

**Dharmsinh Desai University, Nadiad**

Faculty of Technology, Department of Computer Engineering

B. Tech. CE Semester – V

Subject: (CE – 515) Advanced Technologies

Project Title:

**R&R Tours**

**(Tours & Travel Management System)**

By:

Rutvik Kathrecha CE059 (17CEUBS048)

Rutwik Kikani CE063 (17CEUOS068)

Guided by:

Prof. Prashant M. Jadav

Prof. Parth R. Dave

****

**Dharmsinh Desai University, Nadiad**

Faculty of Technology, Department of Computer Engineering

**CERTIFICATE**

This is to certify that the Advanced Technologies project entitled **“R&R Tours (Tours & Travel Management System)”** is a bonafied report of the work carried out by

**1) Rutvik Kathrecha, Roll No: CE059,** Student ID No: **17CEUBS048**

**2) Rutwik Kikani, Roll No: CE063,** Student ID No: **17CEUOS068**

of Department of Computer Engineering, Semester V, academic year 2019-20, under the guidance and supervision for the subject Advanced Technologies.

(Project Guide) (Project Guide) (HOD)

**Prof. Prashant M. Jadav Prof. Parth R. Dave Dr. C. K. Bhensdadia**

Associate Professor Assistant Professor Head of Department

Department of Computer Department of Computer Department of Computer

Engineering, Engineering, Engineering,

Faculty of Technology, Faculty of Technology, Faculty of Technology,

Dharmsinh Desai University, Dharmsinh Desai University Dharmsinh Desai University

Nadiad Nadiad Nadiad

|  |  |  |
| --- | --- | --- |
| **No.** | **Contents** | **Page no.** |
| 1 | Introduction | 1 |
| 2 | Software Requirement Specifications | 2 |
| 3 | Design | 4 |
| 4 | Implementation Detail | 10 |
| 5 | Testing | 11 |
| 6 | Screen-shots | 13 |
| 7 | Conclusion | 19 |
| 8 | Limitation and Future Extension | 20 |
| 9 | Bibliography | 21 |

# Abstract

Now a days there is everything on the internet. People usually want to travel along the world in the vacations. So, we have developed a web portal for the tours and travel company. User can come to the website and find the package for the destination where he/she wants to go. They will have different options for choosing packages for the particular place. So, user can go to the website and can see all the packages on just mouse click, in place of following the old way to call the travel agent and ask for the brochure of the tour.

# Introduction

The objective of the project is to develop a system that automates the process and activities of travel and tourism. The purpose of the system is to provide the more option for travelling places and also giving the basic facility.

User can come to the website and can find the tour packages of the place where he/she wants to go. User can select particular package and can see the details of the packages. Details of the package includes, places to be visited, no. of days, departure date, cost of the package etc. User can request a call back for further booking of the package. User can login to the system by providing email and password. If user doesn’t have account than he/she can create one by filling up the sign-up form.

**Technology used:**

* Cascading style sheet (CSS3) for styling the HTML pages.
* JavaScript for providing dynamic content for the HTML pages.
* Bootstrap 4 for styling the HTML pages using predefined classes of bootstrap.
* Angular 6 for component division of our website.
* Node.js for backend support.
* MongoDB as a database manager to save and fetch data from database using Node.js mongoose module.
* Express modules as a middleware.
* Typescript used in Angular 6 for accessing all the components of the same.

**Platform used:**

* Visual Studio Code

# Software Requirement Specifications

**1) End User:**

R1 - Manage Account

Description: In this, user can do basic CRUD operation on account like Create, Add, Update, Delete.

R1.1 Create account

Input: Firstname, Lastname, email, mobile no

Output: Account will be created and confirmation email will be sent to the email id given by user.

Exception: If any of the data is invalid or user is already existing then account will not be created.

R1.2 Update account

Input: User will have to enter the updated details.

Output: Updated details will be displayed.

R1.3 Delete account

Input: Click the delete account button.

Output: entry of the user will be removed from the database and appropriate message will be displayed.

R2 – Login

Description: By this user can login into the system.

Input: Email id and password.

Output: user will be logged in.

Exception: If credentials provided are wrong then user can’t login.

R3 – Request Callback

Description: User can request callback for the enquiry or booking of the package.

Input: Firstname, lastname, email, mobile no

Output: Request received successfully message will be displayed.

**2) Administrator:**

R1 - Manage Packages

Description: Admin can add the new packages or can delete some packages as required. Admin can also update information in existing packages.

R1.1 Add package

Input: Package details like package id, name, days, to date, from date, price

Output: the new package appears in appropriate category on portal.

R1.2 Update package

Input: details to be update

Output: package details will be updated.

R1.3 Delete package

Input: Package ID.

Output: Package will be deleted and appropriate message will be displayed.

R2 – Manage Users

Description: Admin can add user, show all users, find specific user and delete the user.

R1.1 Add User

Input: Firstname, lastname, email, mobile no

Output: user will be registered.

R1.2 – Show Users:

Input: Show users command

Output: All the users with their details will be displayed.

R1.3 – Find User:

Input: Email id

Output: details of the user will be displayed.

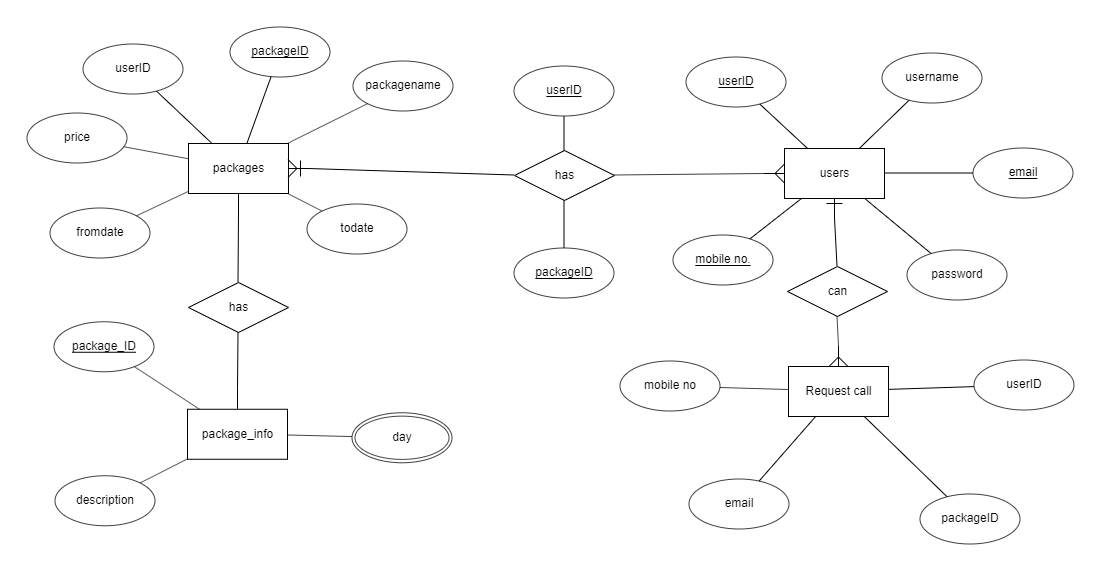
R1.4 – Delete User:

