**TESTING AND SUMMARY OF RESULTS**

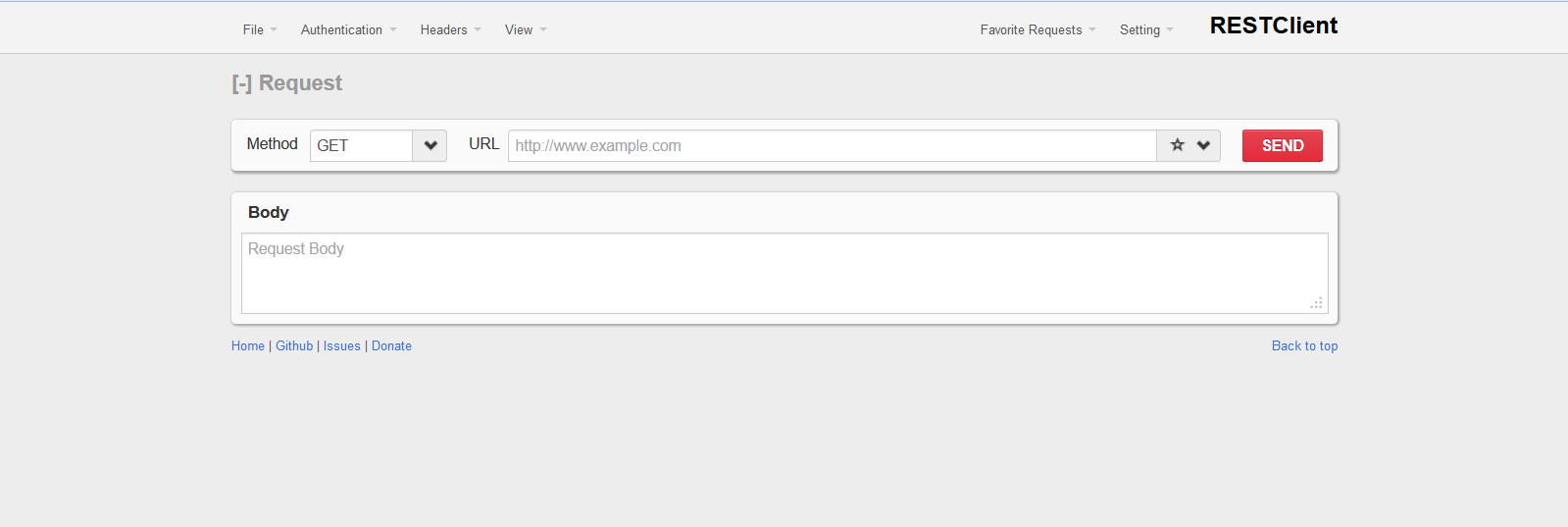
Software testing can be stated as the process of validating and verifying that a computer program/application/product:

* meets the requirements that guided its design and development,
* works as expected,
* can be implemented with the same characteristics,
* and satisfies the needs of stakeholders.

Software testing, depending on the testing method employed, can be implemented at any time in the software development process. A primary purpose of testing is to detect software failures so that defects may be discovered and corrected.

* 1. **Tool Used**: **RestClient[9]**

Rest Client is the testing tool used to test the web services of the web applications. This is basically used to test the web services created by using RESTEasy Web Services. The screenshot showing the RESTClient is as shown below:

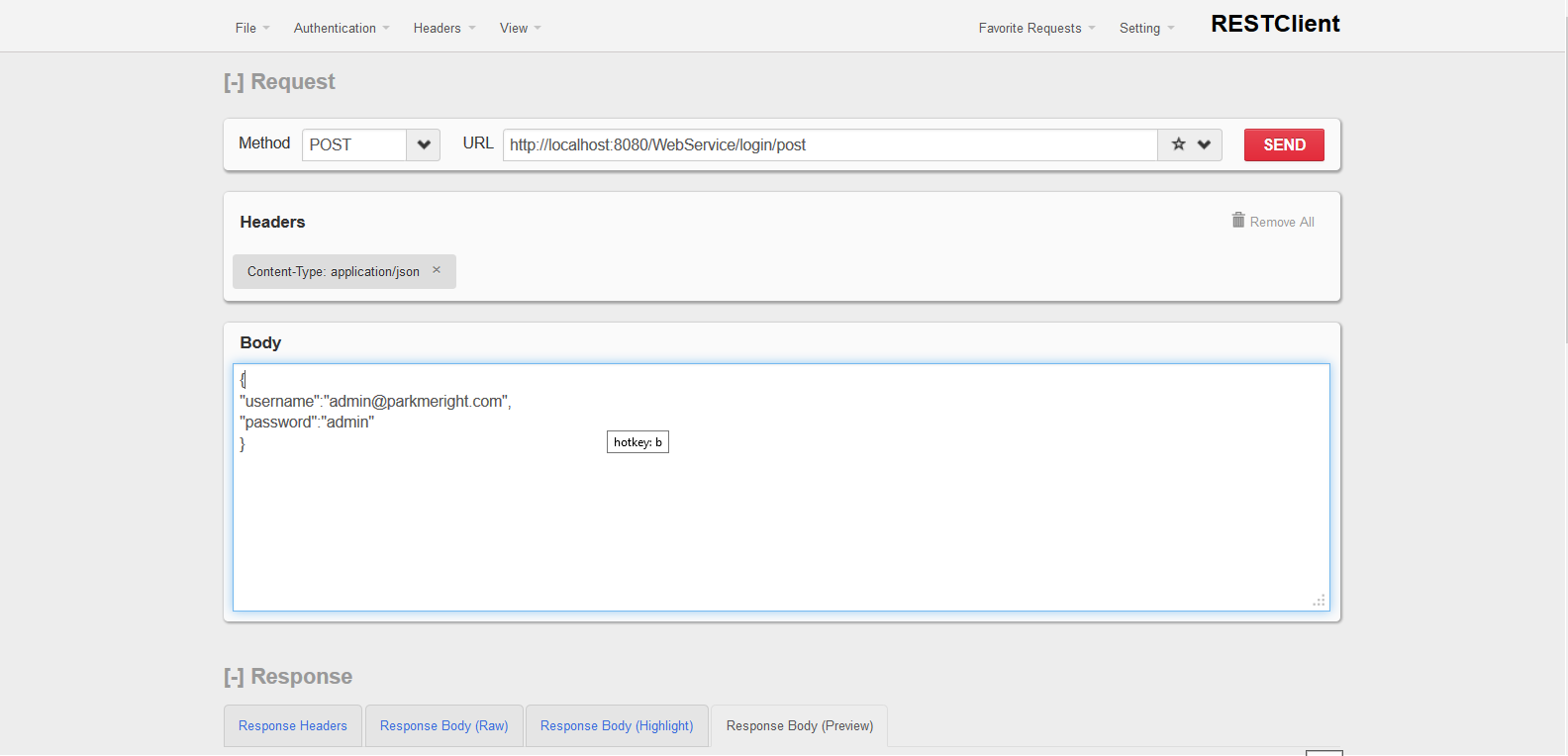


**Figure 7.1: RESTEasy Client**

It provides with the client side environment to test the web services. It provides with the URL bar, content type space, request and response area to input and receive the contents as a request and response respectively.

Various testing done on the web services of the web application, to test the connectivity with the android application are as follows:

1. **Authentication Web Service**

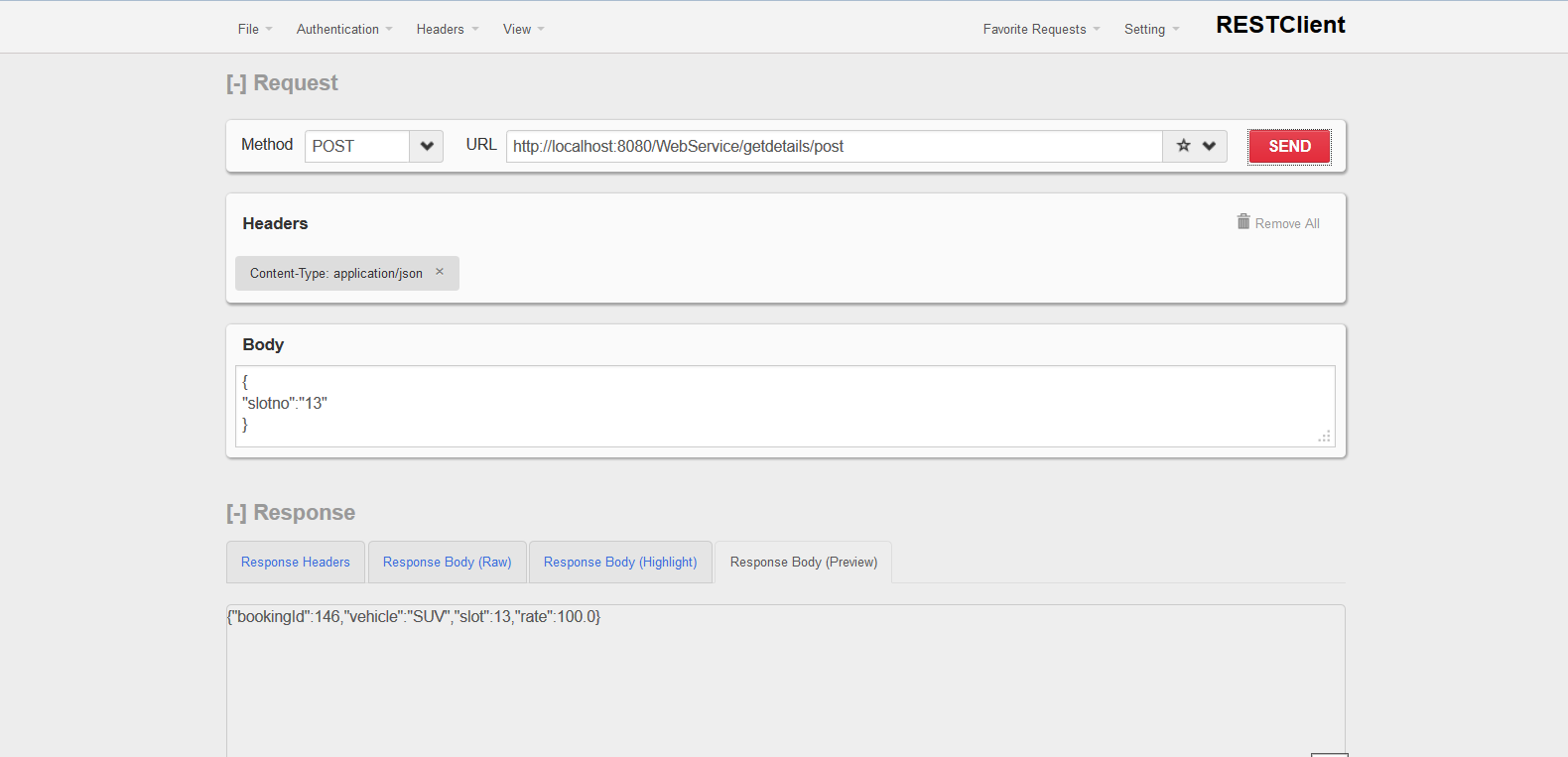
****

**Figure 7.2: Login Web Service**

The Web Service at the mentioned URL connects to database, accepts username and password as a request and returns true or false as a response. If the user is authenticated then a “true” value is returned or else “false” value is returned as a response to the RESTClient.

The request body and the type of content the web service accepts of consumes is as shown in the figure.

1. **Getting details of a particular slot**

****

**Figure 7.3: Get Details Web Service**

This web service accepts slotno as a json and provides with the details like bookingid, vehicle, slot and the rate for that particular slot.

The testing was done and following was recorded:

Request:

Json Format:

{

“slotno”:””

}

Response:

*Json Format:*

{

“bookingId”:””,

“vehicle” :””,

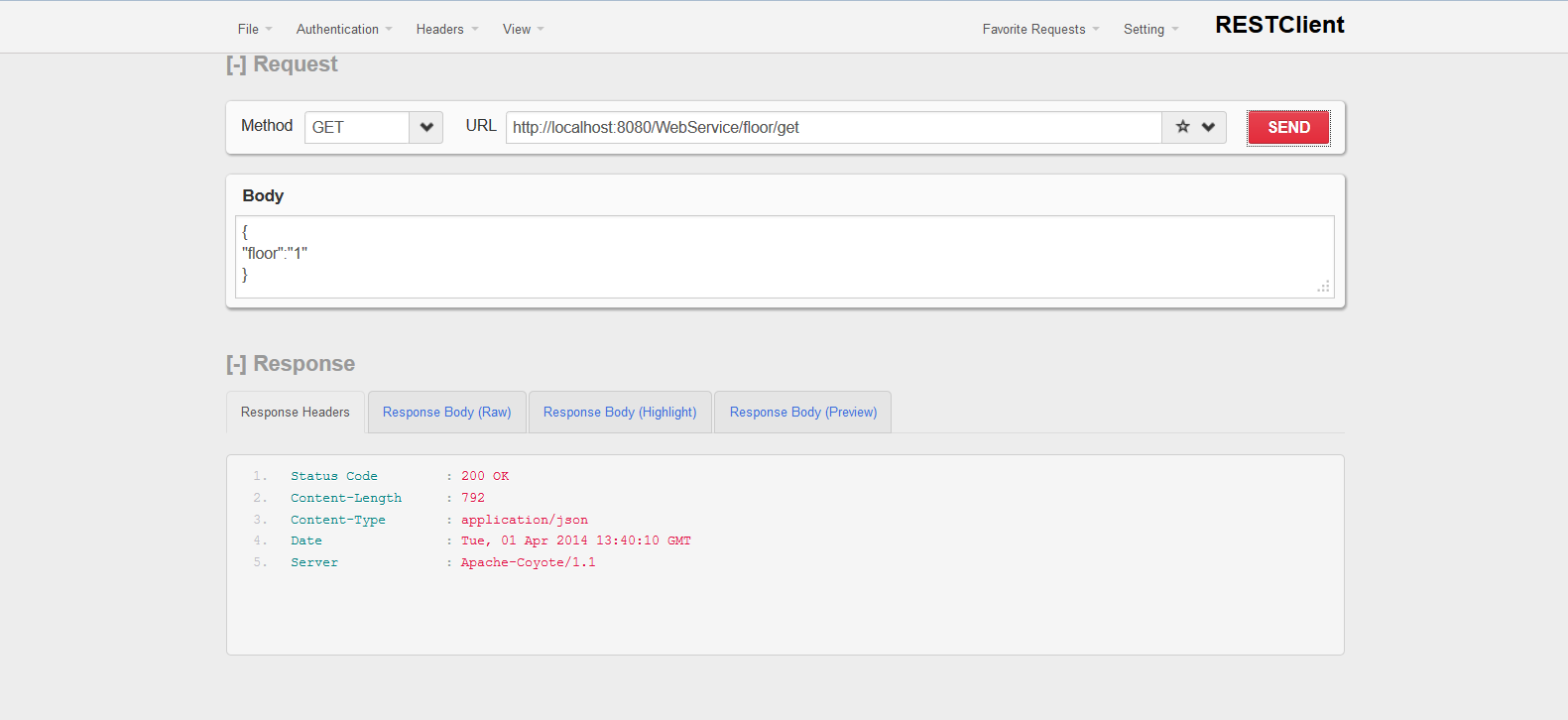
“slot” :””,

“rate” :””

