**Spring MVC Booking System**

The Spring MVC Booking System is a web application that allows users to book appointments, manage reservations, and view available time slots. It’s built using the Spring Framework and follows the Model-View-Controller (MVC) architecture.

**Table of Contents**

1. Introduction
2. Getting Started
3. Usage
4. Contributing
5. License

**PROJECT DESCRIPTION**

The GuraRide Reservation System is a modern web application designed to efficiently handle daily bike bookings in Kigali City. The primary objective is to simplify the reservation process and enhance customer service for both clients and GuraRide staff. Customers can conveniently reserve bikes online, monitor bookings, and save time by avoiding in-person visits. The system enables them to explore available bikes, make reservations, and check real-time availability. For GuraRide, the system offers an easy-to-use interface for managing inventory, tracking bike locations, and overseeing reservation activities.

Clients can easily choose and reserve bikes online, eliminating the need to visit the reservation center. The system's integration with GuraRide's inventory management ensures precise information on bike availability. Staff can log in, review, and confirm or reject reservations, providing GuraRide with centralized control. The system simplifies the check-in and check-out processes, making it straightforward for staff to confirm reservations and handle bike returns in real-time. In summary, the GuraRide Reservation System optimizes efficiency and customer experience, enabling GuraRide to make the most of resources and offer outstanding service in Kigali City.The primary objective is to simplify the reservation process and enhance customer service for both clients and GuraRide staff. Customers can conveniently reserve bikes online, monitor bookings, and save time by avoiding in-person visits. The system enables them to explore available bikes, make reservations, and check real-time availability. For GuraRide, the system offers an easy-to-use interface for managing inventory, tracking bike locations, and overseeing reservation activities.

Clients can easily choose and reserve bikes online, eliminating the need to visit the reservation center. The system's integration with GuraRide's inventory management ensures precise information on bike availability. Staff can log in, review, and confirm or reject reservations, providing GuraRide with centralized control. The system simplifies the check-in and check-out processes, making it straightforward for staff to confirm reservations and handle bike returns in real-time. In summary, the GuraRide Reservation System optimizes efficiency and customer experience, enabling GuraRide to make the most of resources and offer outstanding service in Kigali City.

**Introduction**

The Spring MVC Booking System simplifies appointment scheduling for businesses, clinics, or any service-based organization. Users can book appointments online, view their upcoming reservations, and receive notifications.

Key features:

* User registration and login
* Appointment booking
* Calendar view of available time slots

Email notifications

**II.PROBLEM STATEMENT**

GuraRide currently handles bike rentals using paper-based methods, leading to time-wasting and potential mistakes. They lack a system to monitor bike reservations or customer details, making it challenging to offer top-notch customer service. The GuraRide Booking System intends to address these issues by introducing a digital solution that simplifies the rental process and enables straightforward tracking of inventory and customer data.

**Prerequisites**

Make sure you have the following installed:

* Java Development Kit (JDK) 8 or later
* Maven (for building and managing dependencies)
* MySQL or another database (for storing appointment data)

**Installation**

1. Clone this repository:
2. git clone git@github.com:beckereg/SpringMVC-BookingSystem.git
3. Build the project using Maven:
4. cd SpringMVC-BookingSystem
5. mvn clean install
6. Configure your database connection in application.properties.
7. Run the application:
8. mvn spring-boot:run

**III.** **FUNCTIONAL REQUIREMENT OF THE SYSTEM**

All users should have the ability to log in and create an account.

Customers (renters) should log in, explore system features, view available bikes, reserve bikes for specific times, and track their reservations.

Customers select a bike, make a reservation, and monitor its progress.

GuraRide staff (workers) can browse customer information, confirm or reject reservations, and log in to view reservations, manage bike rentals, and track inventory.

GuraRide staff should be able to confirm the return of rented bikes.

When a customer arrives to pick up a bike, staff can use the system to confirm the reservation and check out the bike.

GuraRide staff can track bike returns using the system.

The admin has full control, able to edit/delete user accounts, register new bikes, edit/delete bike info, and generate reports for managerial functions.

The system generates reports for admin, staff, and customers to help GuraRide track rental revenue, inventory levels, and customer information.

Users can log out at their convenience.

The system implements authentication and authorization through Spring Security for user access to privileged resources.

**IV.** **NON FUNCTIONAL REQUIREMENT OF THE SYSTEM**

Ensure user data privacy and security, allowing only authorized and authenticated individuals with the right privileges to access dedicated resources.

Prioritize system performance and availability.

Design a user-friendly interface that is intuitive, visually appealing, and easy to navigate.

Guarantee compatibility with different devices and browsers, ensuring correct functionality across various platforms.

**V.DATABASE SCHEM**

The GuraRide Bike Rental Management System consists of 3 tables described below.

* ‘Users’

Contains all the information of the user of the application:

\*Names

\*Email

\*Date of Birth

\*Pasword and it confrimations

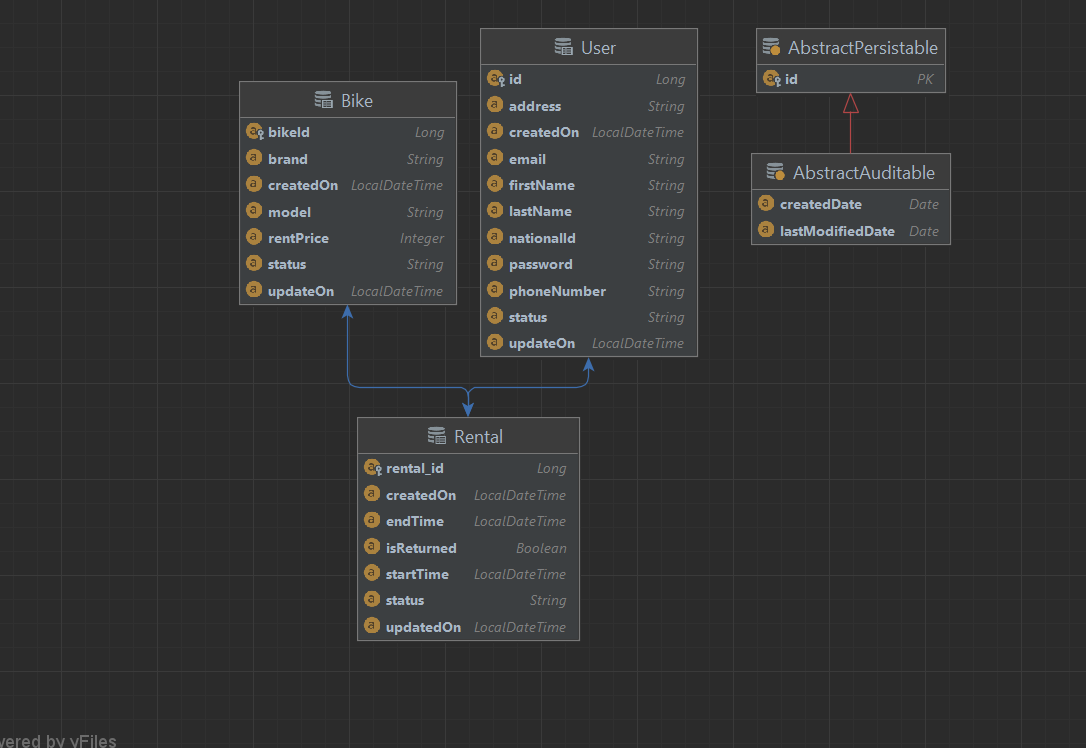
* ‘Bikes’

Also contains the information of the Bikes. As seen in the picture below we shall be recording the bike brand, bike model and the status, + the id, created and the update\_info.

* ‘Rentals’

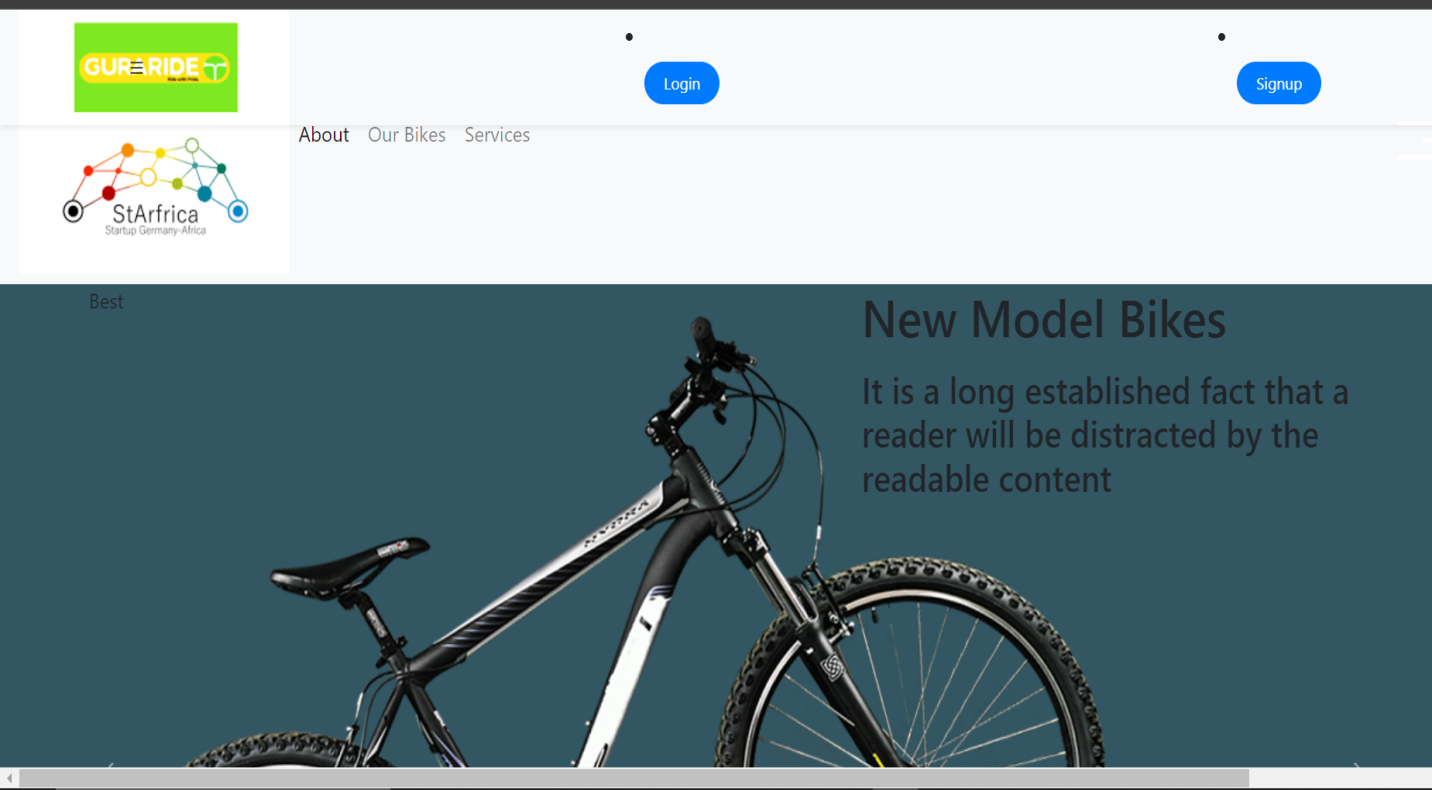
Finally the rentals table will be used to record the daily transaction of the rentals operation. It contains the ids foreign keys from the users and bikes table, where at each transaction we shall be saving the user\_id to indicate the person who rented or reserved the bike and the bike id to indicate the bikes which s/he rents or reserve.