Input: Email id

Output: User will be deleted and appropriate message will be shown.

# Design

1. E-R Diagram



1. Data Dictionary

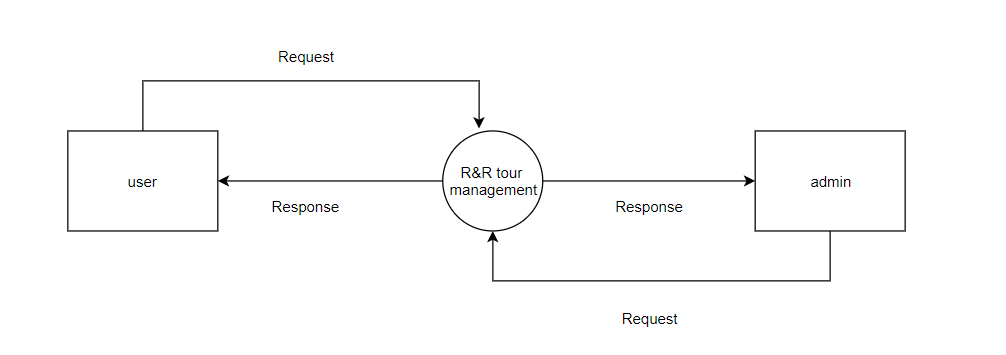
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Users | | | | | | | | |
| Sr.  No. | Field  Name | Data  Type | Width | Required | Unique | PK/FK | Referenced  Table | Description |
| 1 | \_Id | Varchar2 | 10 | Yes | Yes | PK |  | Unique ID provided by mongo db |
| 2 | First name | Varchar2 | 20 | Yes |  |  |  |  |
| 3 | Last name | Varchar2 | 20 | Yes |  |  |  |  |
| 4 | Password | Varchar2 | 10 | Yes |  |  |  |  |
| 5 | Mobile no. | Number | 10 | Yes | Yes |  |  |  |
| 6 | Email Id | Varchar2 | 15 | Yes | Yes |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Package | | | | | | | | |
| Sr.  No. | Field  Name | Data  Type | Width | Required | Unique | PK/FK | Referenced  Table | Description |
| 1 | PId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2 | Pname | Varchar2 | 10 | Yes |  |  |  |  |
| 3 | Days | Number Integer | 10 | Yes |  |  |  |  |
| 4 | to | Date | 10 | Yes |  |  |  | Check the value is not same as the 'from' attribute |
| 5 | From | Date | 15 | Yes |  |  |  | Check the value is not same as the 'to' attribute |
| 6 | Price | Number | 6 | Yes |  |  |  | Check the price for '-ve'  always '+ve' |

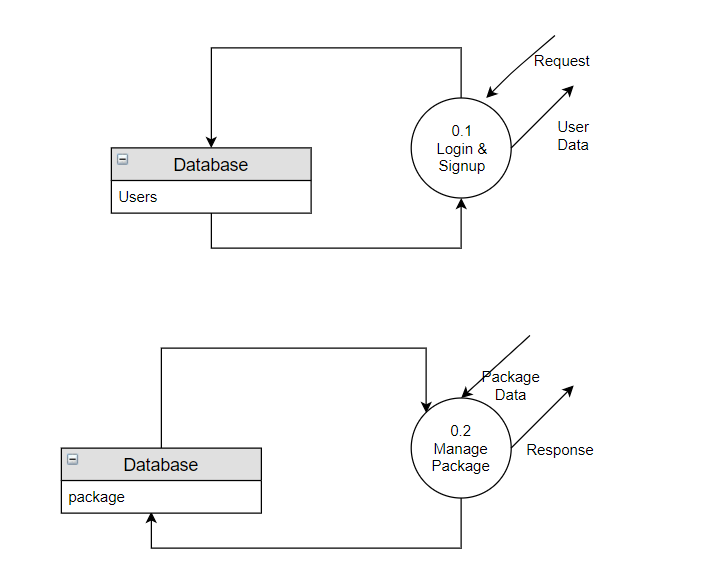
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Package\_info | | | | | | | | |
| Sr.  No. | Field  Name | Data  Type | Width | Required | Unique | PK/FK | Referenced  Table | Description |
| 1 | PId | Varchar2 | 10 | Yes | Yes | FK | Package |  |
| 2 | Day | Number | 10 | Yes |  |  |  |  |
| 3 | Description | Varchar2 | 25 | Yes |  |  |  |  |

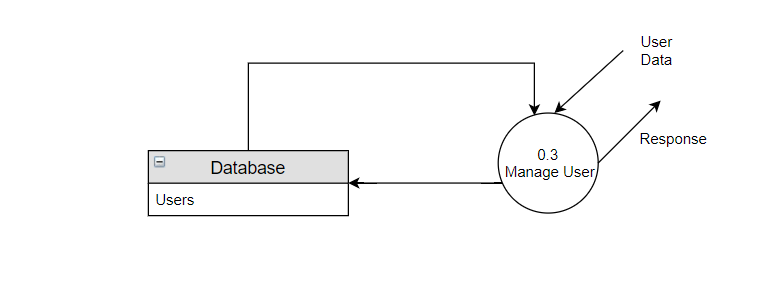
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requests | | | | | | | | |
| Sr.  No. | Field  Name | Data  Type | Width | Required | Unique | PK/FK | Referenced  Table | Description |
| 1 | First name | Varchar2 | 10 | Yes | Yes | FK | Package |  |
| 2 | Last name | Number Integer | 10 | Yes |  |  |  |  |
| 3 | email | Varchar2 | 25 | Yes |  |  |  |  |
| 4 | Mobile no | number |  |  |  |  |  |  |