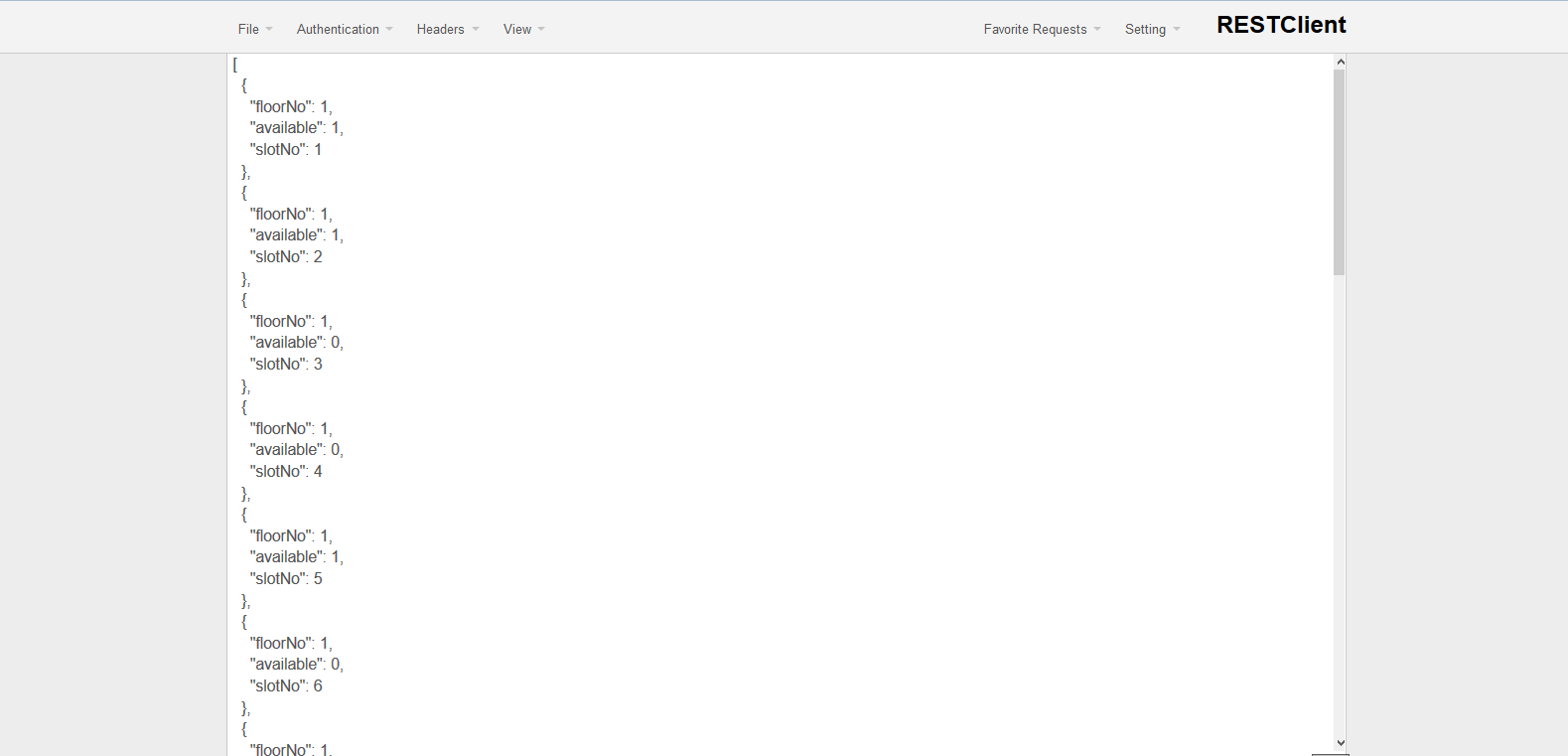
}

1. **Fetching floor details**

This WebService provides with the details of parking slots related with a particular floor.

****

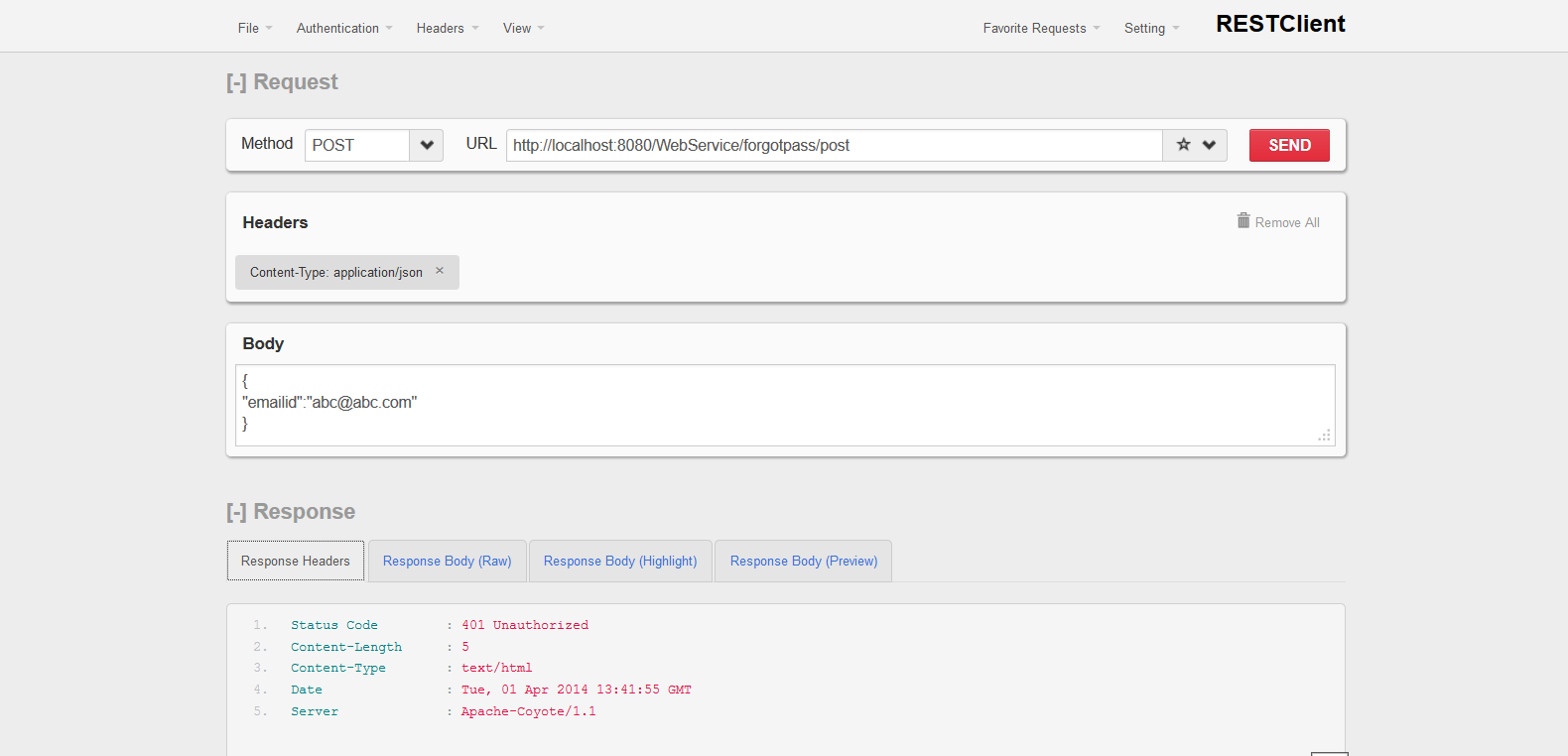
**Figure 7.4: Request to fetch floor plan web service**

****

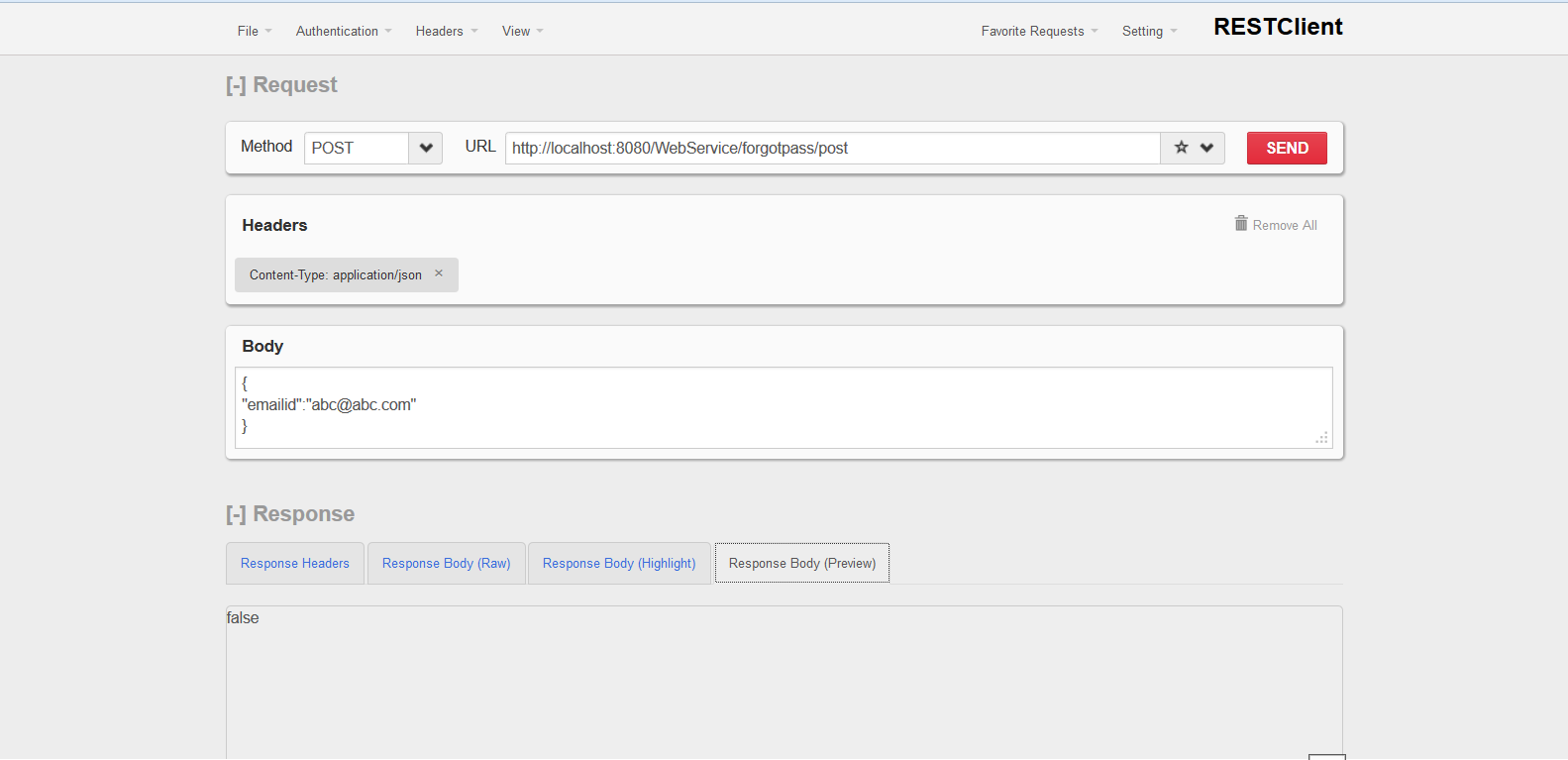
**Figure 7.5: Response of the fetch floor plan web service**