In the below ERD the rentals bike\_id and the user\_id are mapped from the bikes and users tables respectively.

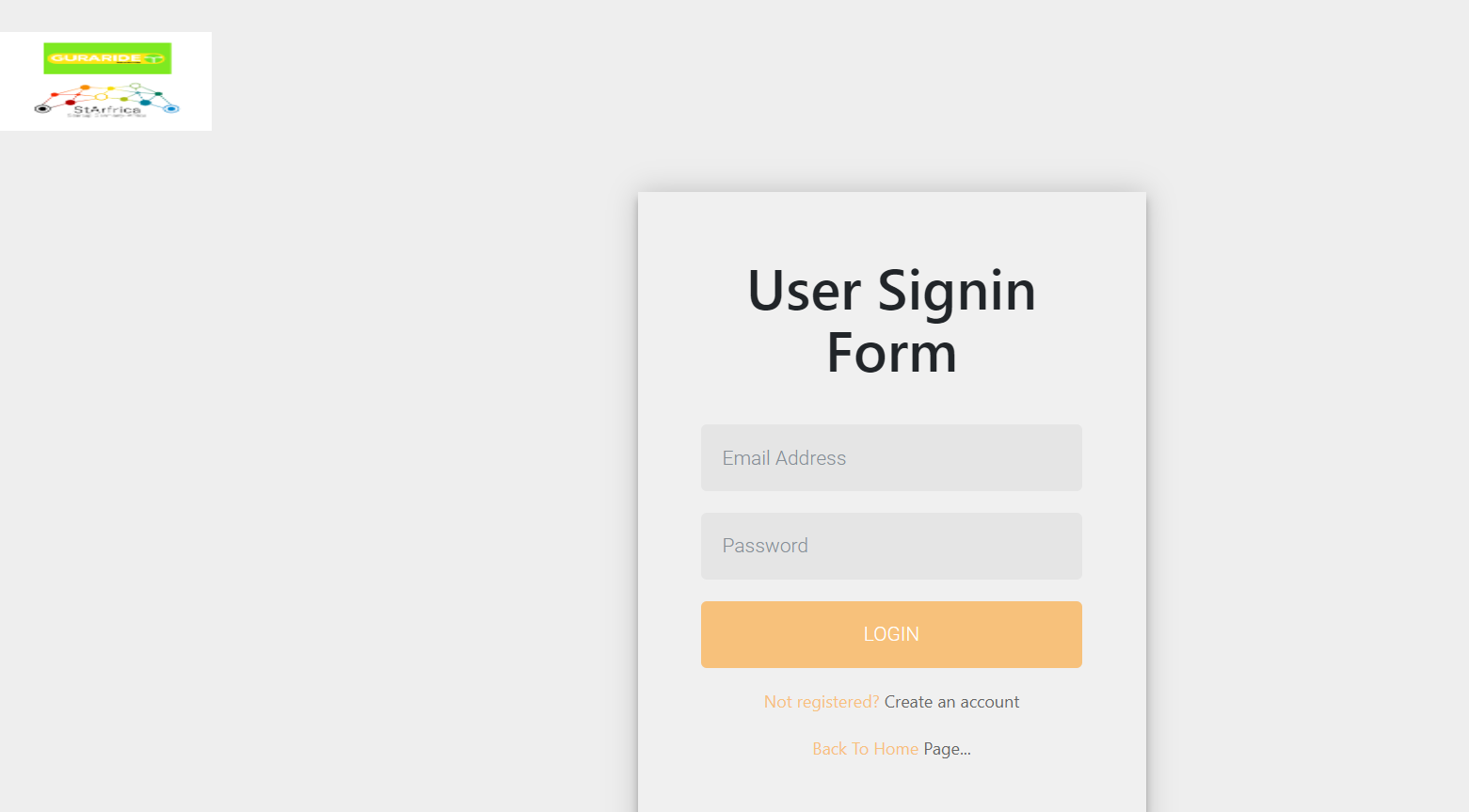
****

**VI. USER DOCUMENTATION**

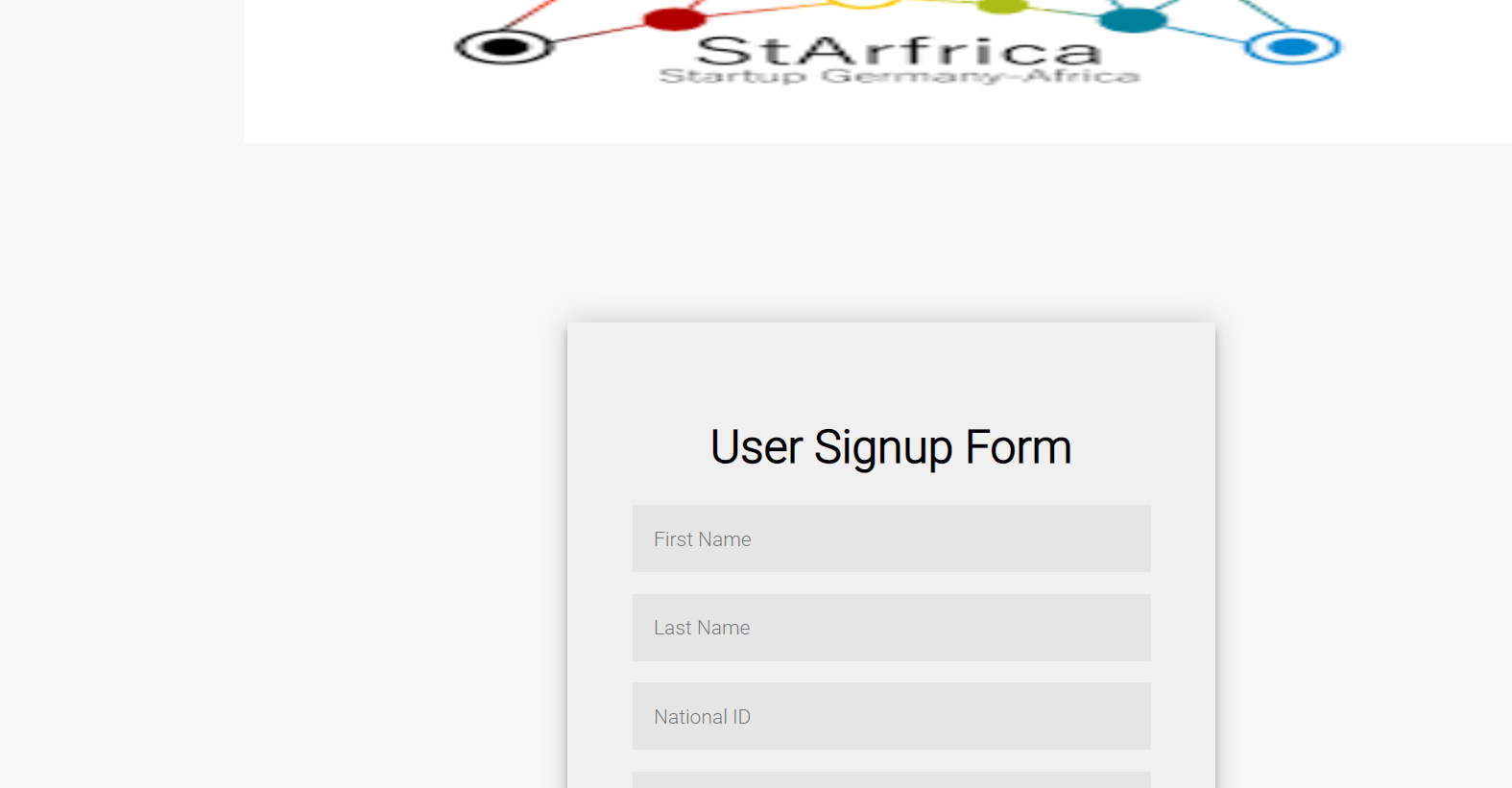
#.This is the ‘/’ path accessible when you run the project for the first time.