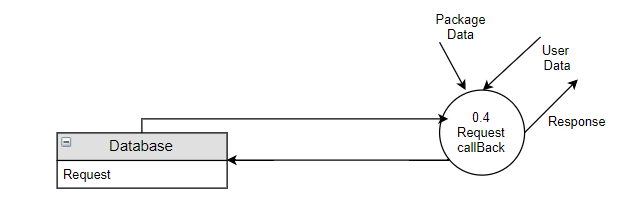
1. DFD Diagram



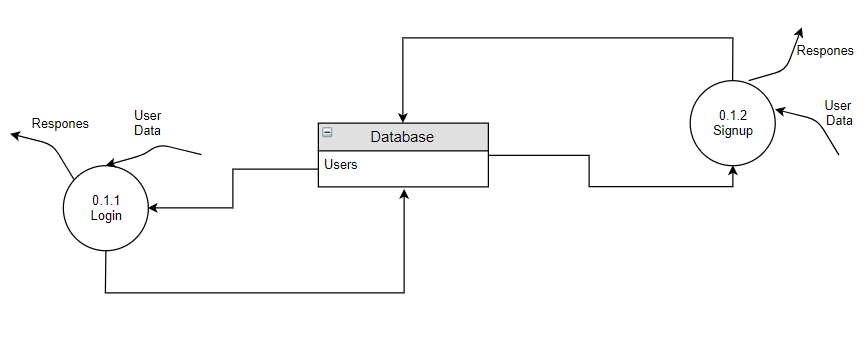
Level-0

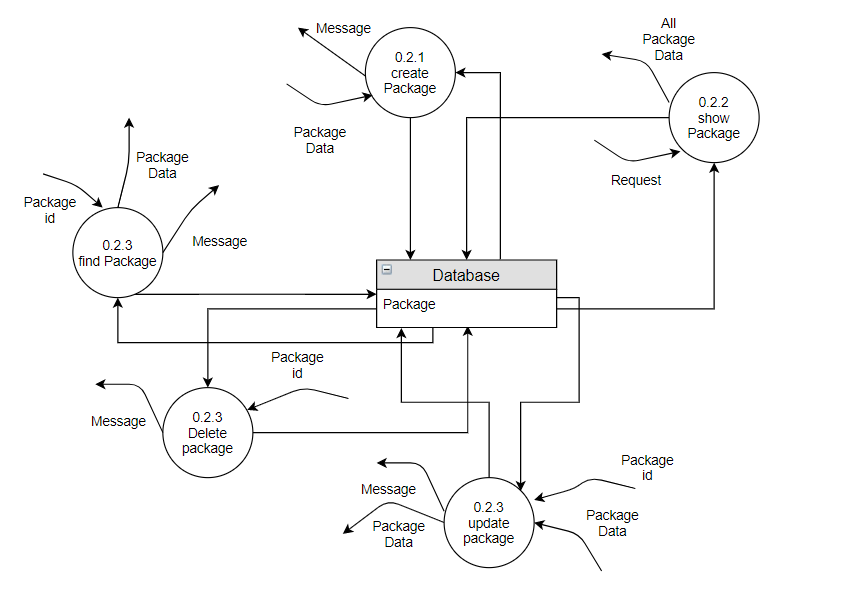


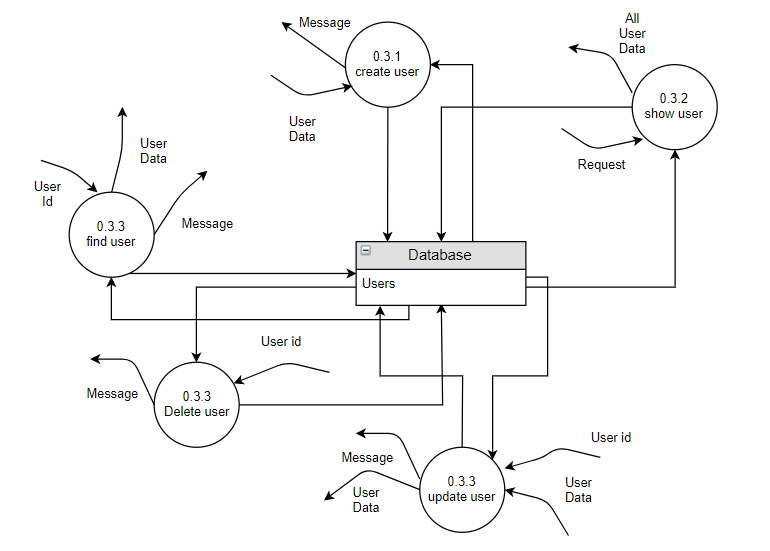




Level-1







Level-2

# Implementation Detail

Home Module:

In the home page first slide show is shown with beautiful pictures of worldwide places. At the middle of the page two sections is displayed, one for Indian tours and other for international tours.

Login/Signup Module:

In signup module user can register themselves by providing details like name, email, password and mobile no. User will be validated at time of registration that it is previously exists or not and the details provided by him is valid. After registering themselves user can login to the system by providing email and password. At the time of login user will be validate.

Package Module:

In the packages page user will be displayed various packages according to his/her selection. User can view different packages and can see the details and cost of each package.

Request Module:

User can request for the package inquiry by selecting request callback function and then submitting the necessary details.

