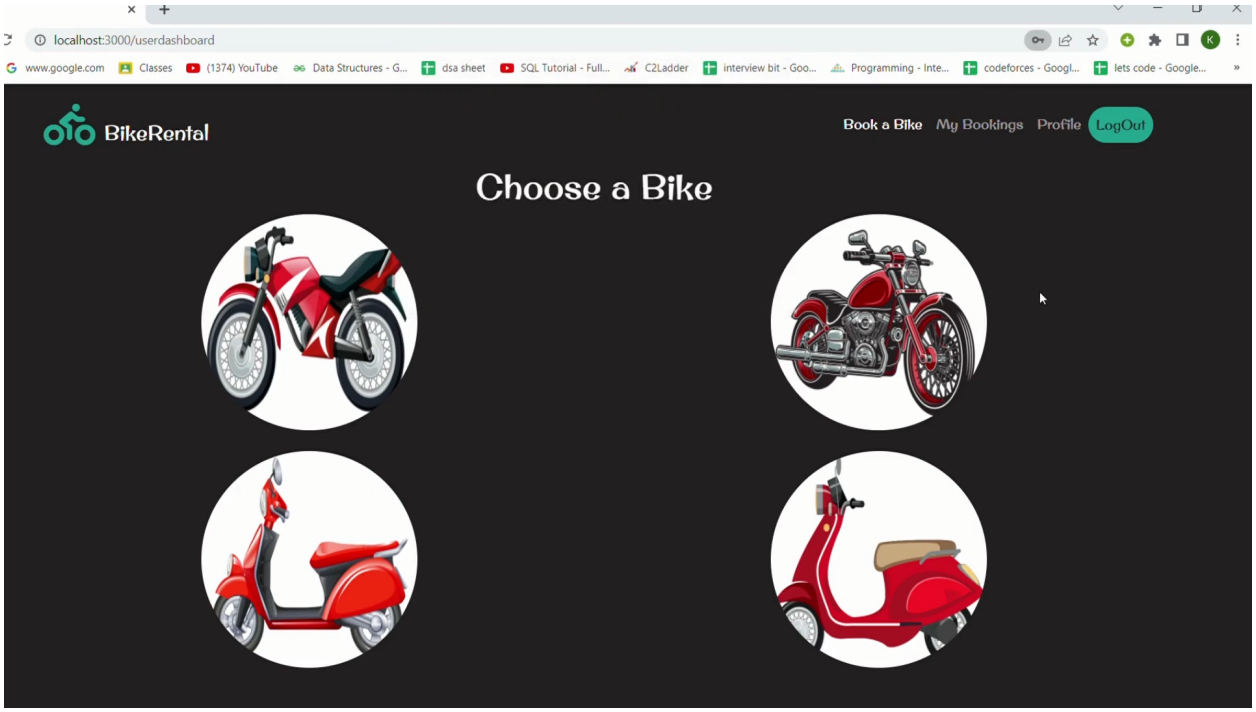
11. User Registration



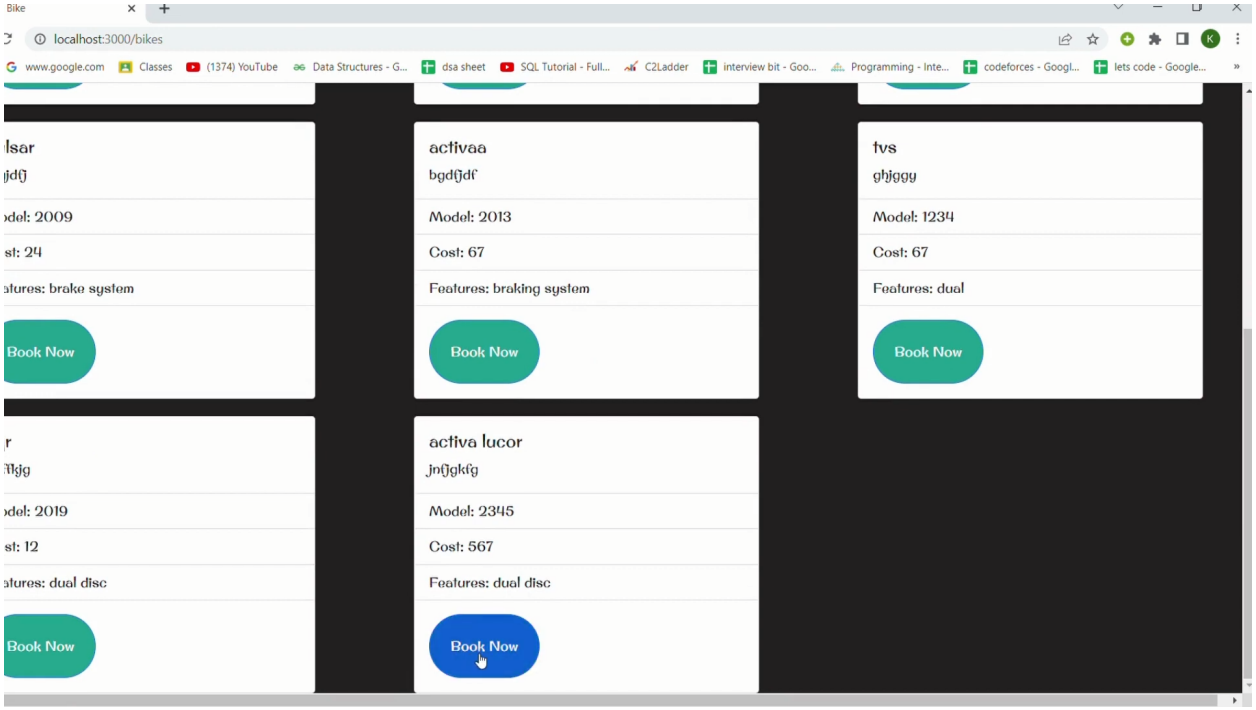
12. User Login



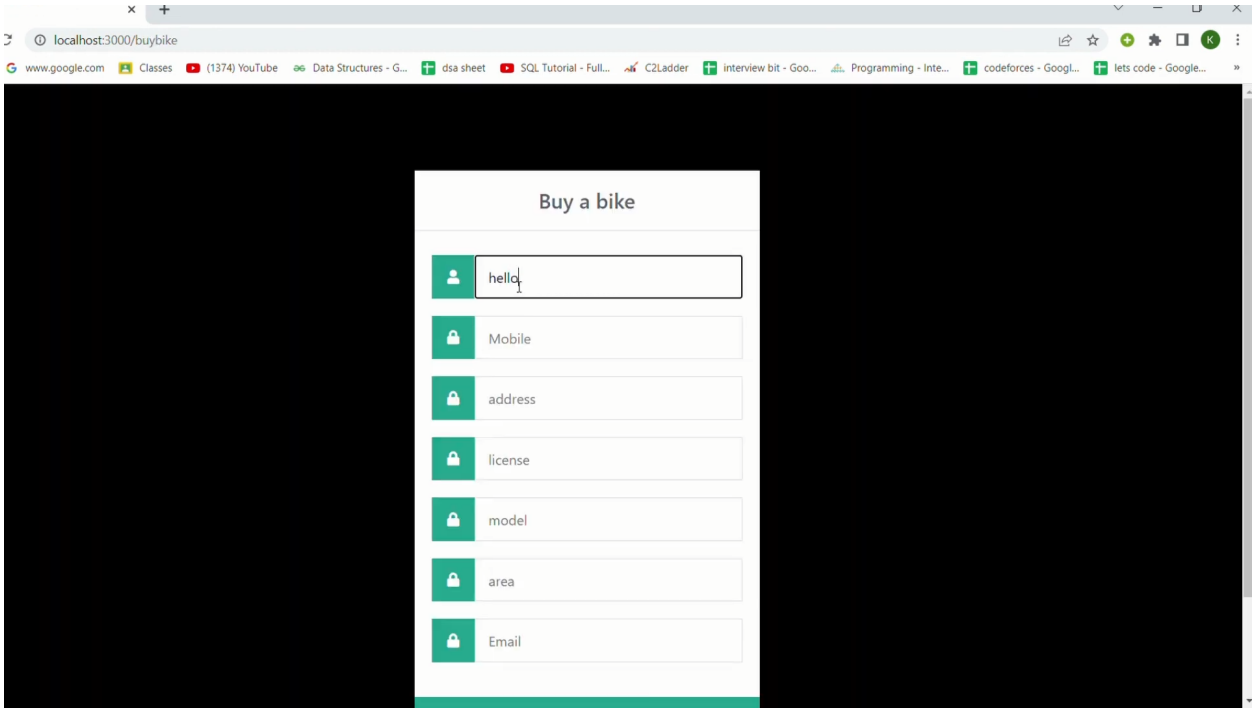
13. User Dashboard



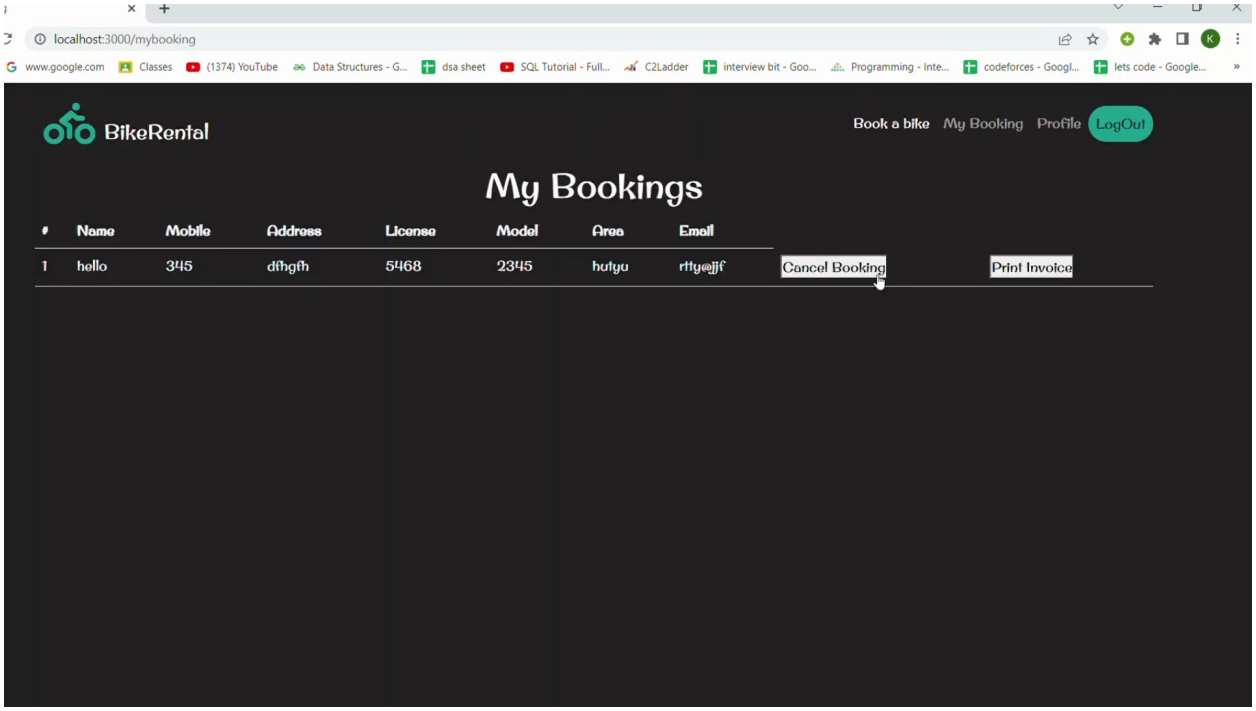
14. List of bikes available for rent



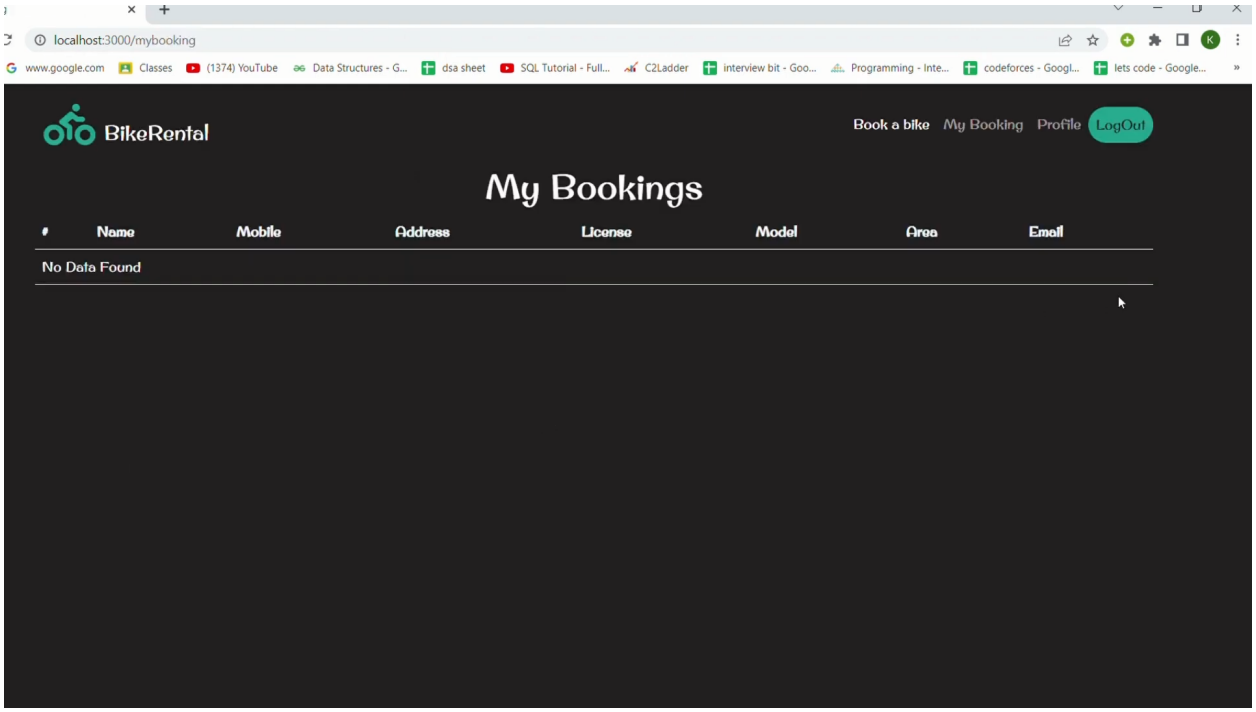