Login page for the users to enter in the system

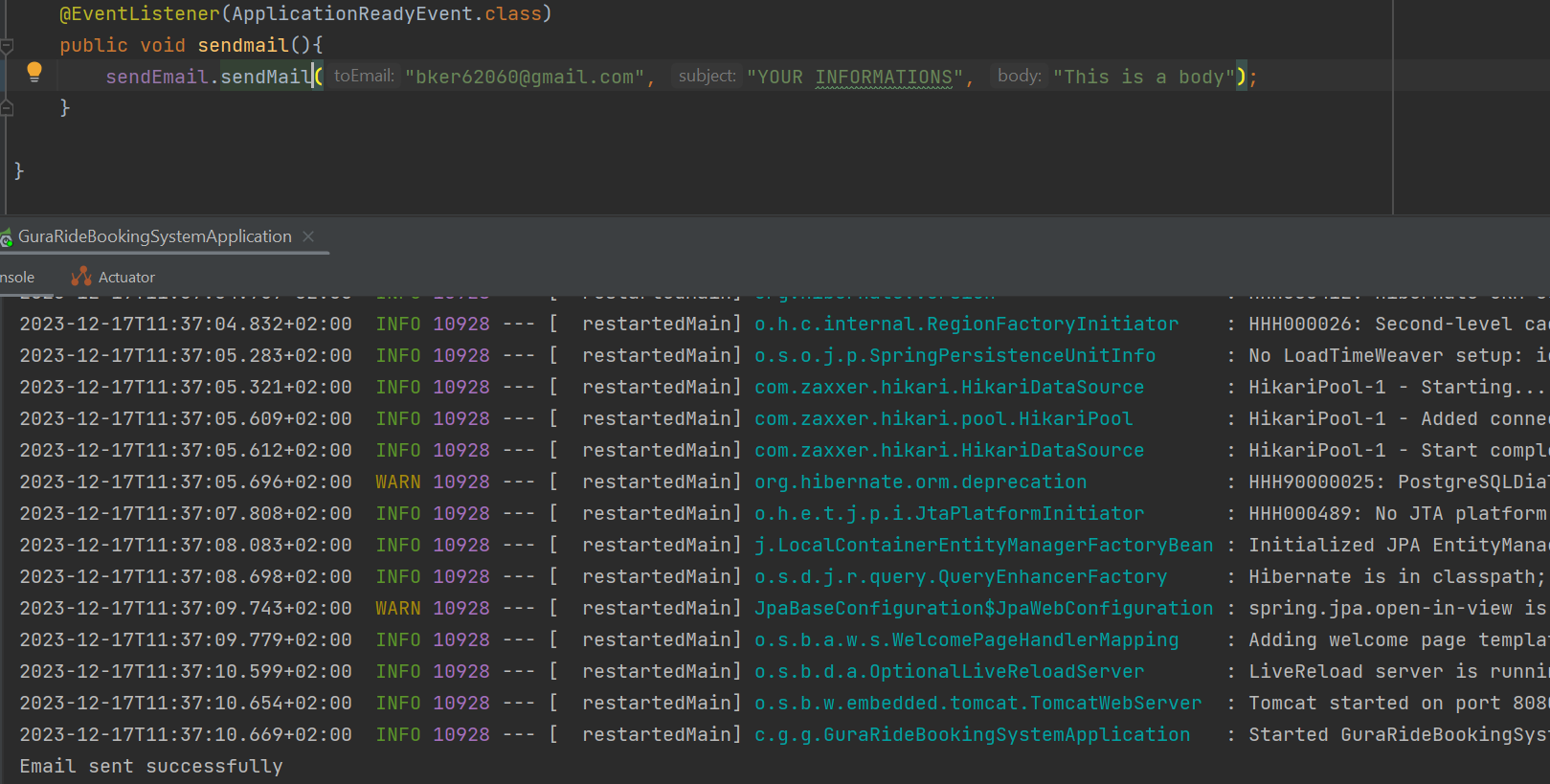


Signup page for the first time users to use throughout the system

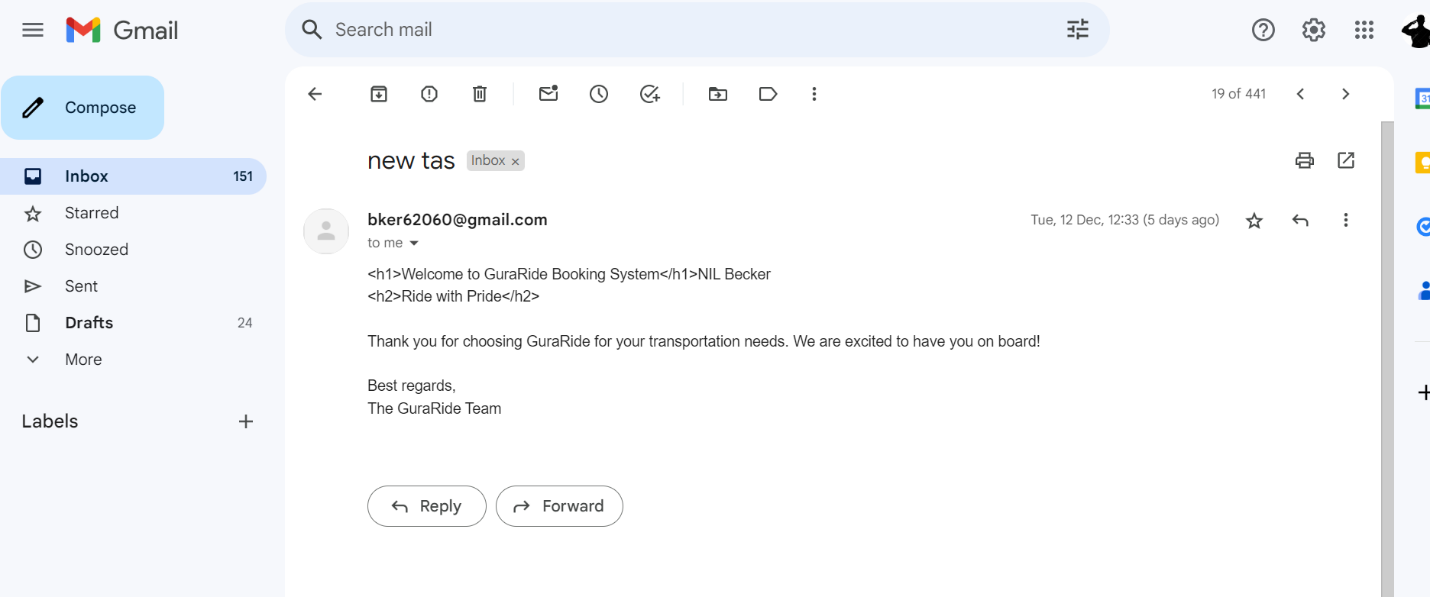


The application uses 3 personnel; which include the admin, worker and renter. User should first provide their credentials [email and password ] for authentication so that they can access the privileged resources and each user will be redirected to a respective page according to his/her role in the application

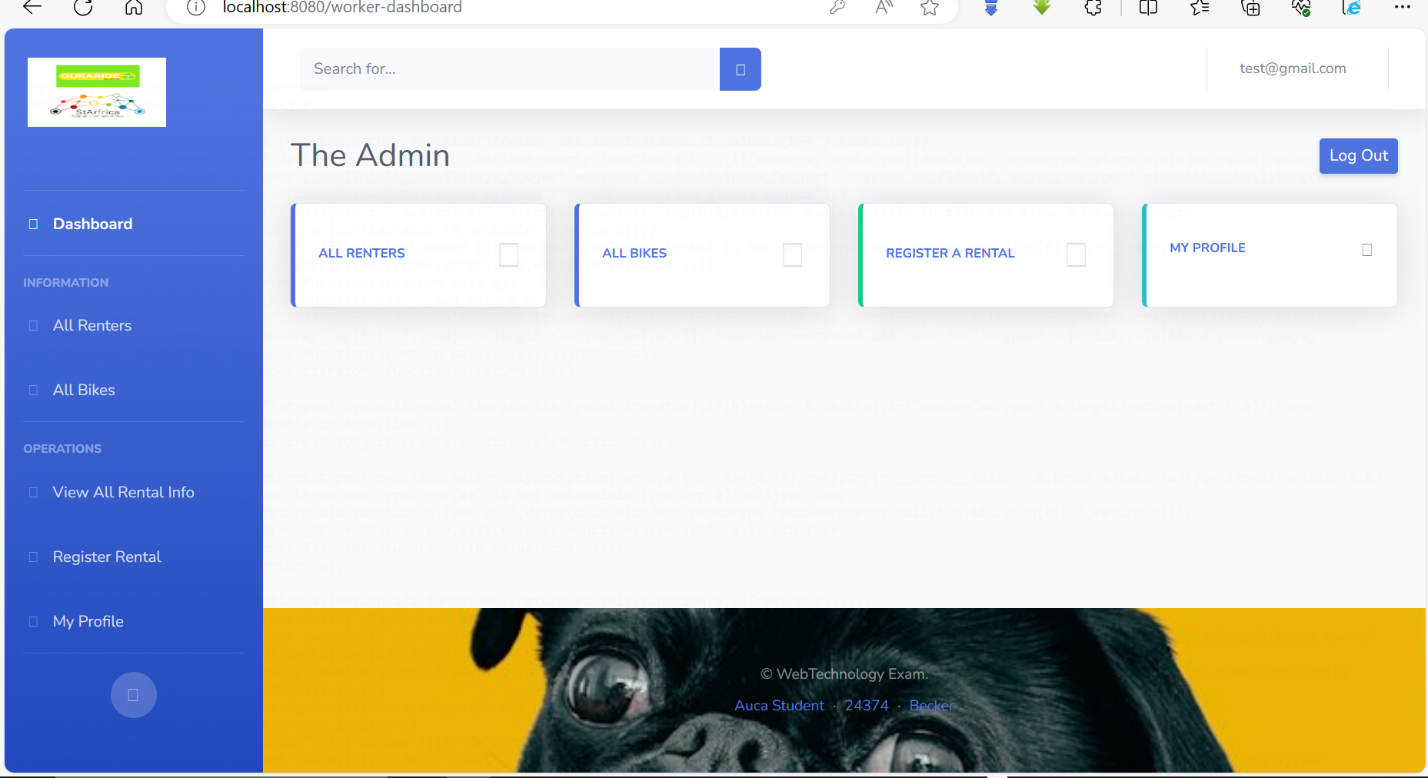
After any one who make a Signup either user or a new Admin will get an Email (from [bker62060@gmail.com](mailto:bker62060@gmail.com)) that contain the Information’s for the client