Profile Module:

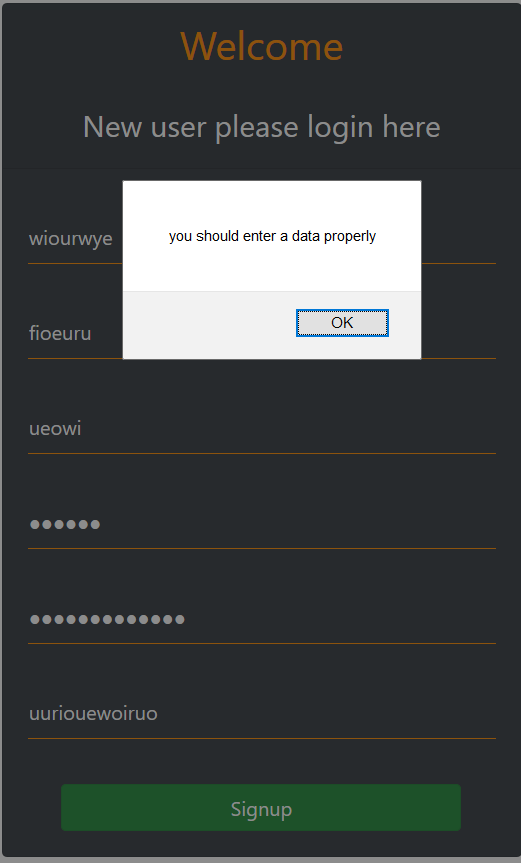
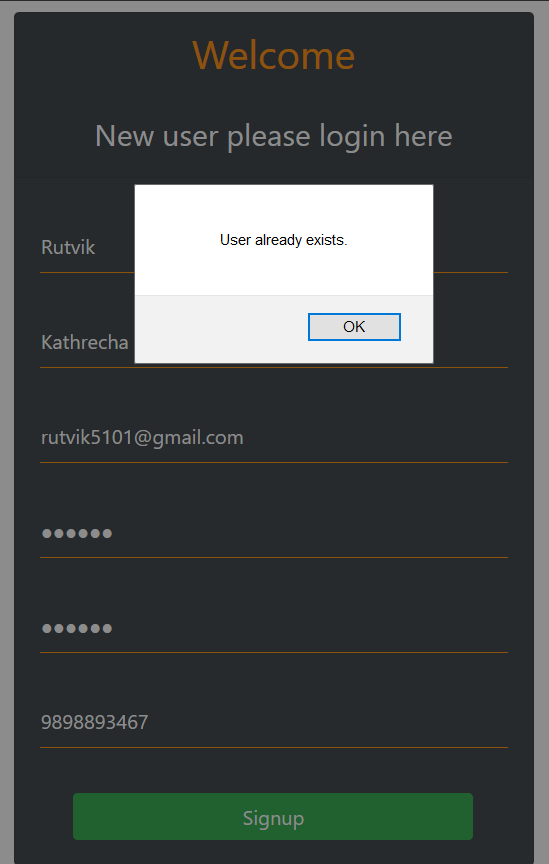
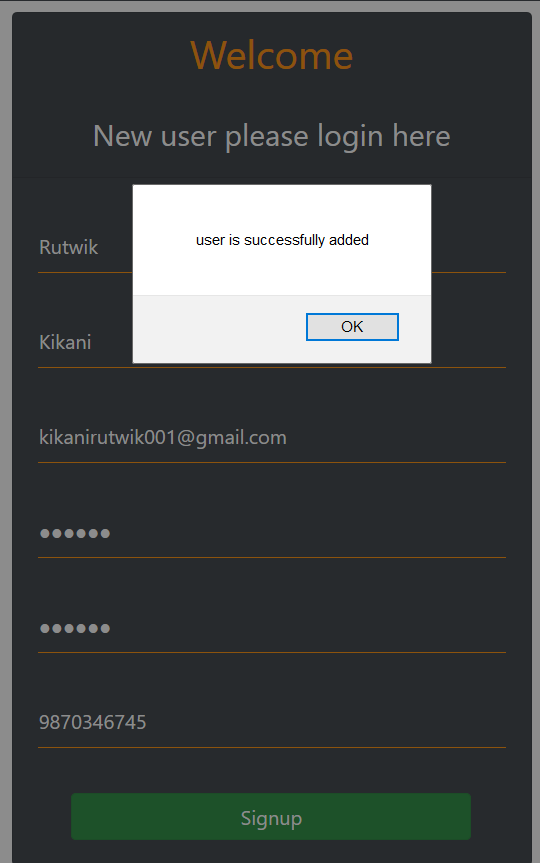
In the profile page user can see his/her profile and can update the details

Admin Module:

In the admin module admin can do the CRUD operation on the data. Admin can find user, view user, add user and delete the user.

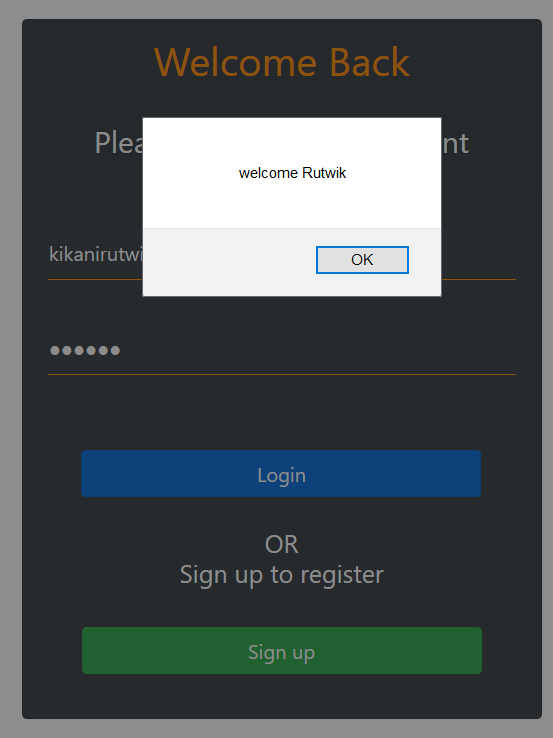
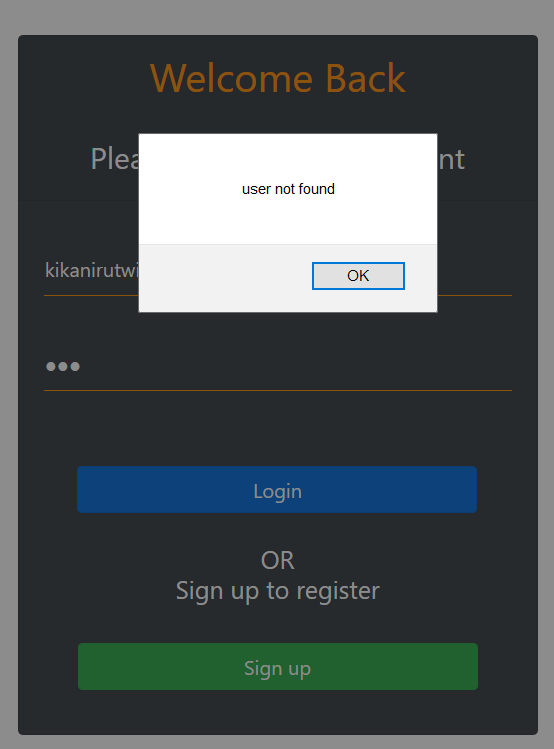
# Testing