1. **Forgot password Web Service**

Web service to check the correctness of a particular email id entered by the user

****

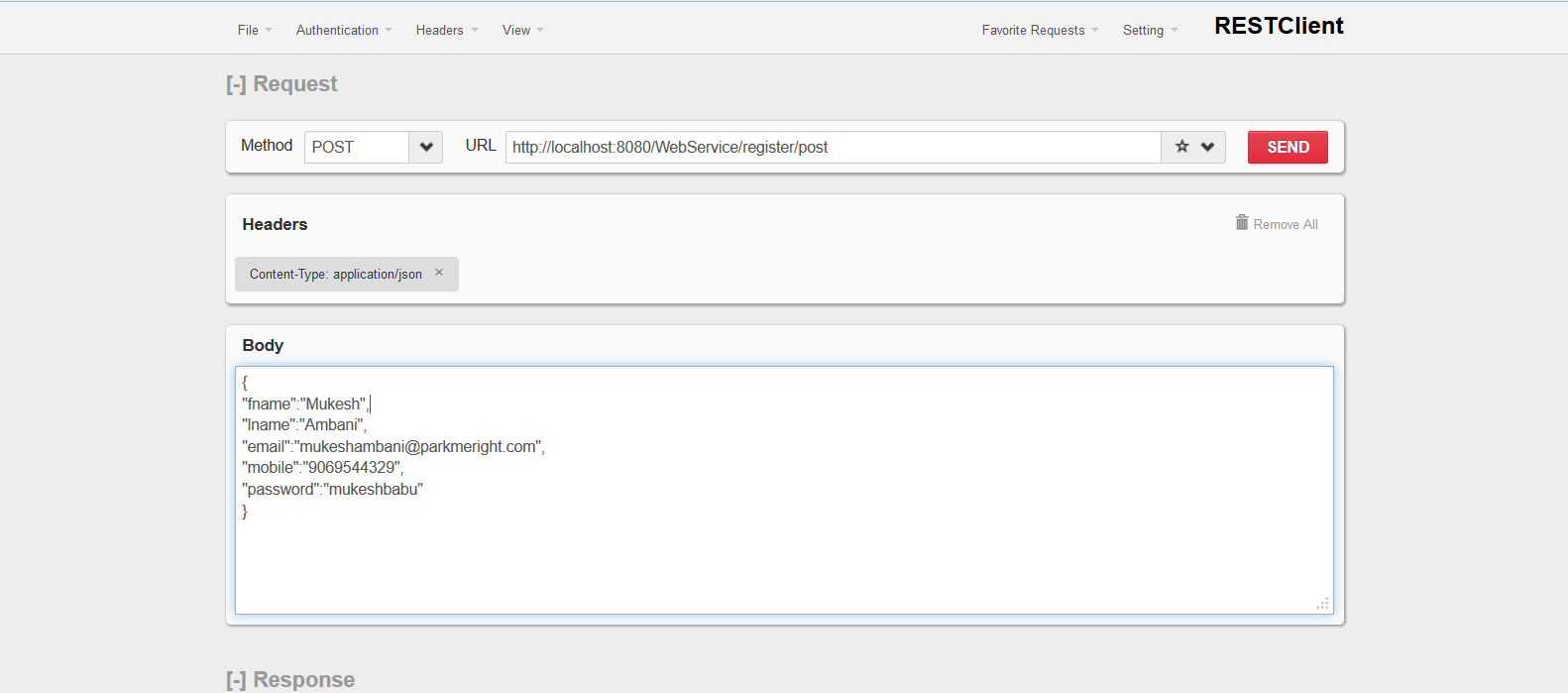
**Figure 7.6: Request to the forgot password web service**

****

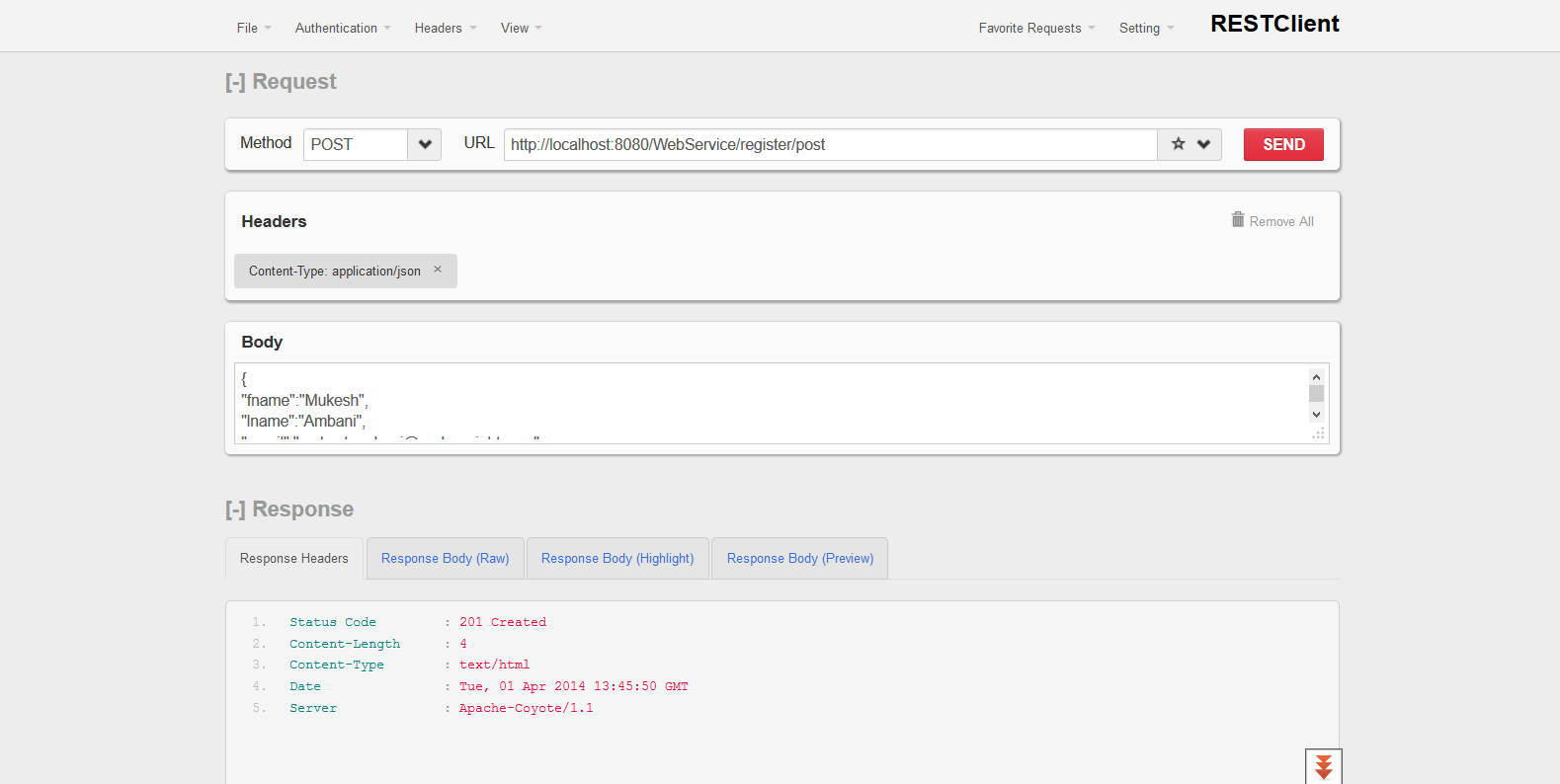
**Figure 7.7: Response from the forgot password web service**

1. **Registration Web Service**

This web service handles the request for user regisration data. If database is successfully updated then true is sent to client in response or vice versa.