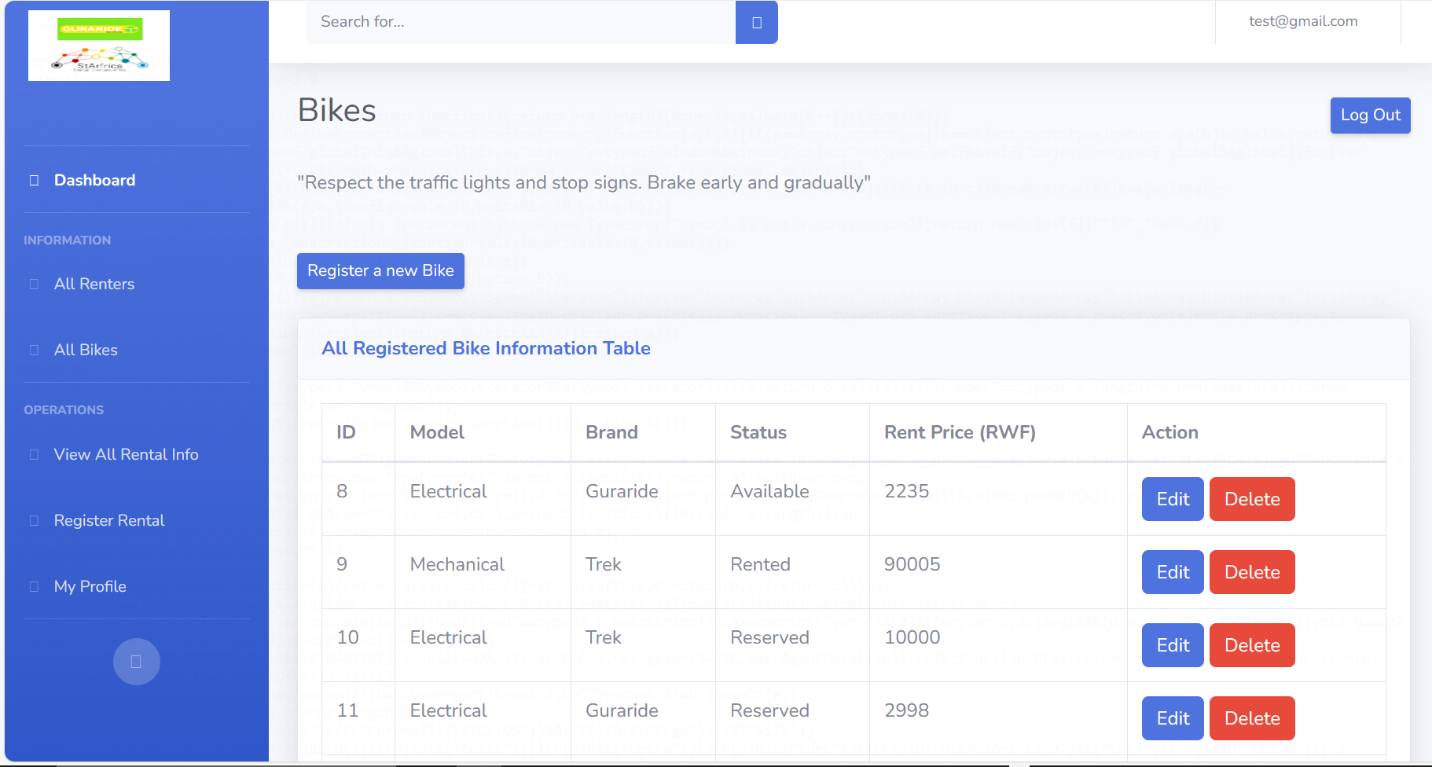
So here there is the email sent:



The Admin Dashboard View, where he/she has access to view/edit/delete all the user information, and bikes information:

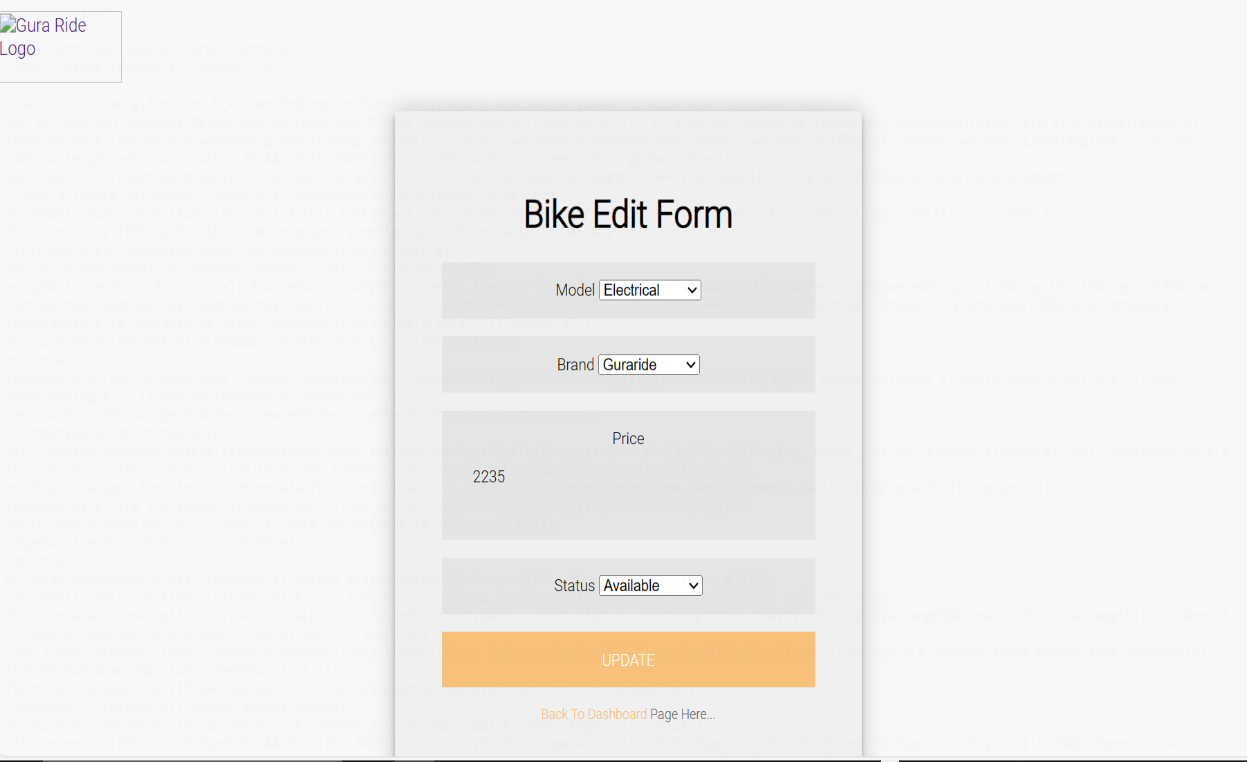


Admin User can also view the bike information and also register a new bike to the system.