For testing of login and signup forms we have used Angular inbuilt Model driven forms (Reactive forms). It is derived by importing the Reactive Forms Module.



Bad Input User exists Successful registration

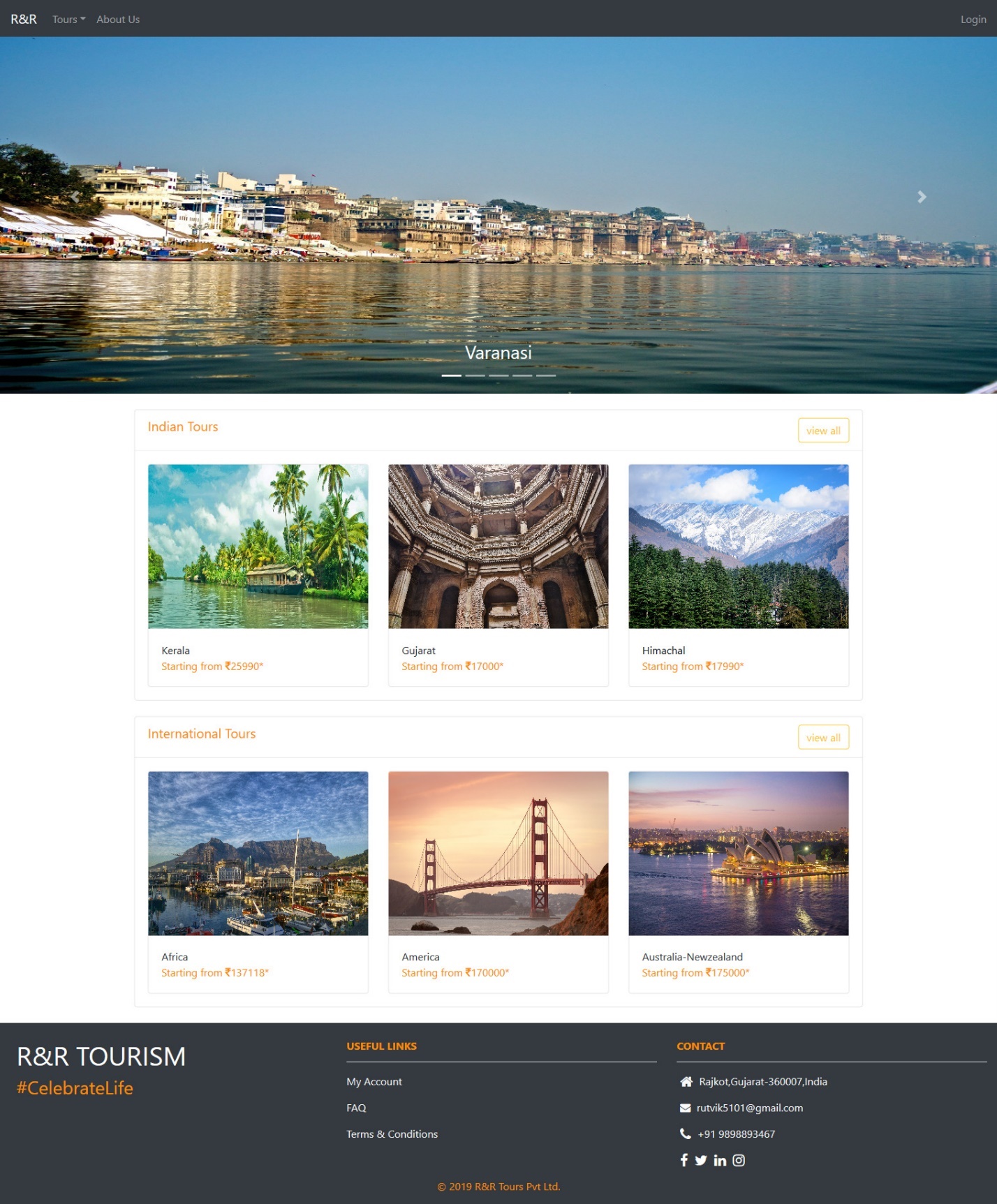
**Signup Testing**



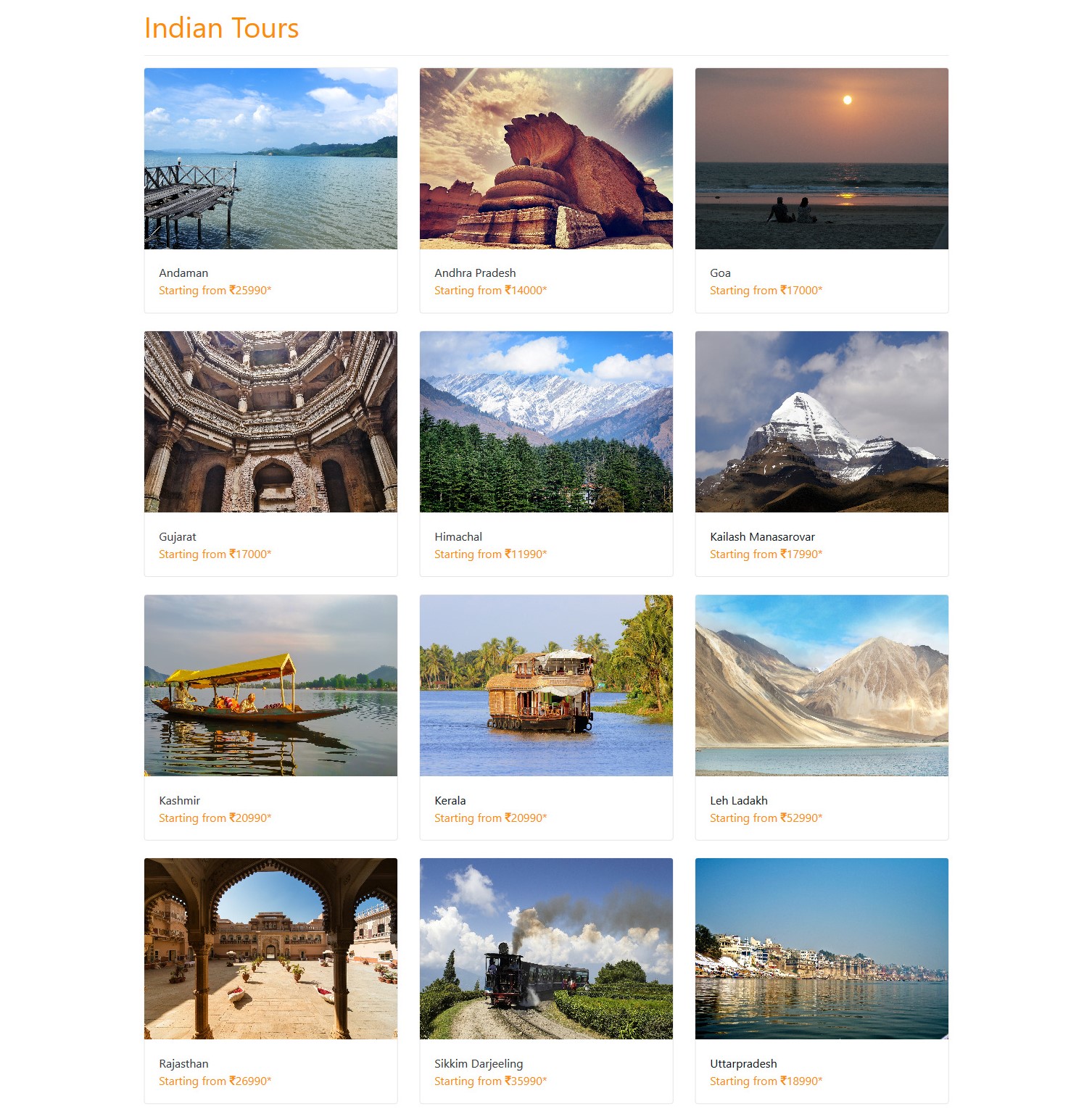
Invalid Input Successful login