15. User buying a bike



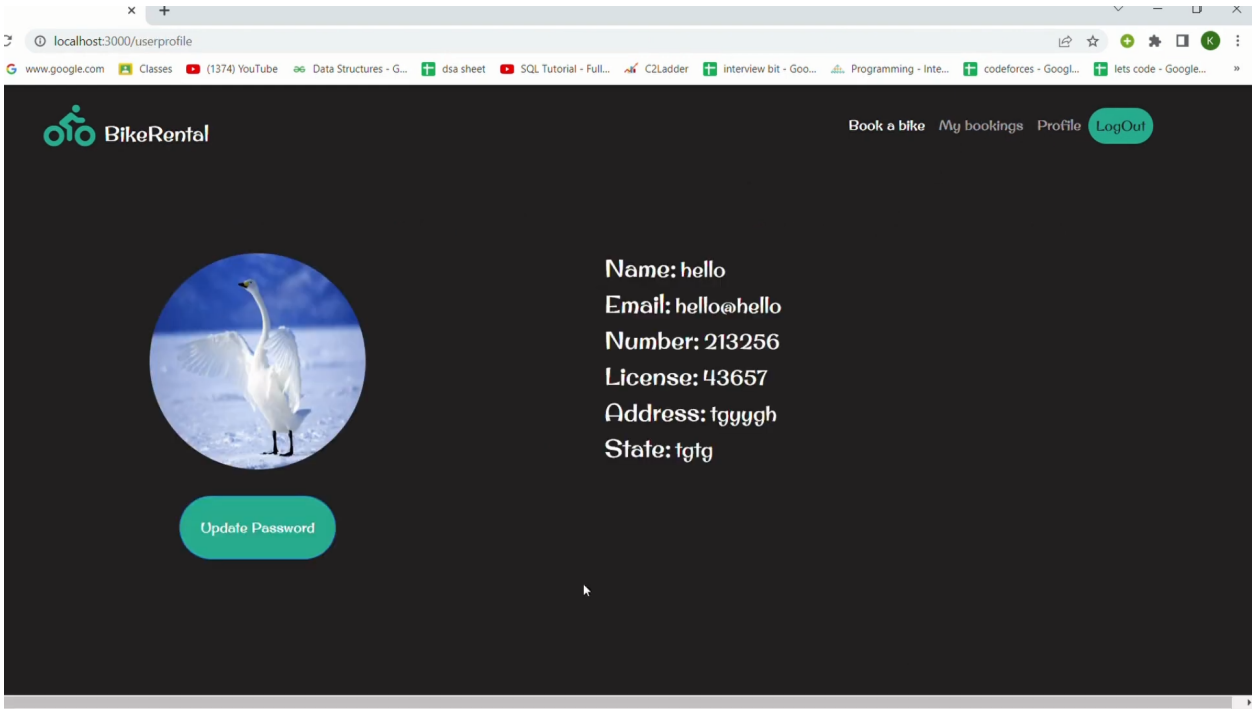
16. Bike bookings of a user



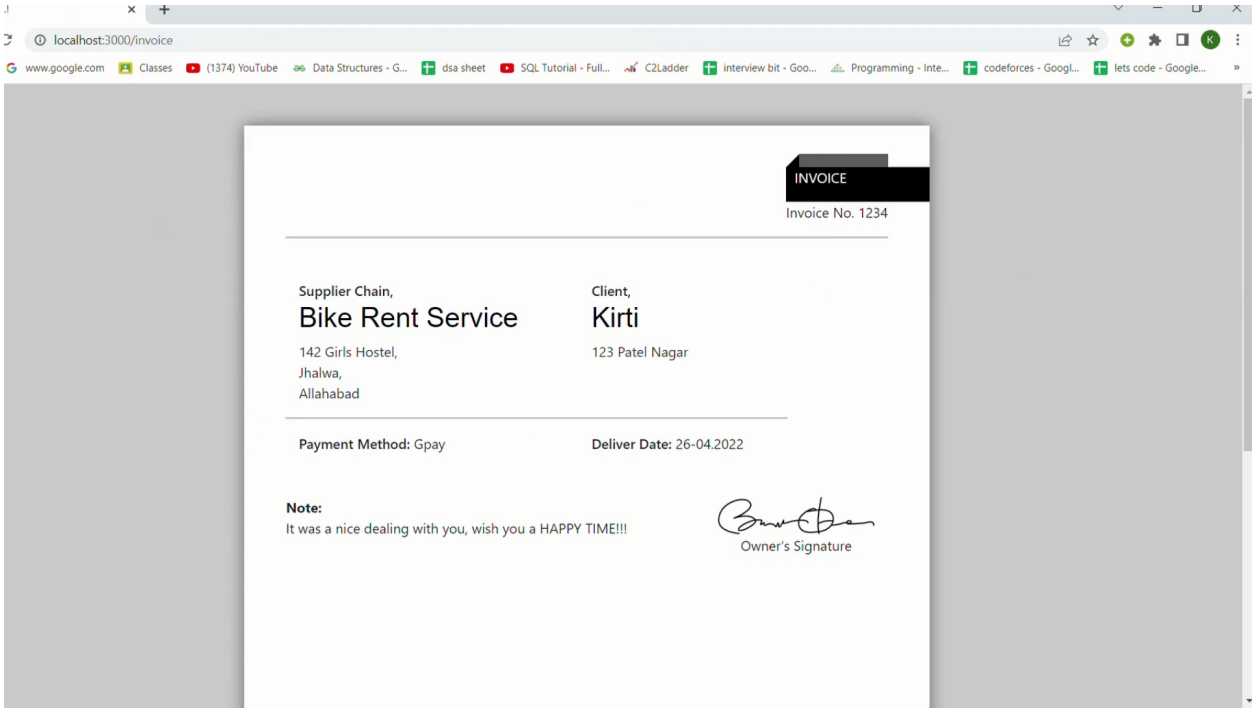
17. Cancellation of a booking



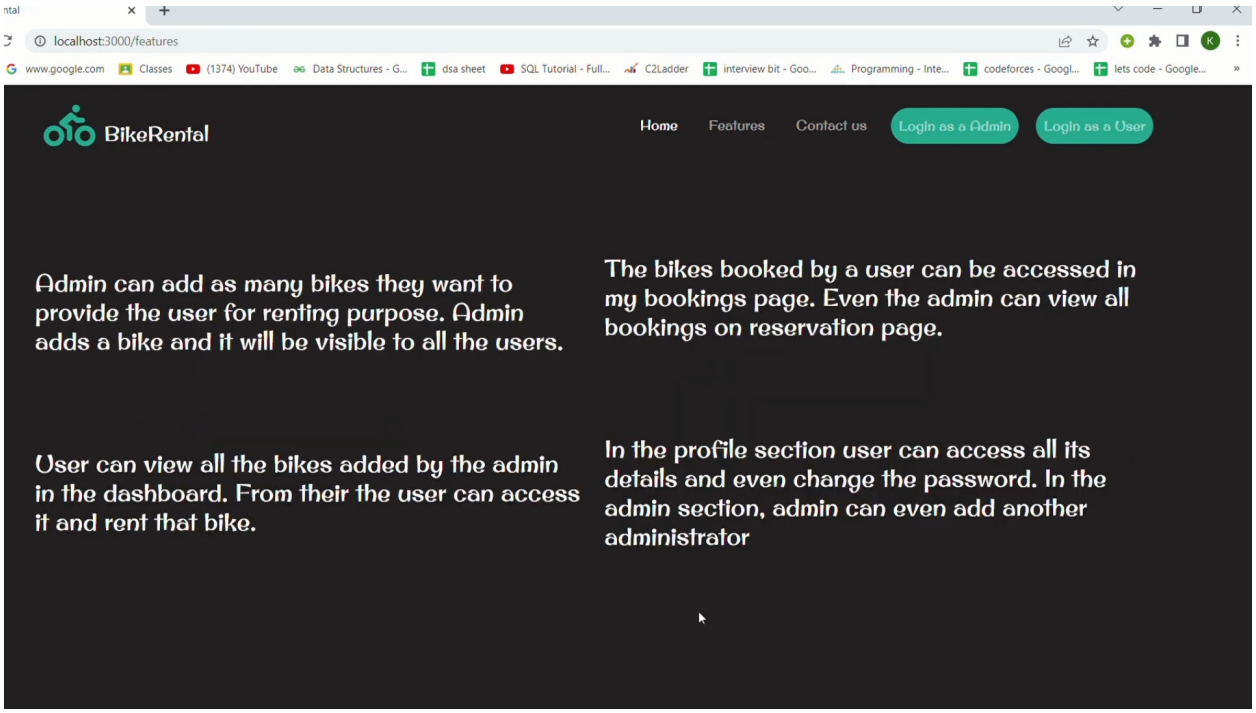
18. User profile page



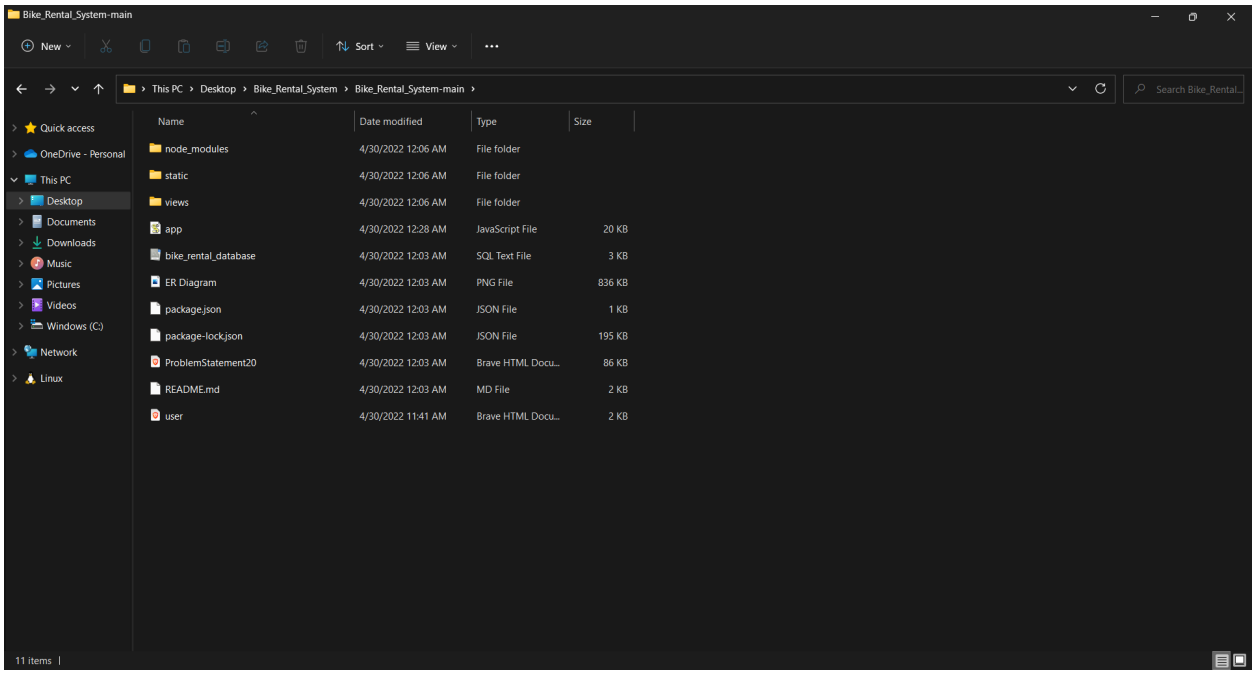
19. Invoice of the booking



20. Features page



21. Pdf of bill by the name user.pdf will be downloaded in the same directory



Scope

- It functions as a system for storing, processing and handling customer information and the information of the most valuable assets which are the bikes for the company.
- The web administrator is the one who has full control and authority to control the privilege and update the proposed system from time-to-time. In addition, they are also required to ensure that the different types of user access via their permitting level.
- The admin are restricted to view, update or delete any information about the registered members' information such as car reservation history. They are allowed to view and update details of customers as well as of themselves.
- In addition, the registered members are allowed to update their personal information if necessary with their own valid username and password through the proposed system. Each of the updated personal information will be stored and kept in the database and can be categorized as confidential information.
- The reports are generated by the proposed system based on the management requirement such as print and view the invoice.
- The management is able to improve and provide the better services from time-to-time or make it as future references.
- The proposed system provides a two tier login concept. It means that different users will login with different permission or privilege. Therefore the users are able to access different functionality of the proposed system.