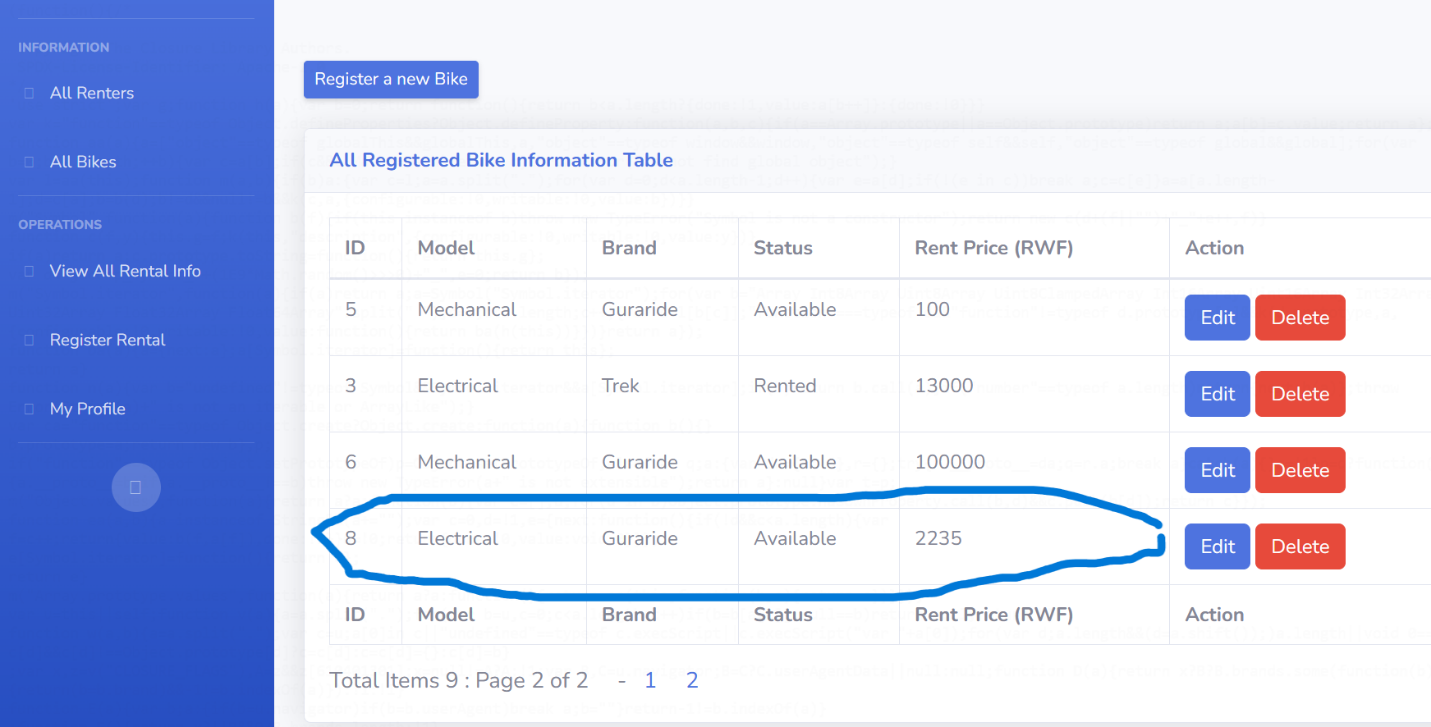


Here the Admin is the one to edit and to delete any Registered bike information

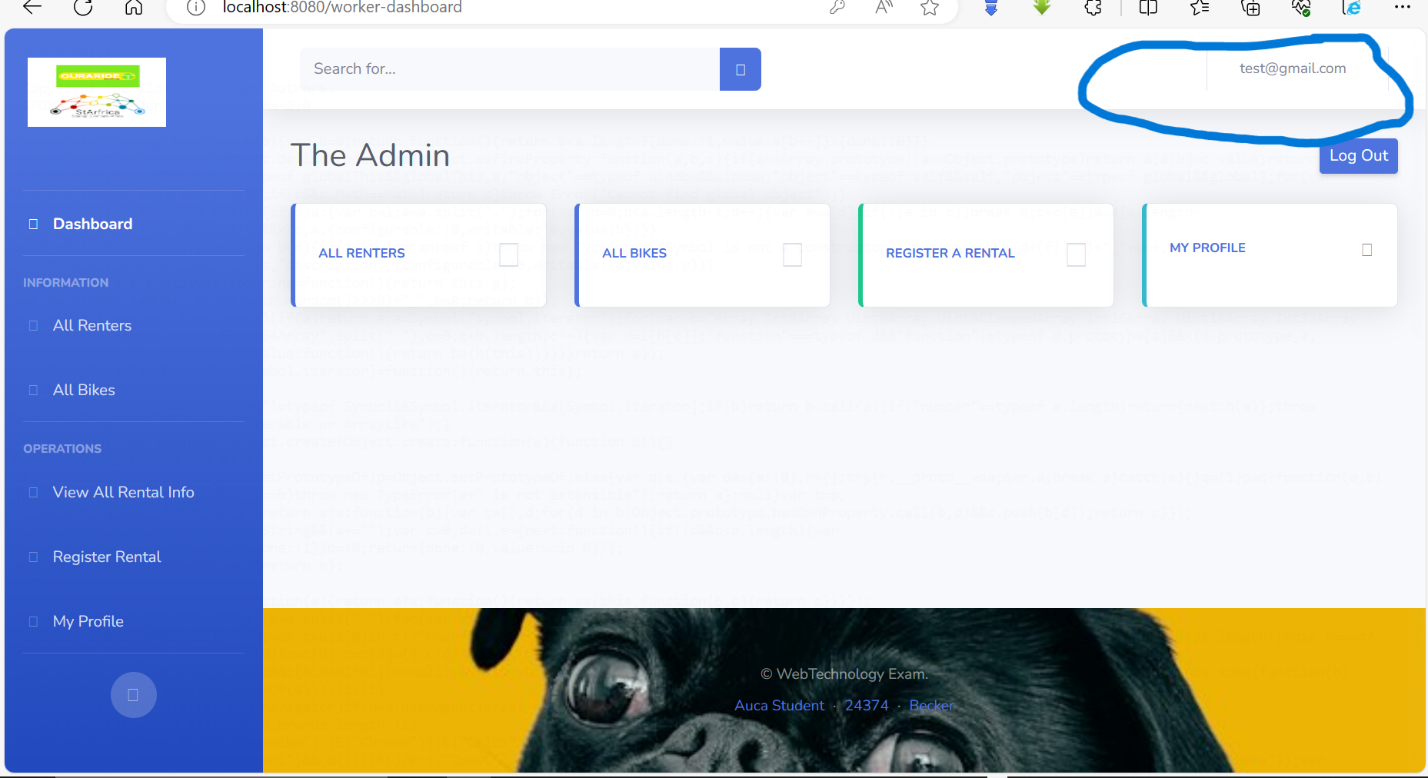
As here the admin is editing



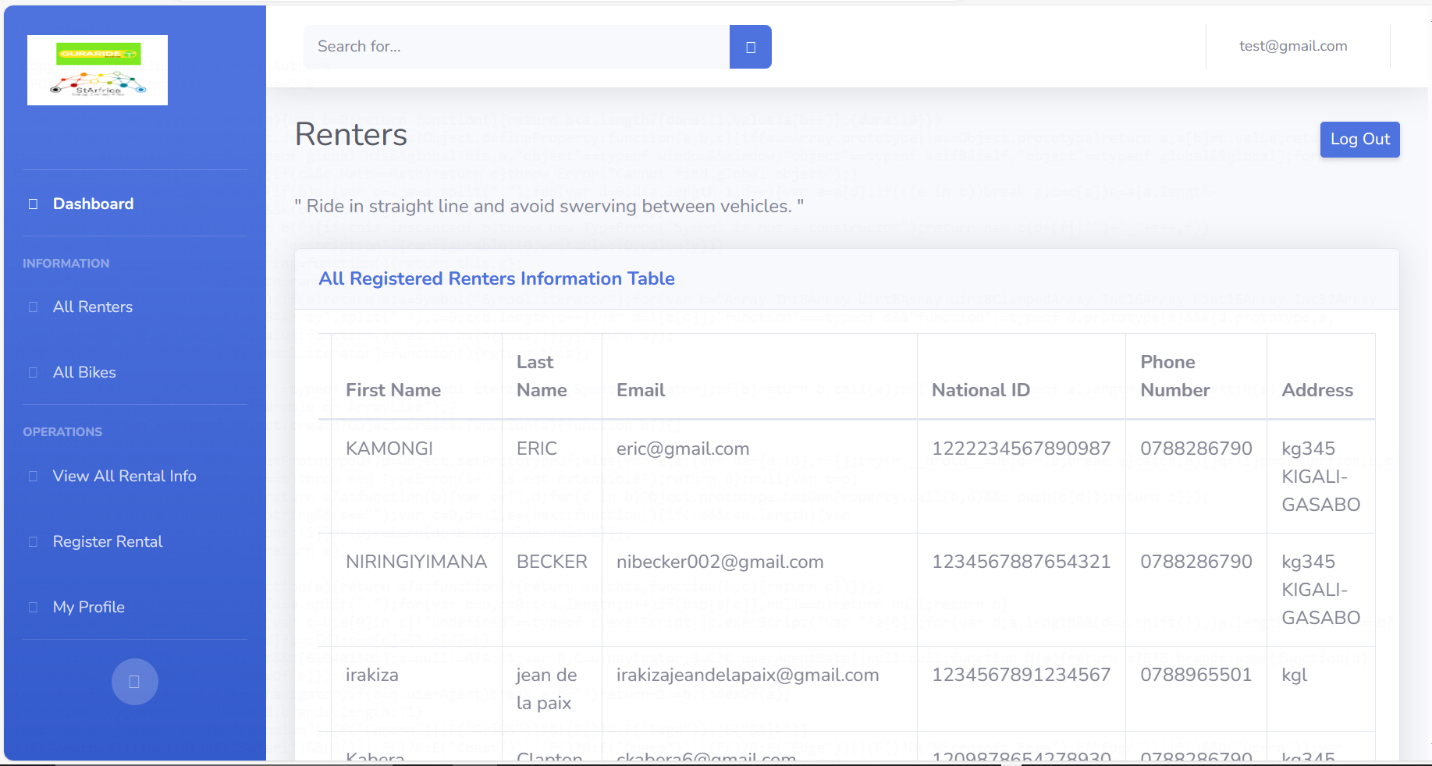
After this now the purpose was to edit the price Then it become:



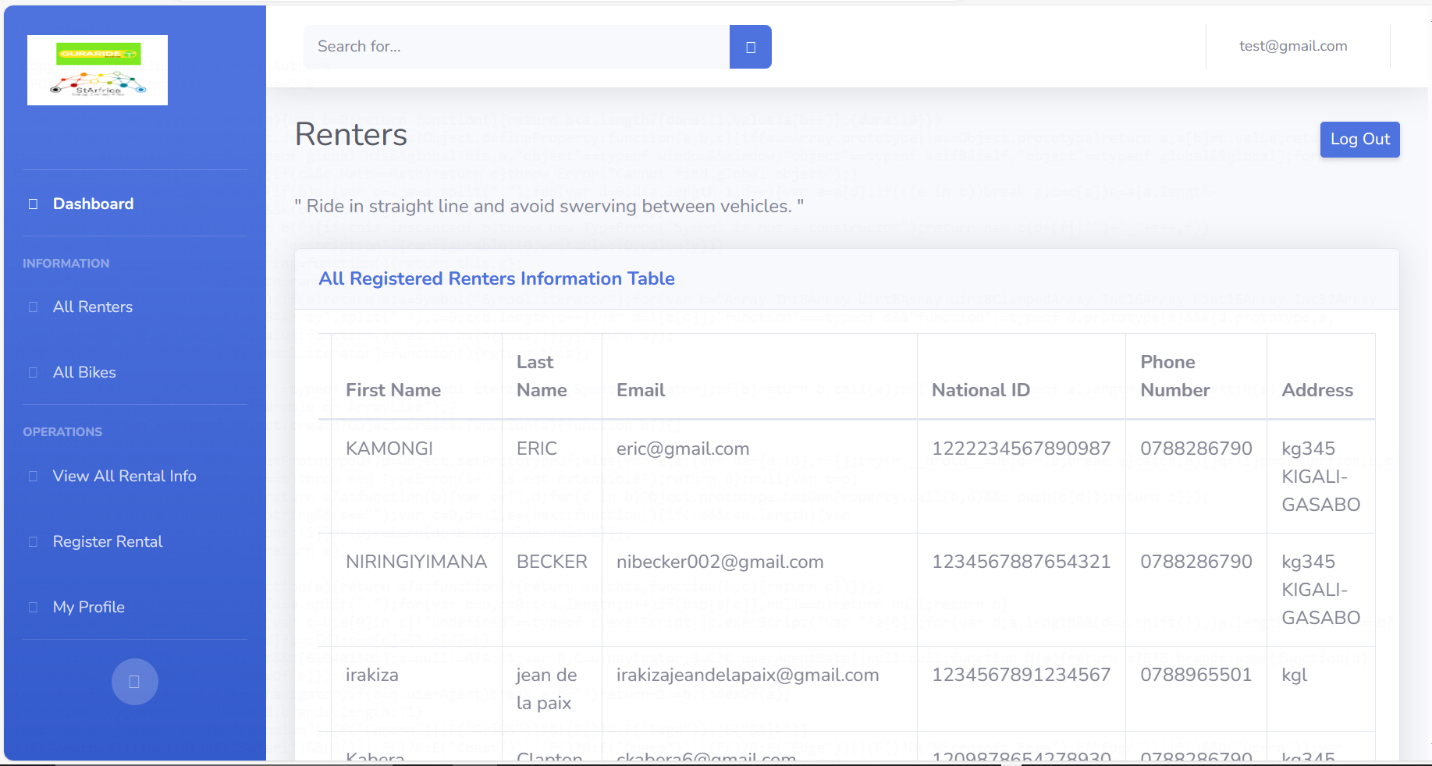
The Guraride Staff dashboard view the admin account is [test@gmail.com](mailto:test@gmail.com)