****

**Figure 7.8: Request for submitting user registration data**

****

**Figure 7.9: Response by the registration web service.**

* 1. **Results**

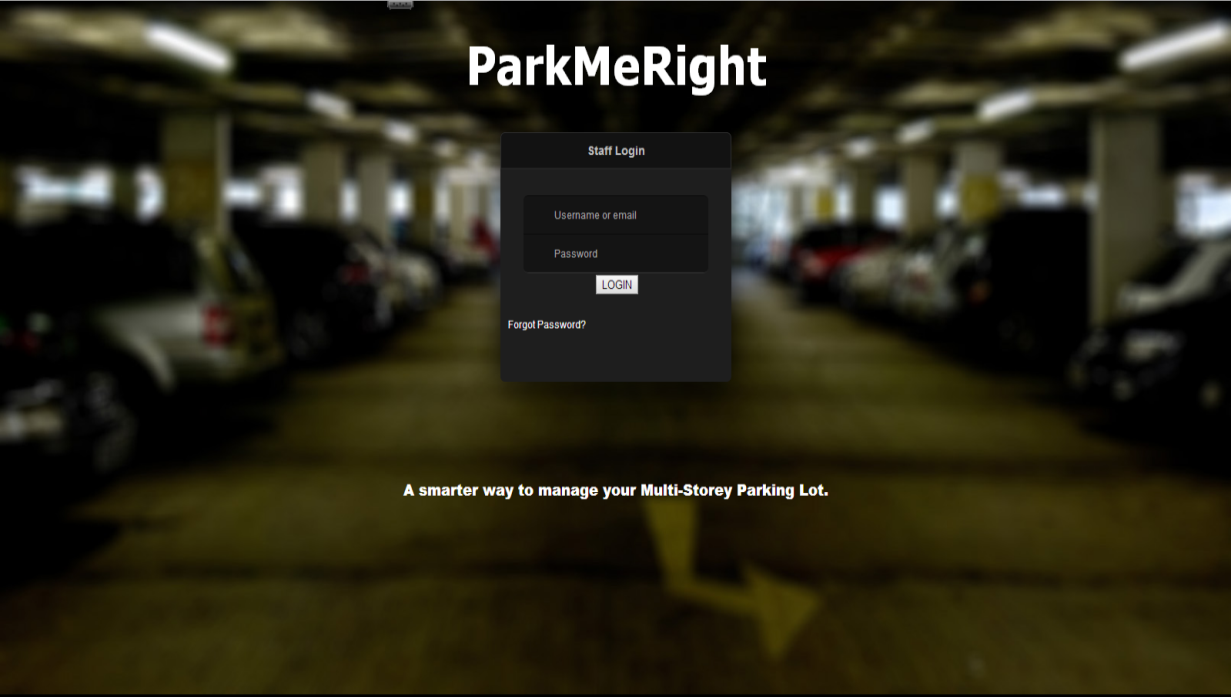
The screenshots showing the working and result of the project are as follows:

Note: The actual results and the GUI look and feel would be different depending upon the actual underlying hardware and version of the software packages being used.

* + 1. **Web Application**

1. **Login Page**

The current and trending design technique called the Flat UI design is implemented.

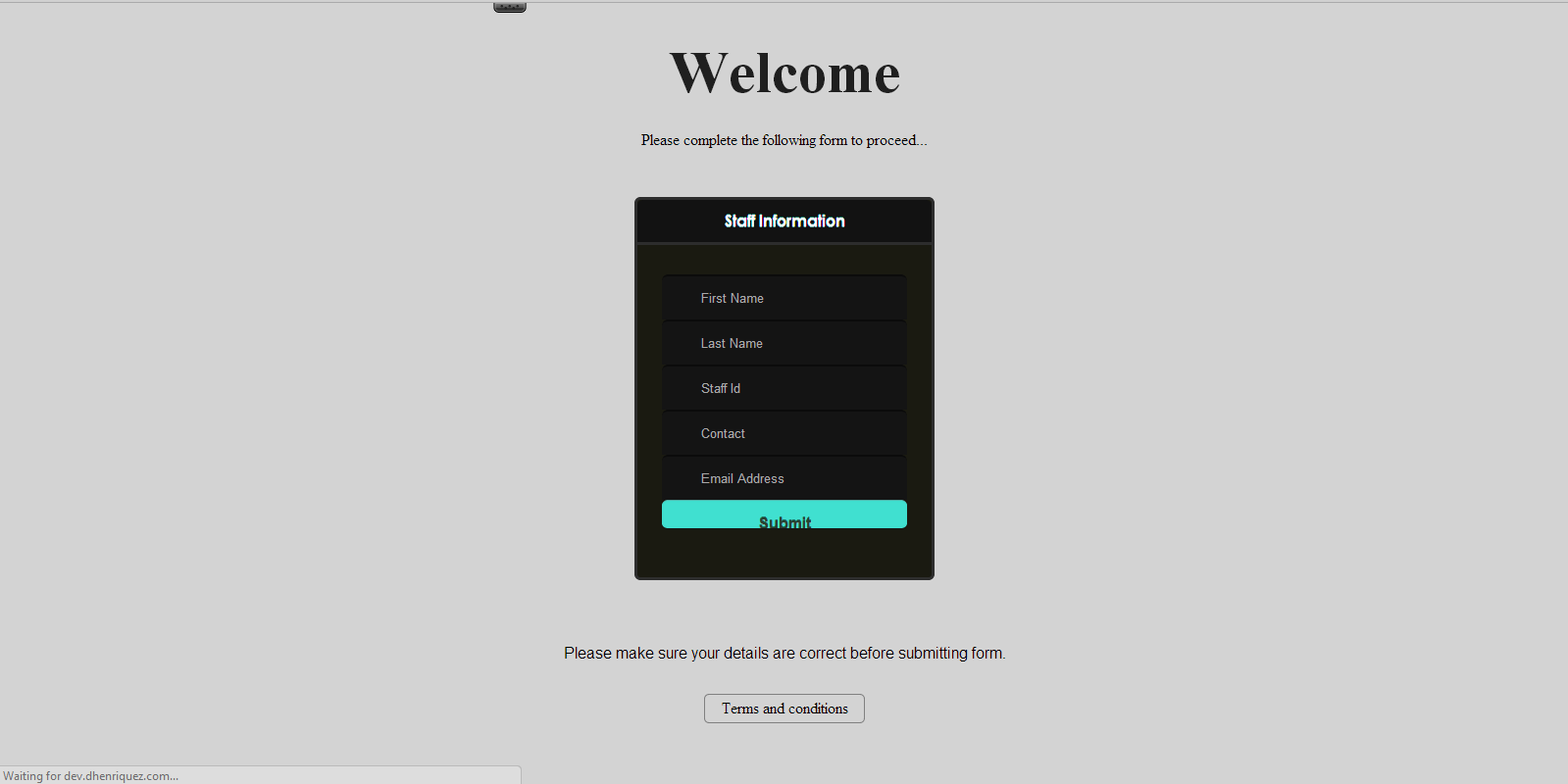


**Figure 7.10: Login Page**

The operator of the the parking lot is allotted a password and a username by his/ her Admin or the In charge of the parking lot or the parking area. The operator needs to input his/ her login credentials to be able to operate the parking lot web application. On successful login he/she is directed to the main page of the application.

1. **Registration Page**

If the user or the operator is new and has not filled in registration details yet, he/she has to fill in the details before moving to the main page of the application.