**Login Testing**

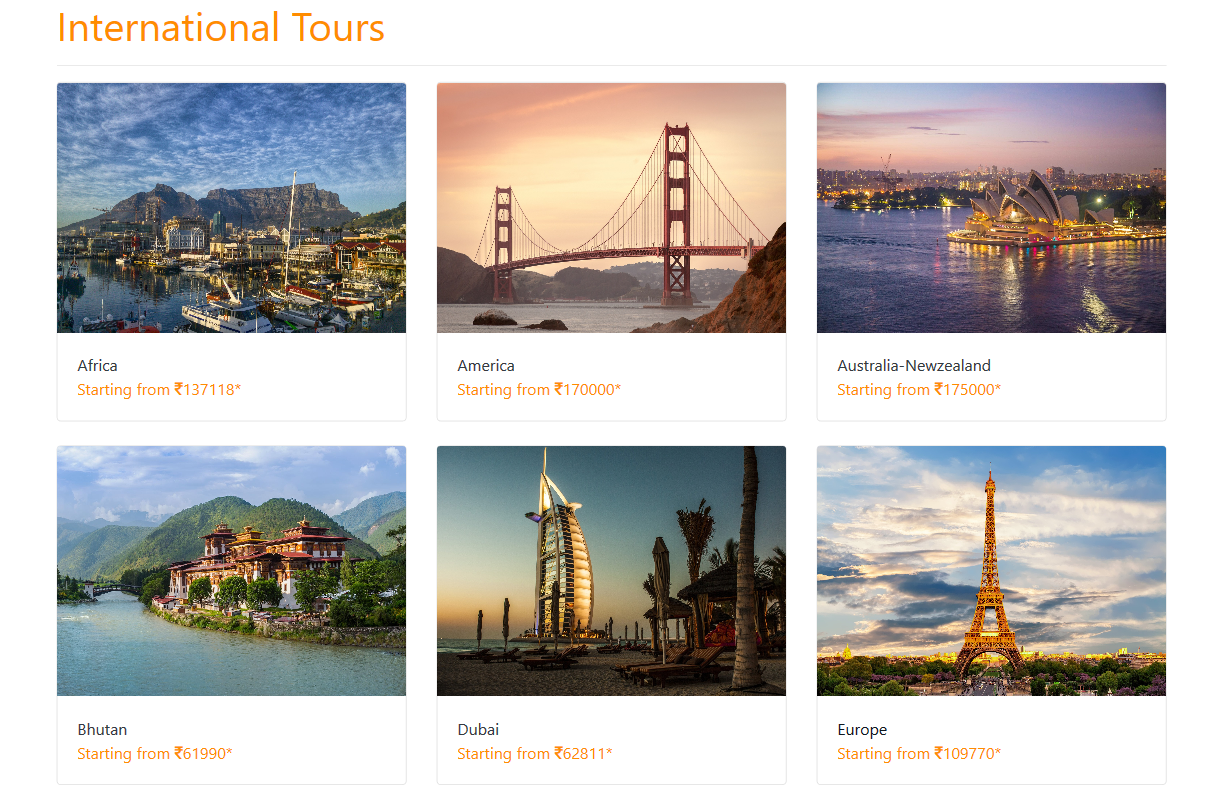
# Screen shots



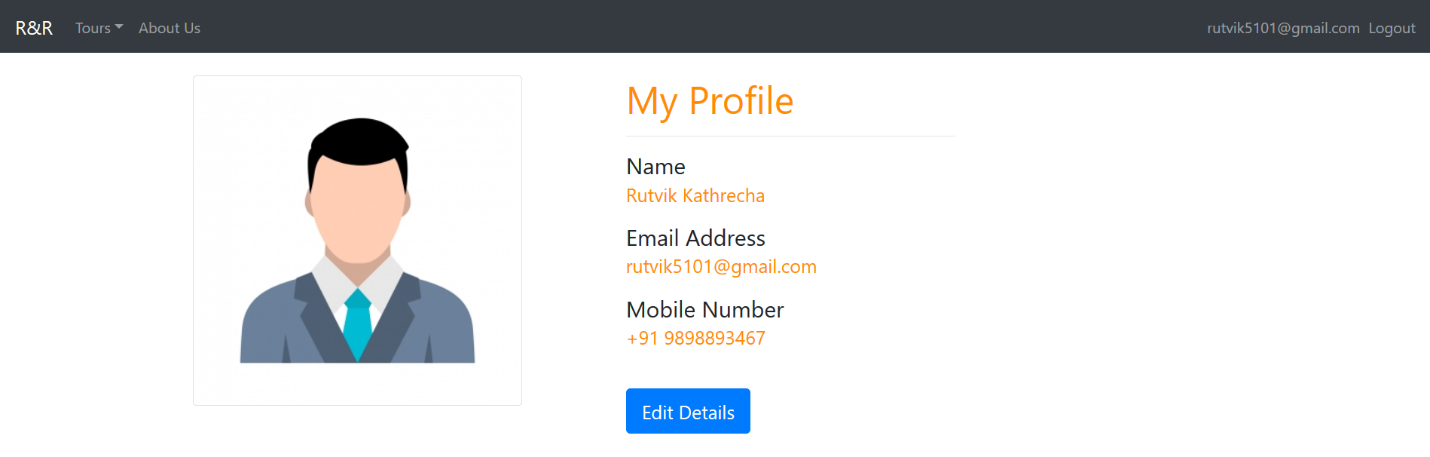
Home Page



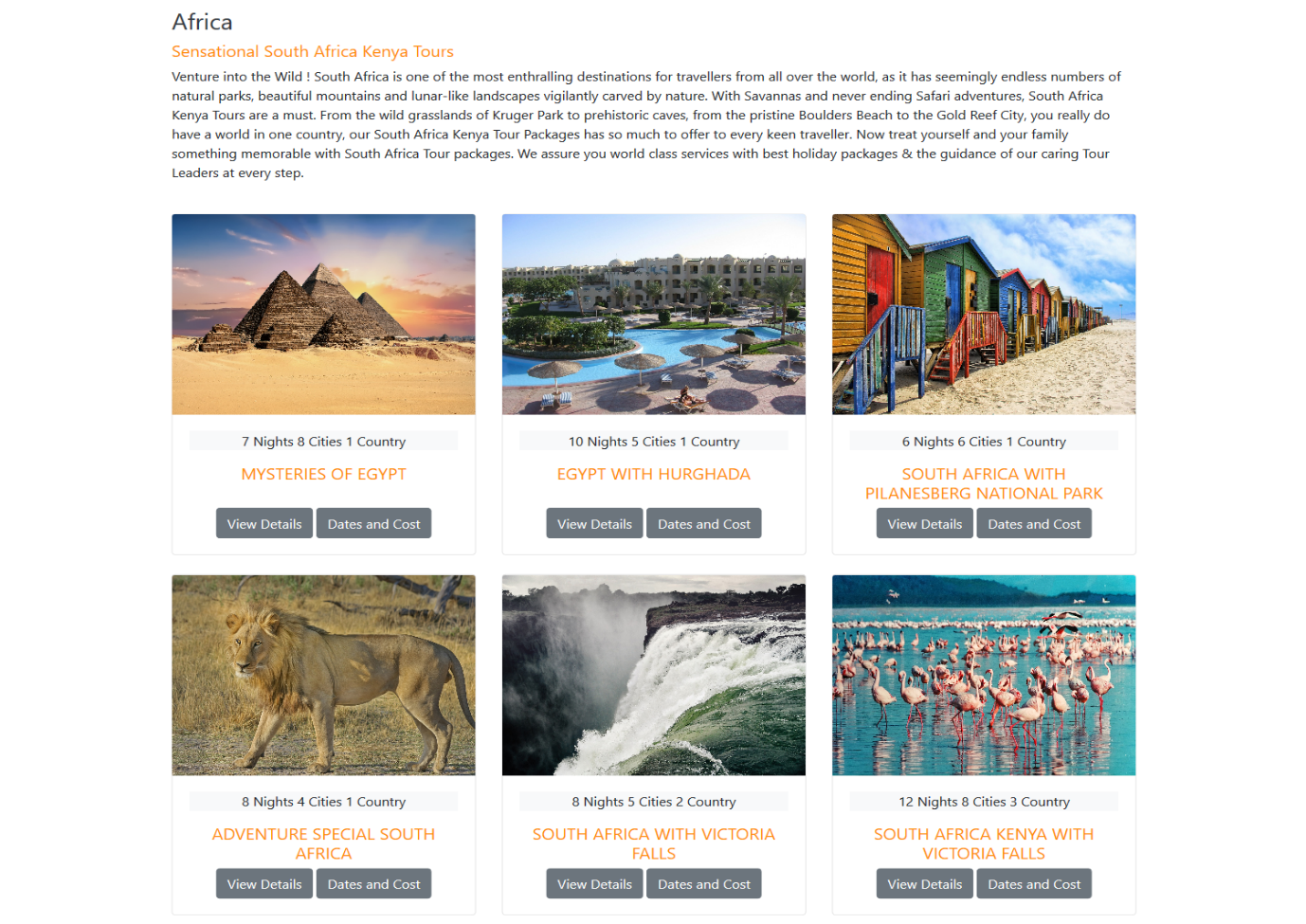