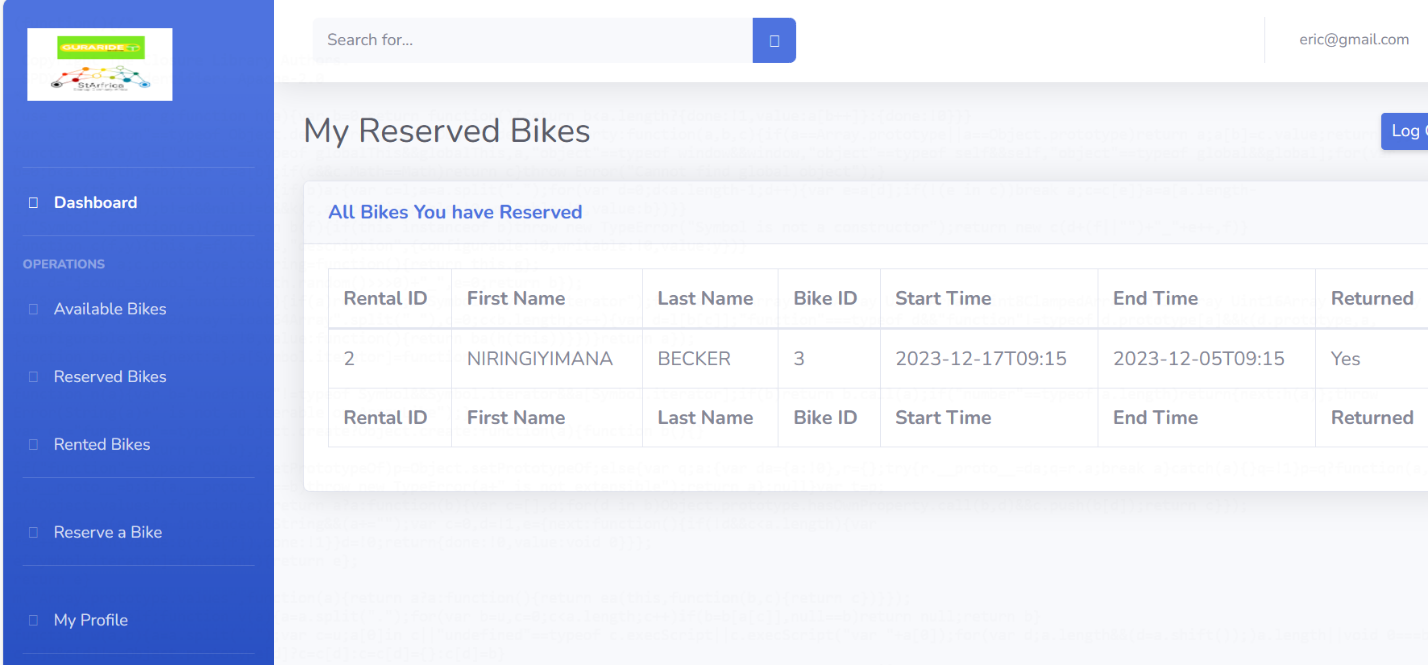
The Staff can view all the users who have rented or reserved the bikes, with their information



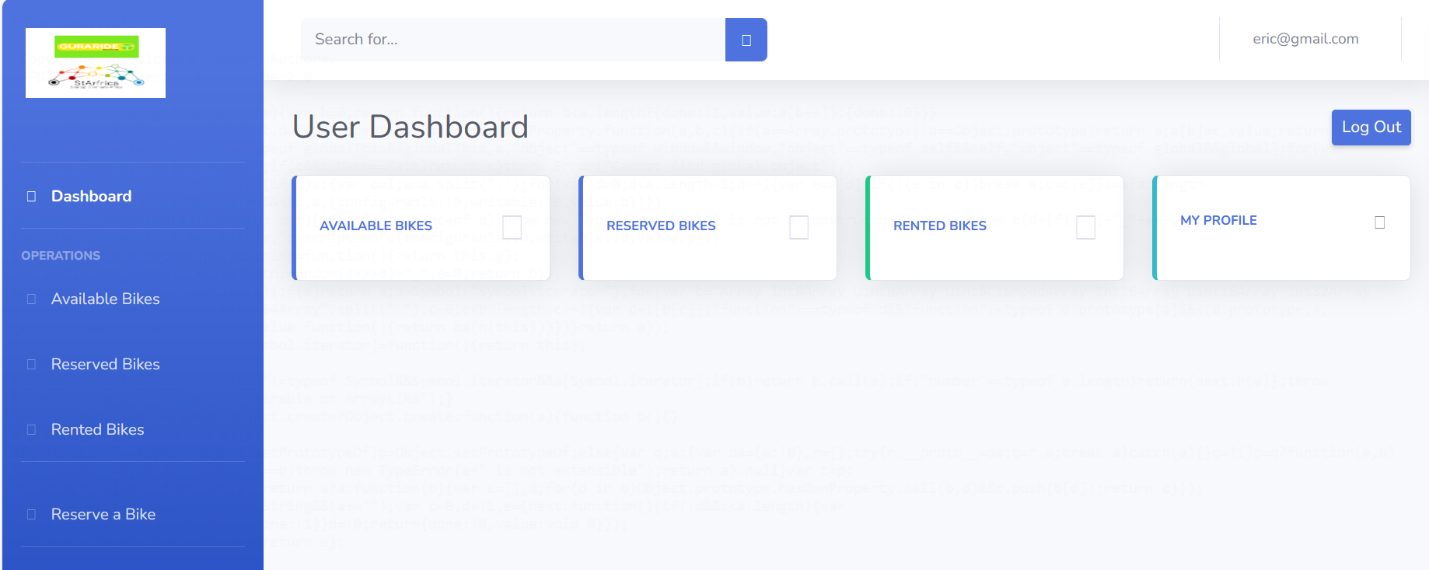
The Guraride Admin can also view all bike information and be able to delete/edit the bikes and also register new bikes.



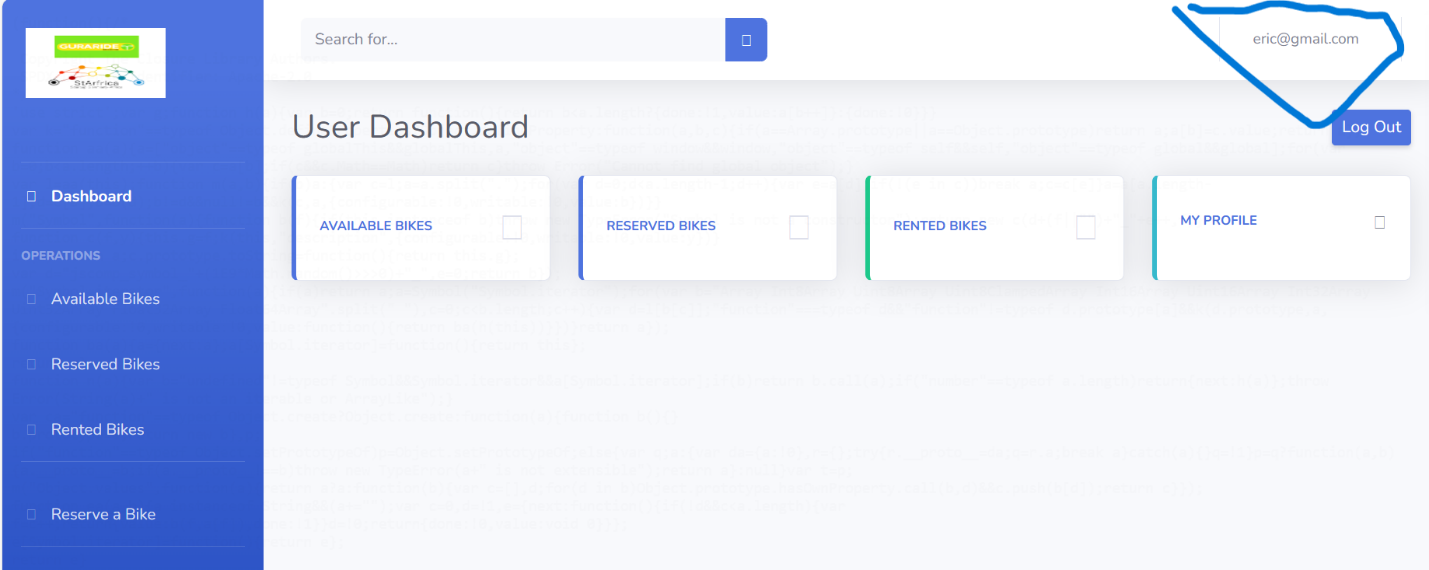
The GuraRide Admin will be the one to check and approve if bikes have been returned by the users, and record them here. For all the bike which have not been returned there will be that button for the staff to approve and the time the users returns the bike



So for the user side :