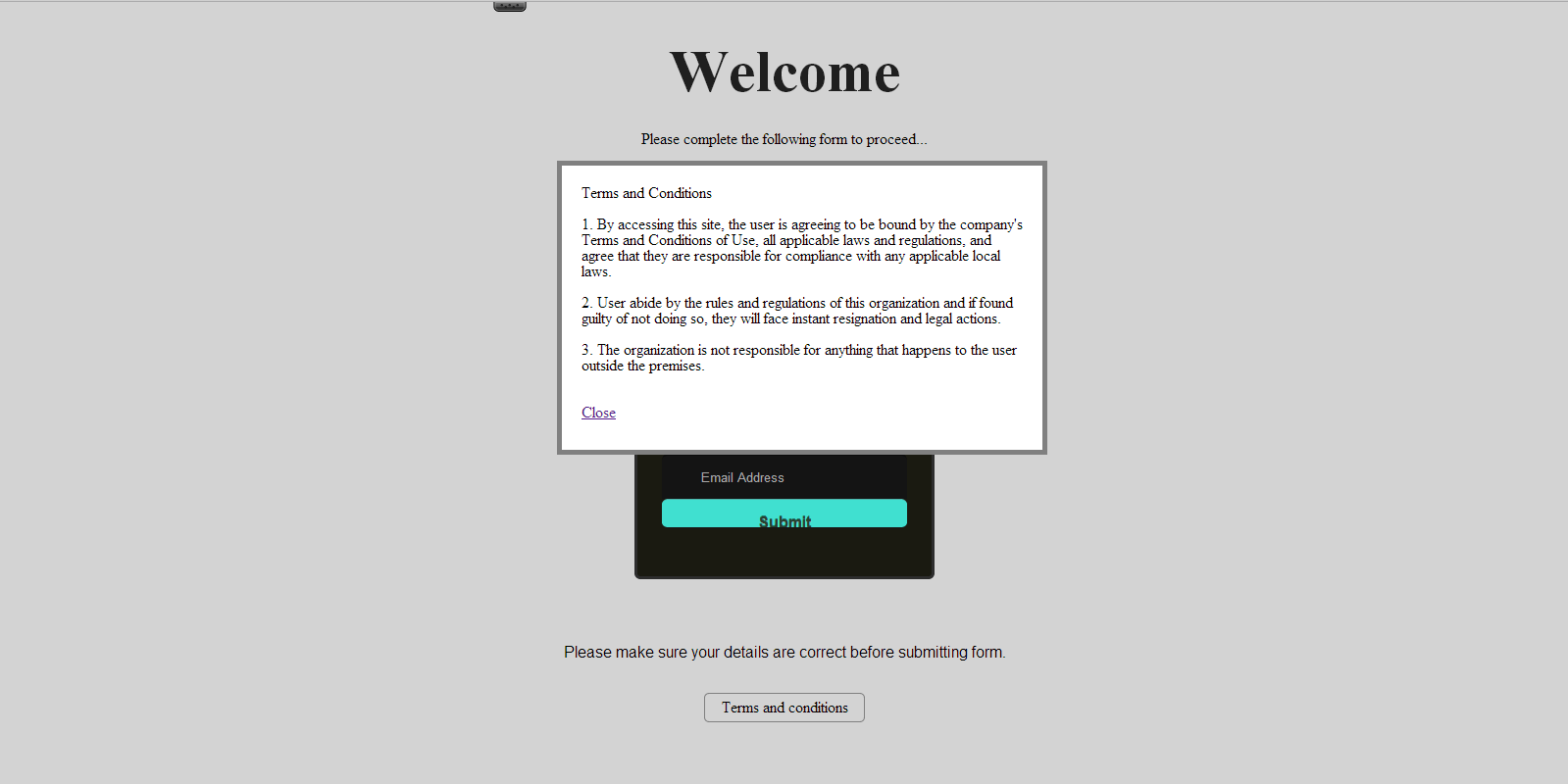


**Figure 7.11: Sign Up Page**

The user or the operator has to input his personal information like the first name, last name, phone no, staff id and email id. After successful entry of the given information to the database, he/she is directed to the main page of the application. There is a term and conditions button as well. Thse user can view the terms and conditions and promises to abide by the same.

1. **Terms and Conditions**

The user is provided with a list of terms and conditions. The user has to abide by the rules or the terms and conditions to continue to work with the application.

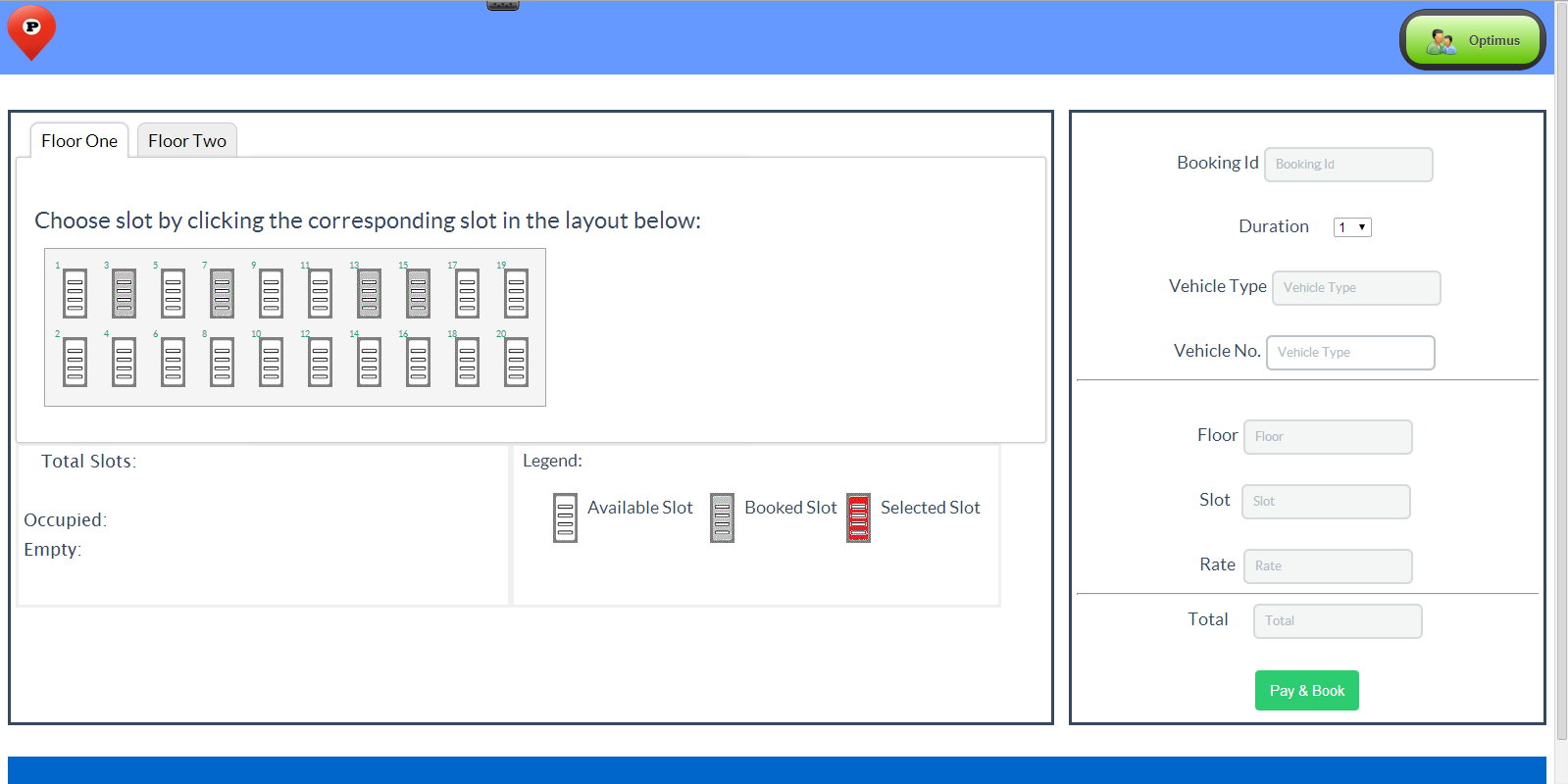


**Figure 7.12: Terms and Conditions**

A checkbox can be provided there so that the user cannot be allowed to move or be redirected to the main working page of the application. This is somewhat similar to validation where a user is allowed to move to next page only if the checkbox is ticked.

1. **Main Page**

This is the main page of the web application. All the parking area management is done through this page. On successful login or successful registration, the user is directed to this page. There’s no admin page provided as of now.