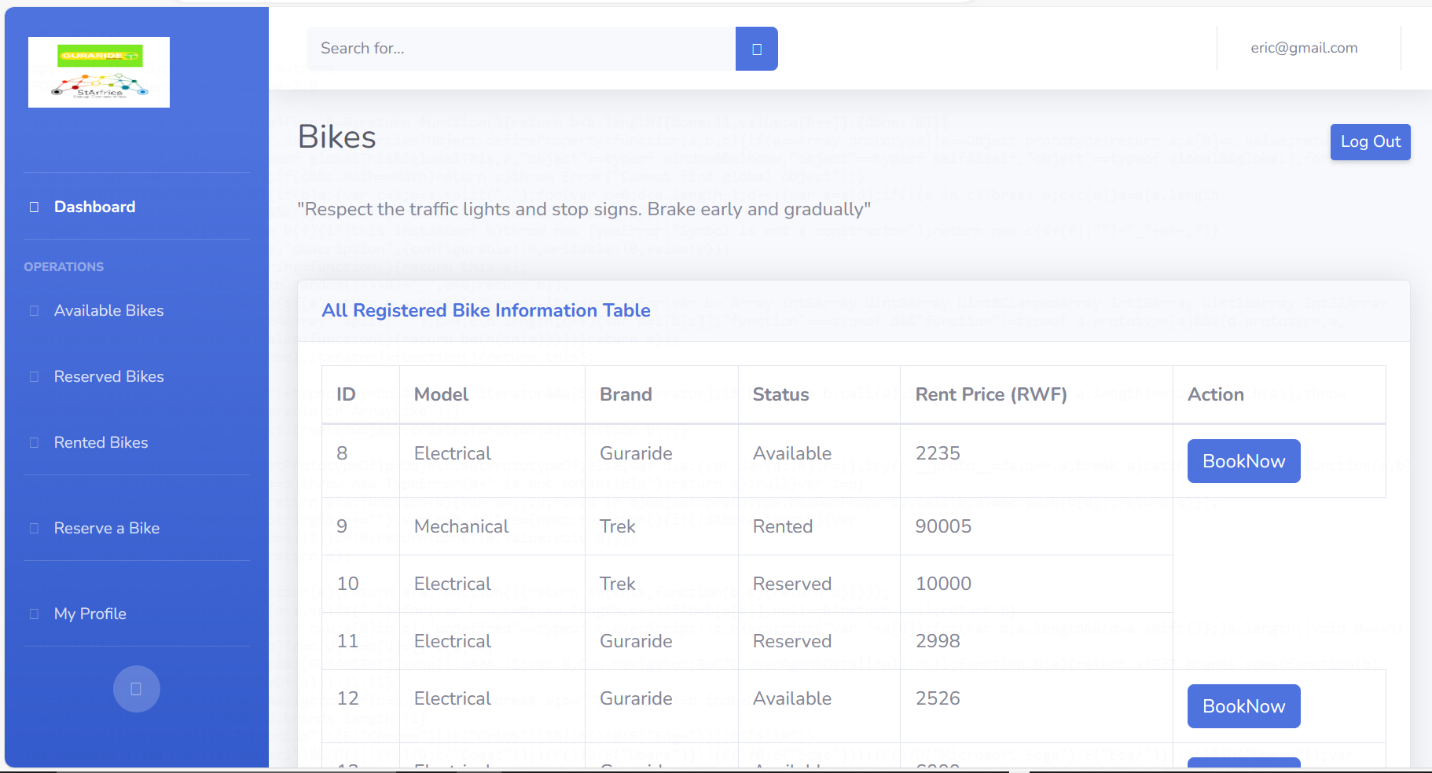


Here the the user account used was [eric@gmail.com](mailto:eric@gmail.com)



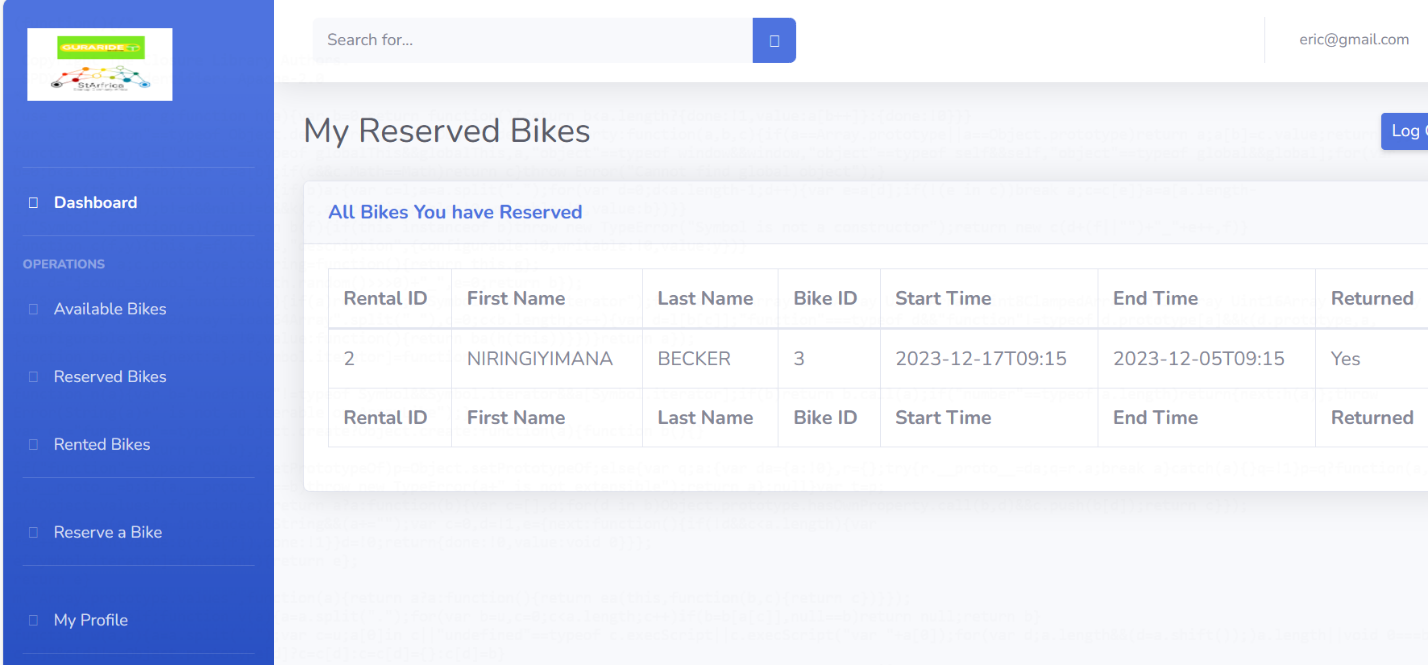
The User will be able to view the dashboard, view all the available bikes, reserve a bike and view information about his/her history of the bikes rental.

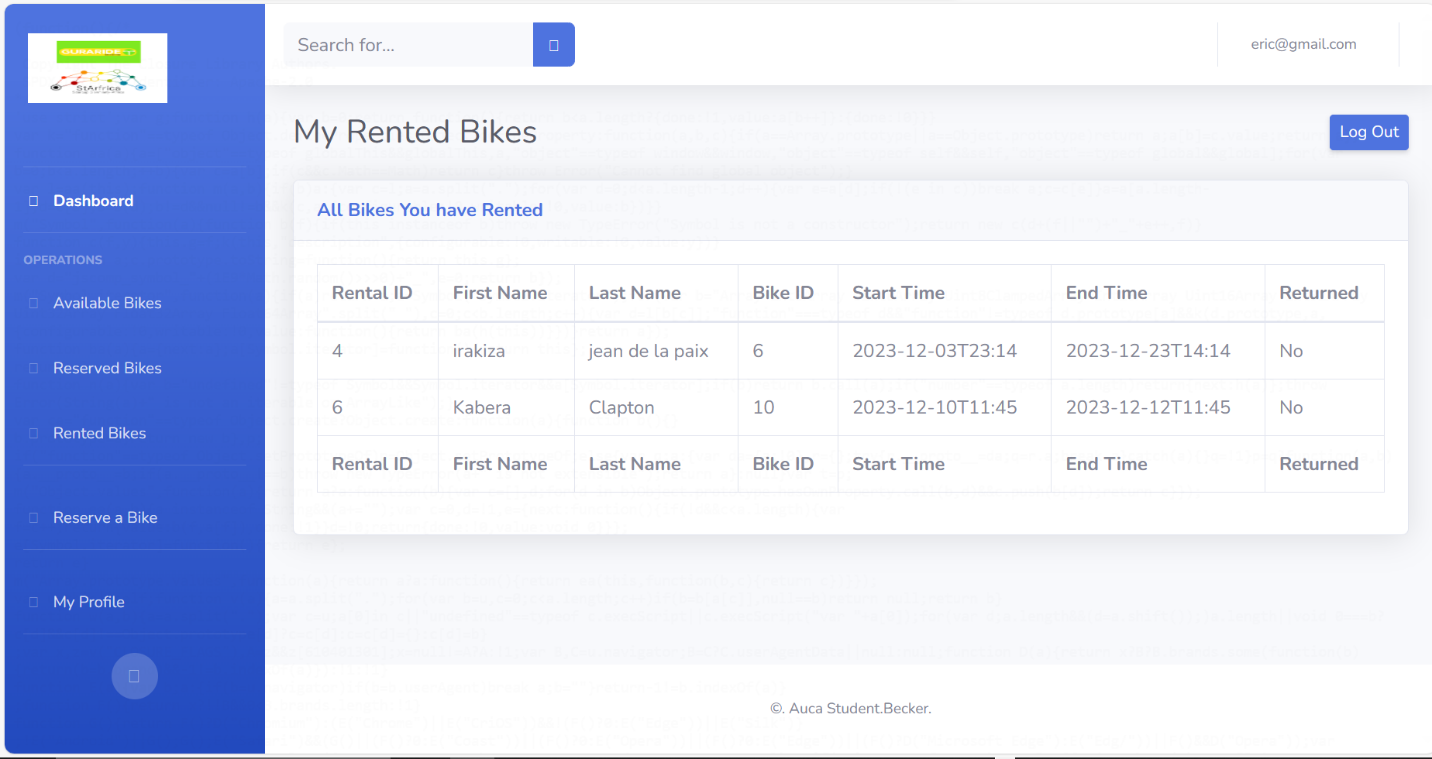
The user can view all bikes that are available to be rented and booked