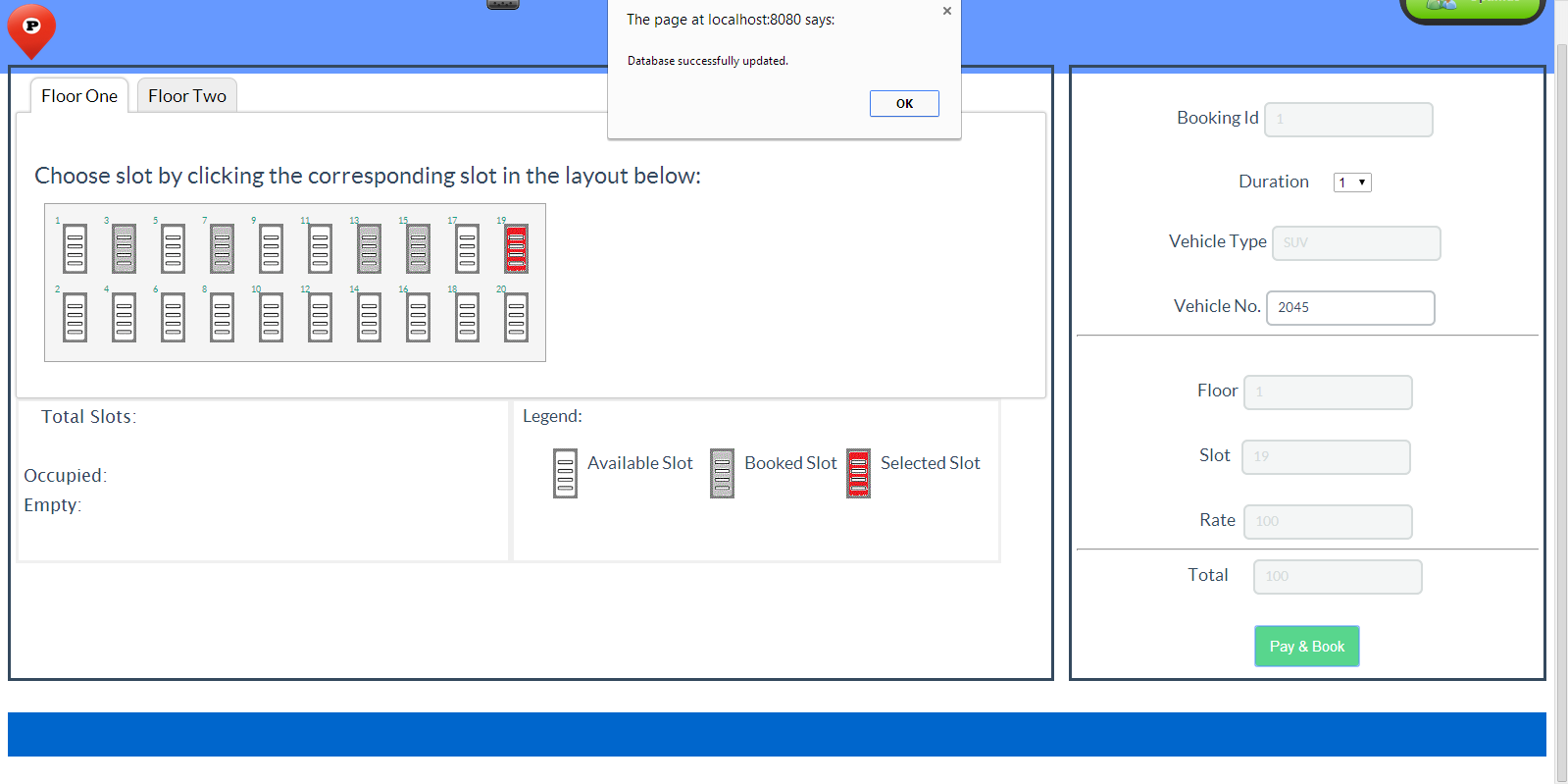
**Figure 7.13: Main Page**

On the left hand side the floor wise parking slots are displayed. The operator has to click on a particular Floor Tab, as shown on the top of the left panel. Based on the floor number selected, the parking slots are displayed. The parking slots may be empty, booked or already occupied. The available slots are shown via white parking slot image, the booked are shown using grey parking slot image and the currently selected parking slots are displayed using the red parking slot image.

Total number of slots, total number of empty and total number of occupied slots status is shown on the bottom left corner of the page.

1. **Booking a slot**

This is the way an operator selects a slot for a particular customer on demand, and generates bill or receipt for the same. The operator needs to select the floor first. Different floors house different types of cars or vehicles. Usually the lower level floors are meant for heavy weight vehicles and the higher level floors are meant for the lighter vehicles.

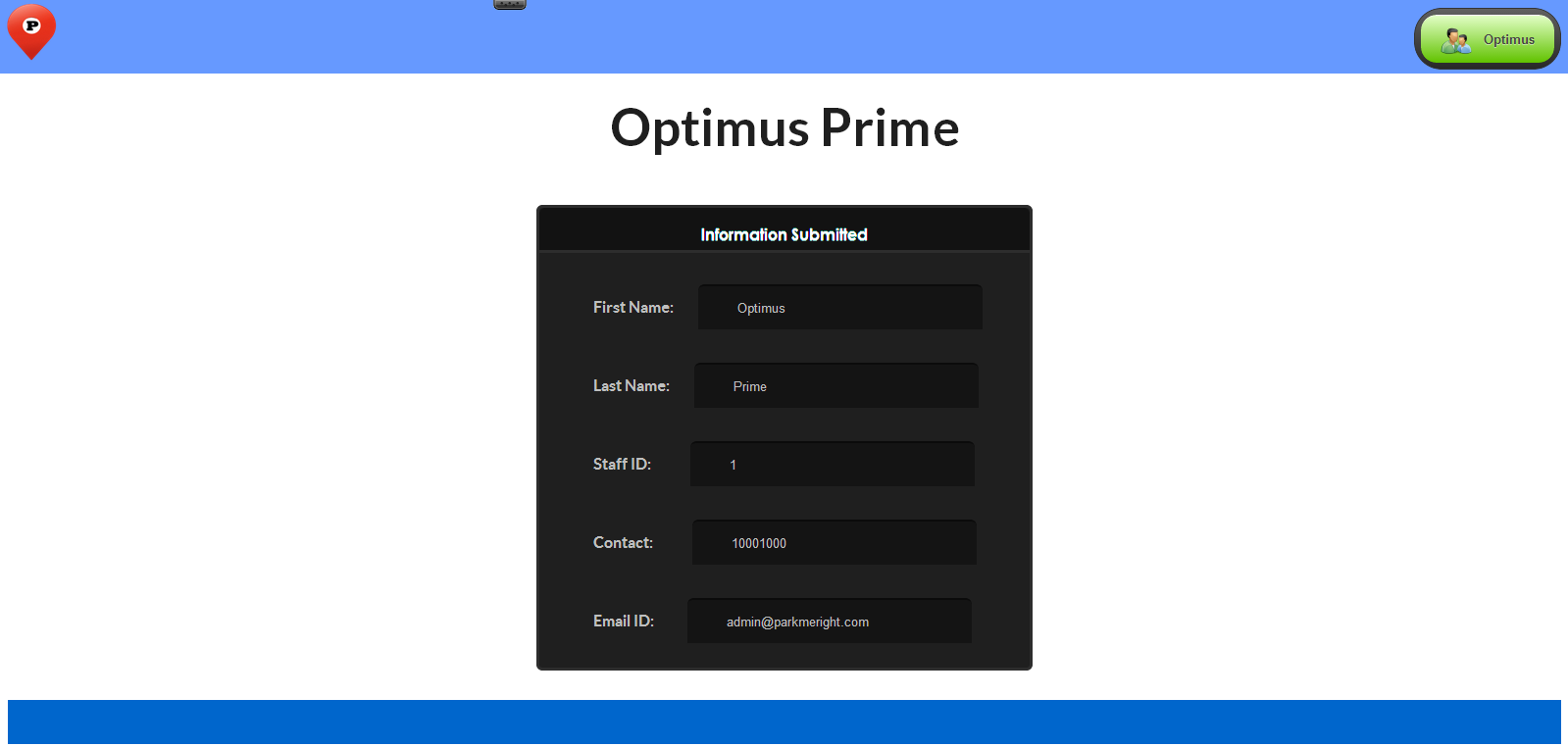


**Figure 7.14: Booking a slot**

The operator needs to select one slot out of the many available slots for parking. After the selection is made, he/she has to enter the remaining details in the text field specified on the right panel of the main page. The details are vehicle number, the total money paid and the entry time. All other details like the booking id, slot number, floor number, rate and the balance are fetched, calculated and displayed automatically. After all this, when the pay and book button is pressed, a bill is generated carrying all the information entered and selected by the operator.

1. **Profile Page**

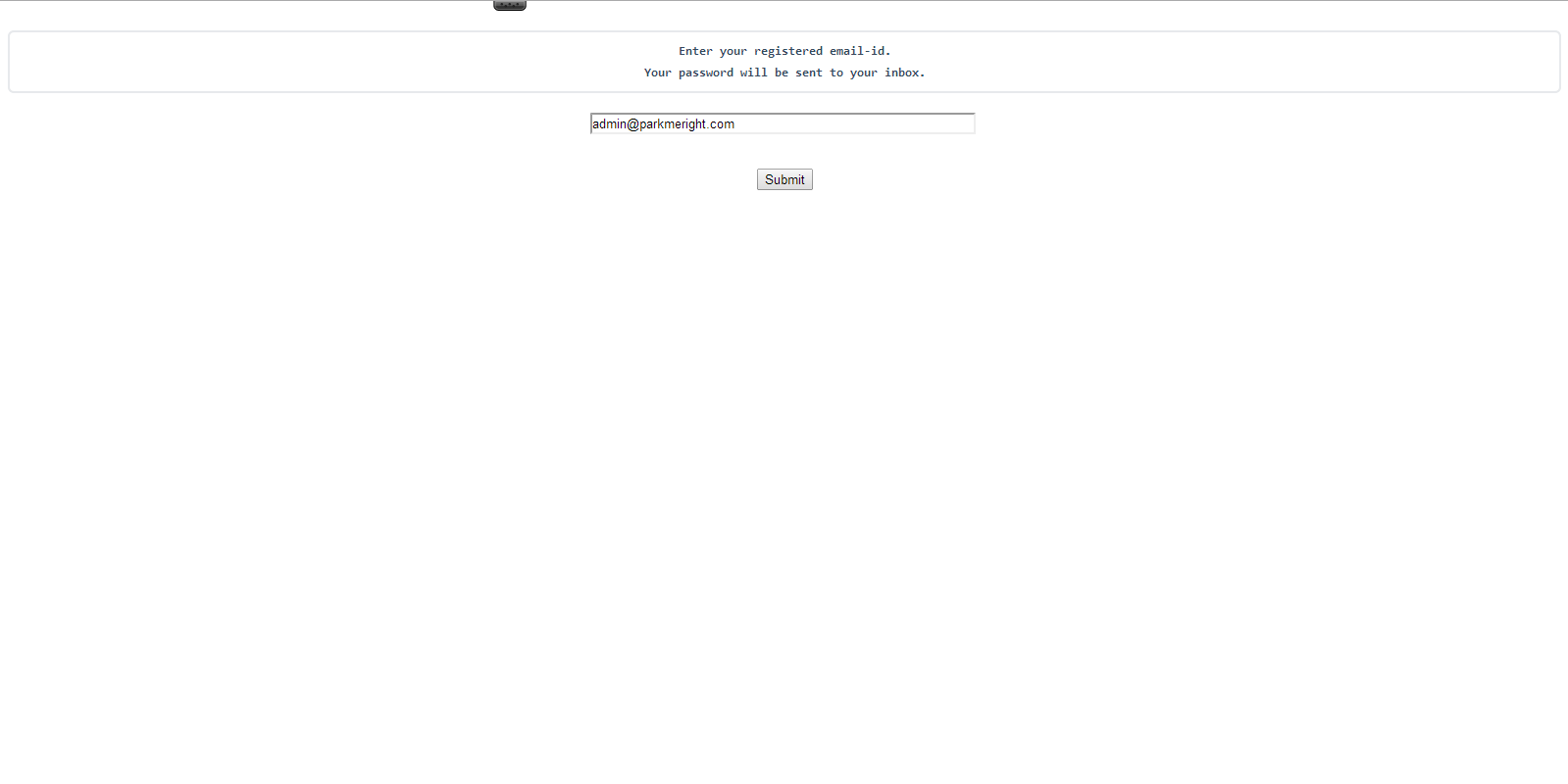
This page displays the stored information of the particular operator currently using the web application of the currently logged in operator.



**Figure 7.15: Profile Page**

This sample page displays the information of the operator named Optimus Prime. The fields mentioned or displayed are not at all editable. They can only be read but no new information can be entered through this page. The updating facility is provided with the admin of the application or the in charge of the parking lot.

1. **Forgot password facility**



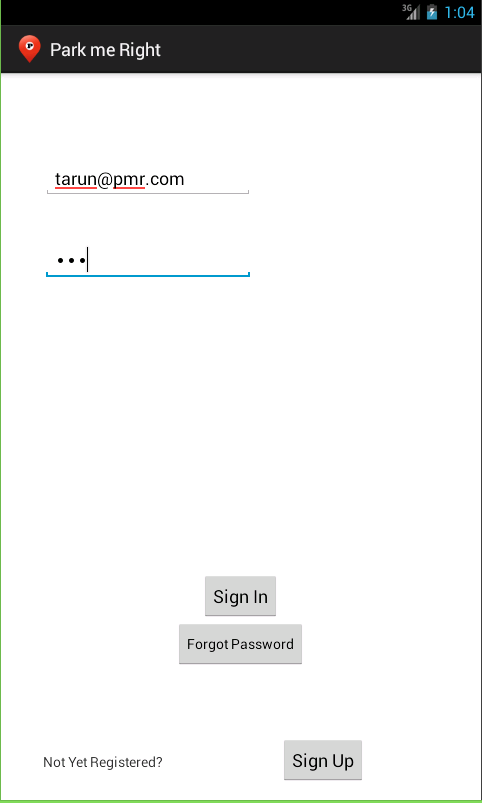
**Figure 7.16: Forgot password facility**

This page helps a user to reset his/her forgotten password. This facility provided by the application helps recover the lost or forgotten passwords by sending login credentials as an email to the registered users only. If the user happens to be not registered with the parkmeright application then no email is sent. An error message is displayed instead.

* + 1. **Android Application**

1. **Login Activity**

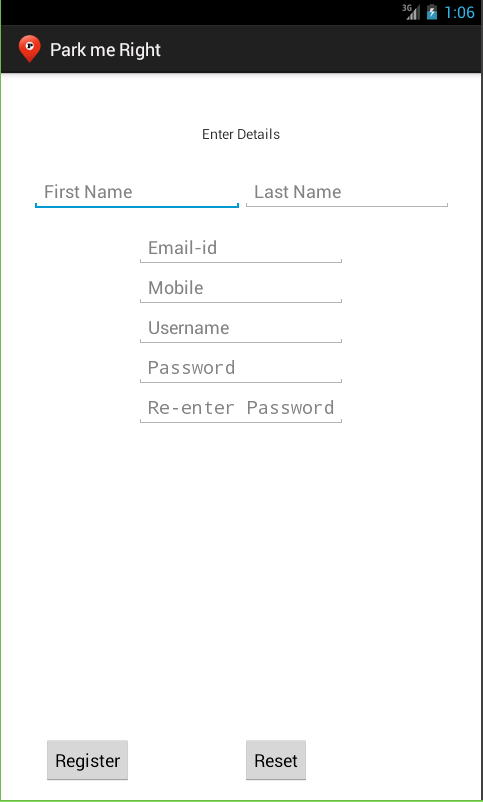
Any android application supporting multiple users needs to have a login page so as to differentiate, identify, create and manage sessions for the authorized users. Our application also has a login page to allow only registered users with the parkmeright application or a particular parking lot, to access the android application.



**Figure 7.17: Android App Login**

1. **Registration Activity:**

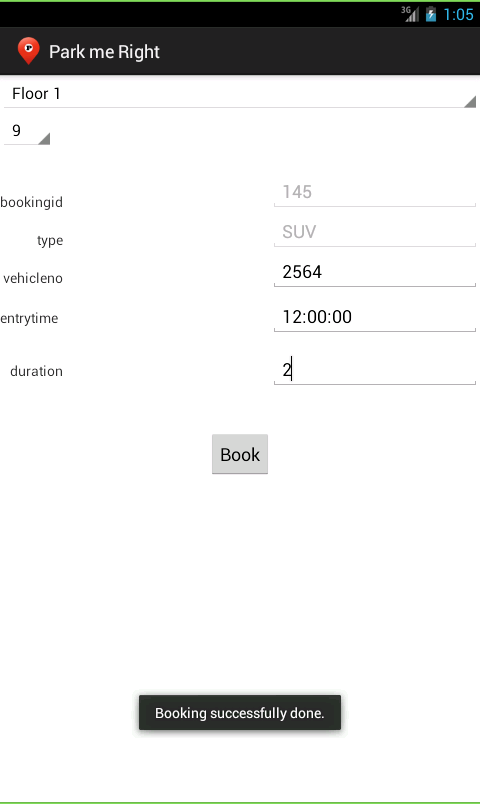
If the user wants to avail the facility of online slot booking using the android app, then they would have to register first and have their account created. For creating the account, they need to enter their details such as first name, last name, Email-ID, mobile number, Username and the password they want.

****

**Figure 7.18: Registration Page**

1. **Booking Activity:**

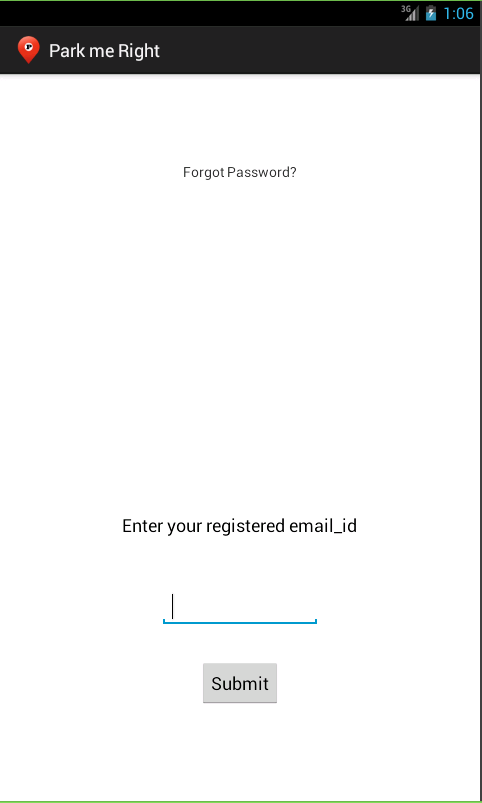
The users upon successful login would be able to interact with the administrator and can book a slot for themselves. The user has to select the floor on which they would like to park their vehicle. On selecting the floor, they would be able to see the available slots from which the user can select the one they want to. They would then just have to feed their vehicle number, time of entry and the duration of booking. On clicking the “book” button, their slot would get booked.

****

**Figure 7.19: Advance Booking of Slot**

1. **Forgot Password Activity:**

If in some case, the user forgets their password, they can ask for a new password. By entering their email id, the new password would be sent to their id.



**Figure 7.20: Forgot Password**