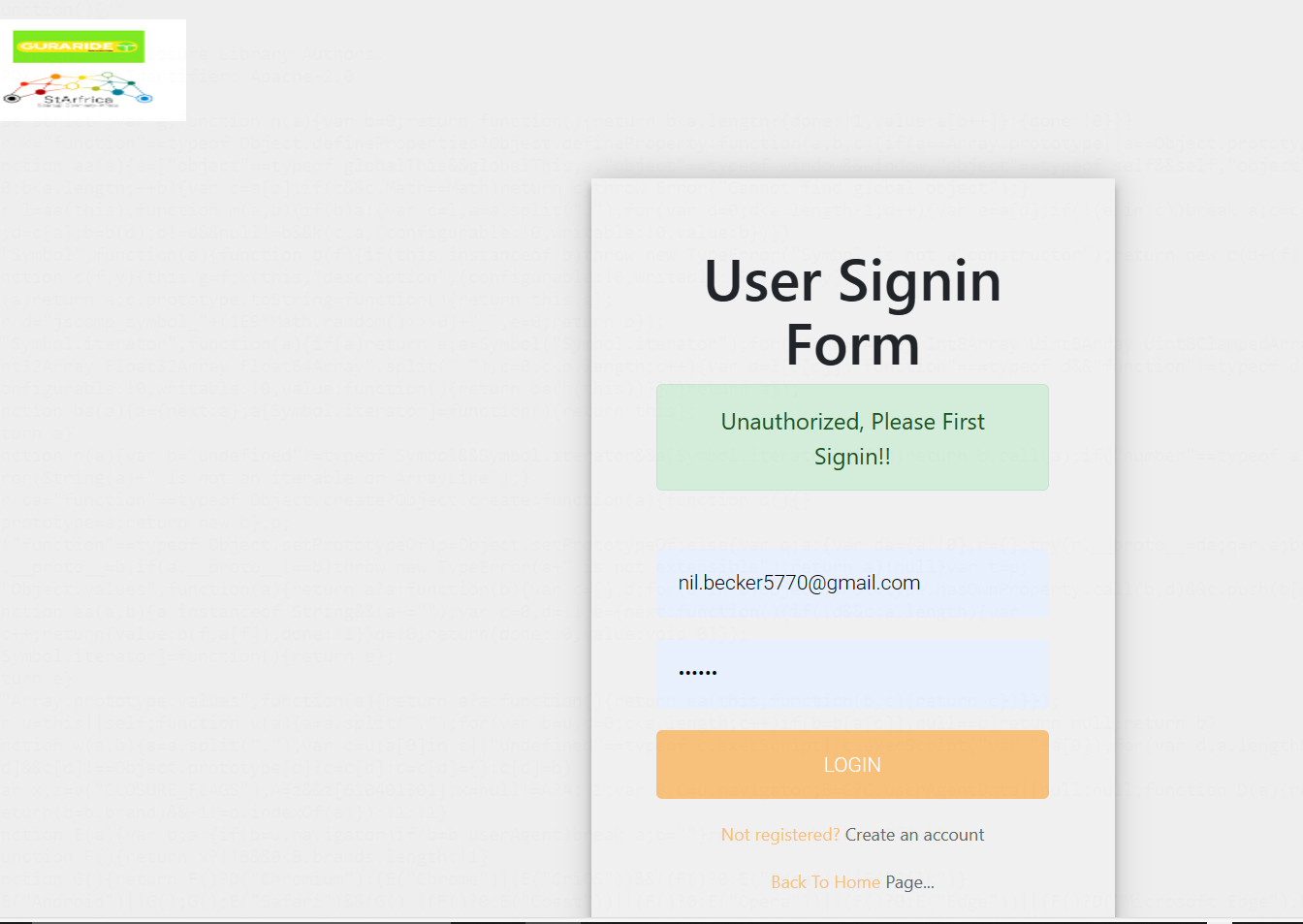
If the user see that there are available bike she/he can press the **Book now** button

Here he can view what she/he booked



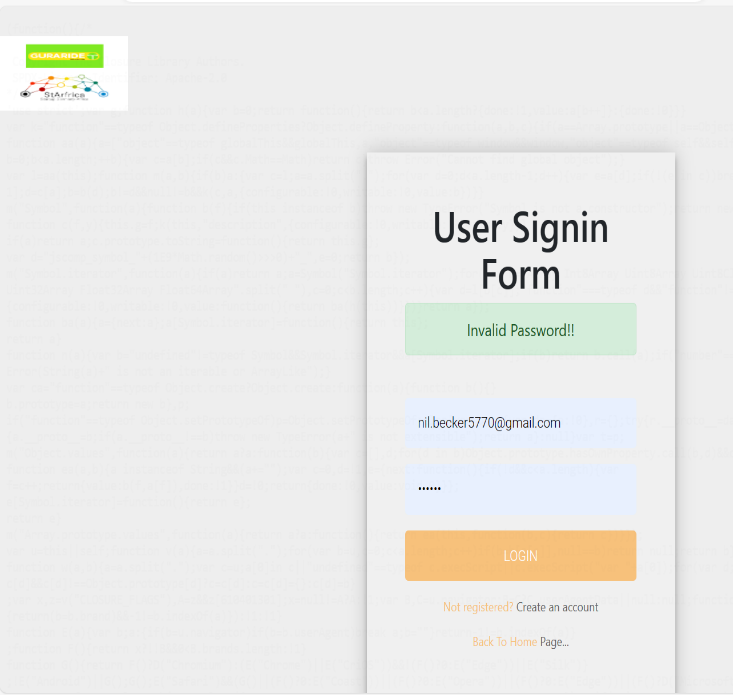
The User view of the rented bikes, and the status whether their returned them or not

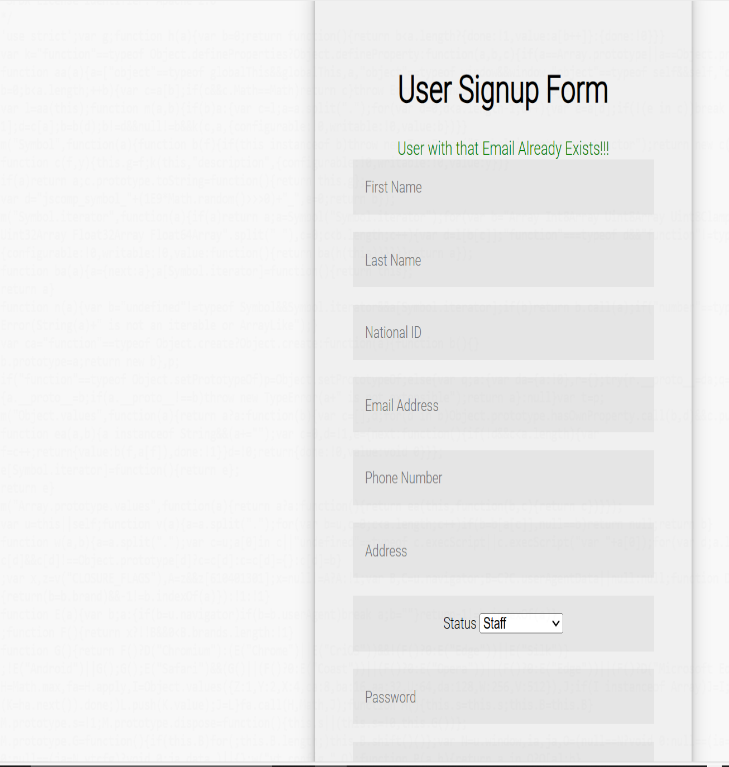
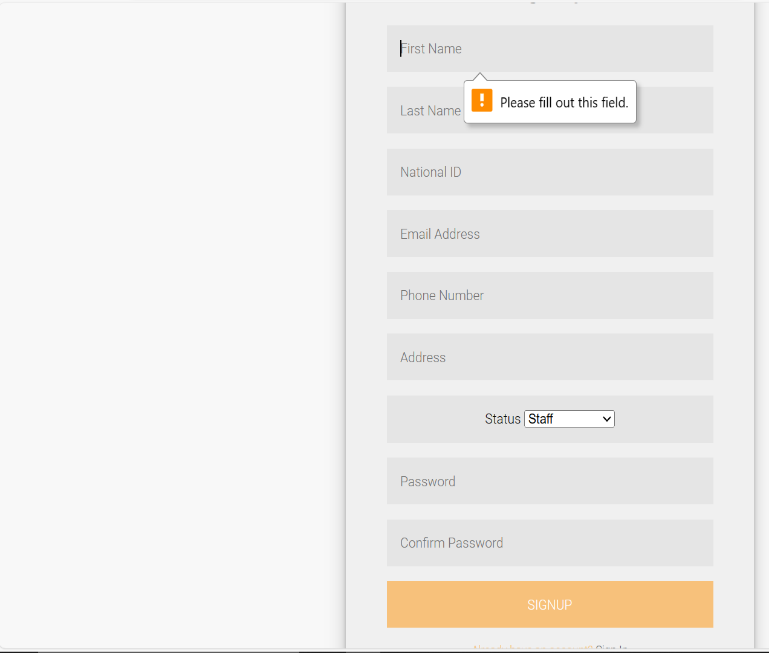
For the addition all the user should be able to logout to the system and I have applied authorization rules to privileged paths so that only privileged user can view the page.



The validations:

Also I have applied validation on the form submission to ensure that inputs from the user side are well validated, both when the password is not correct or user don’t have an account.



**Languages that were used in this project**

For the backend I have used Java and I have used HTML, CSS for the backend

**Frameworks/modules and dependencies that were used in the project are**

* Spring Boot
* Spring Web
* Spring Security
* Thymeleaf
* Maven
* Lombook
* PostgreSQL

**Source Code:**

I have this application on github and it can be accessed

**(**[**git@github.com:beckereg/SpringMVC-BookingSystem.git**](mailto:git@github.com:beckereg/SpringMVC-BookingSystem.git)**)**

**The Final deployment of my project is on Heroku and can be accessed here.**

**(https://bikebookingsystem.onrender.com)**