Indian Tours page



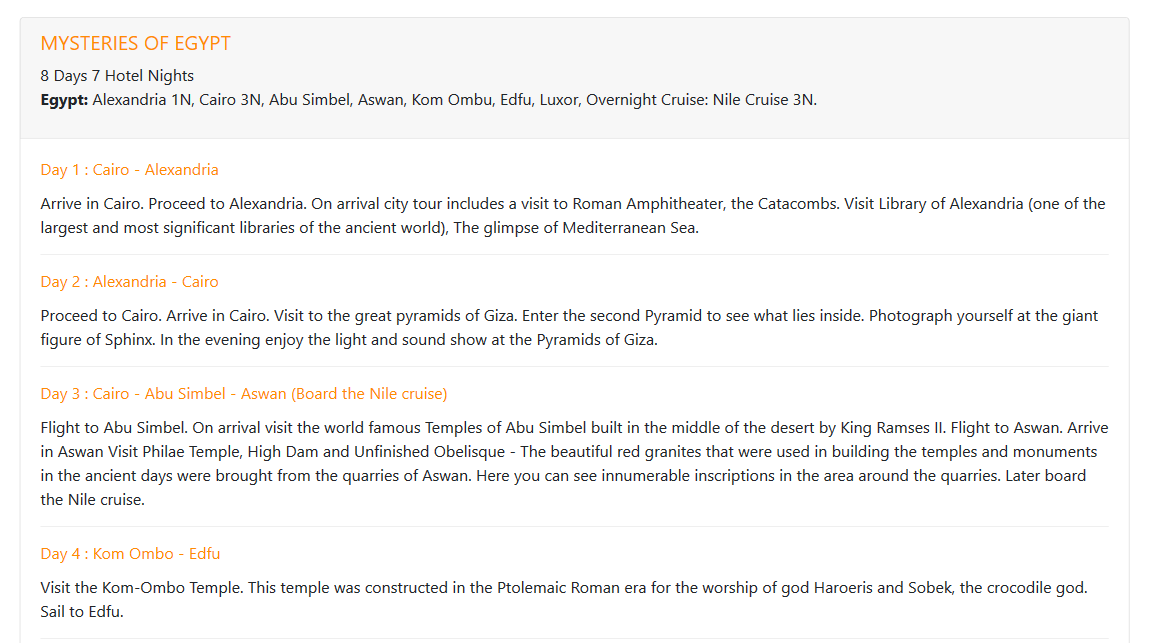
International Tours page



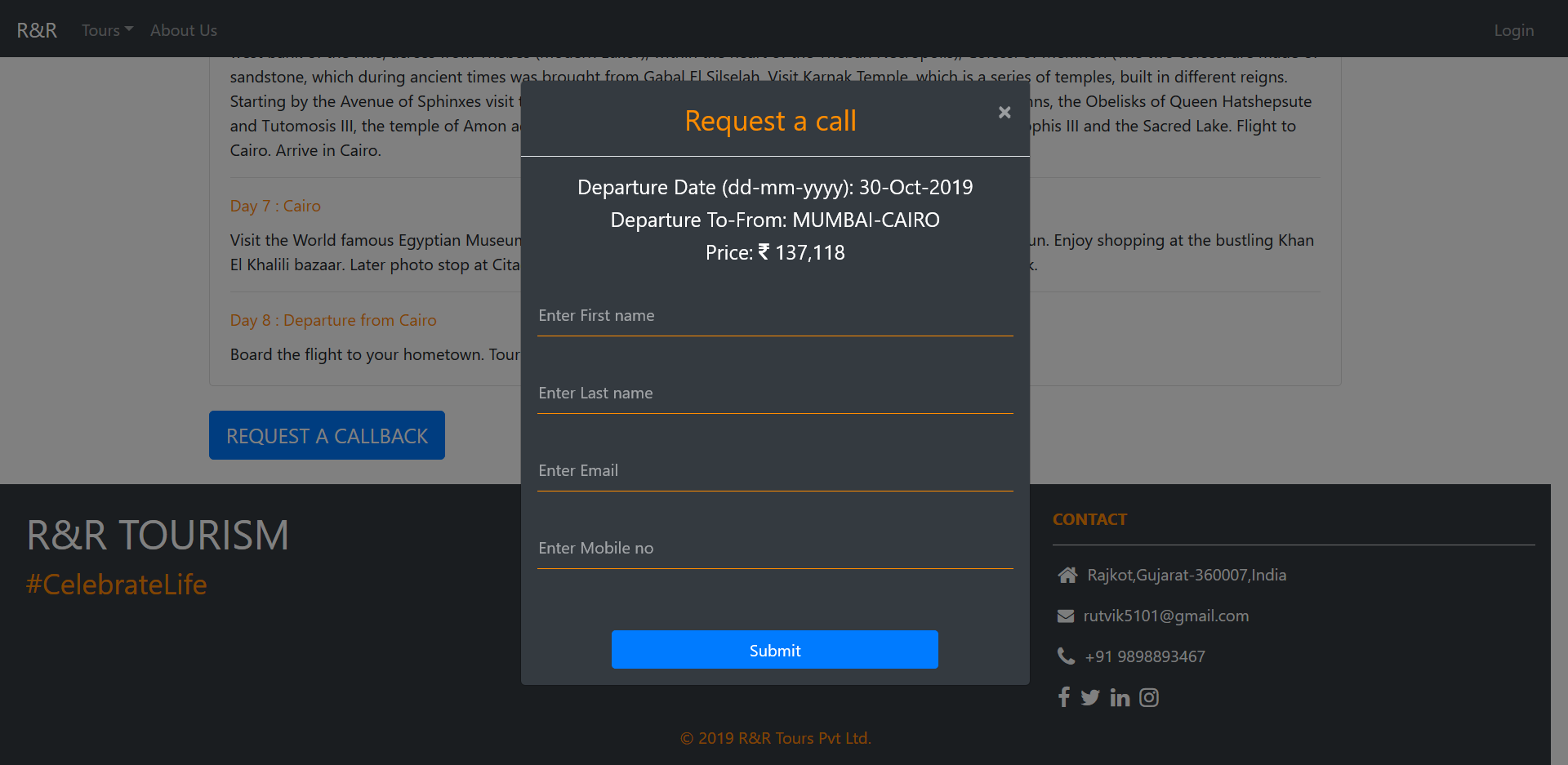
Profile Page



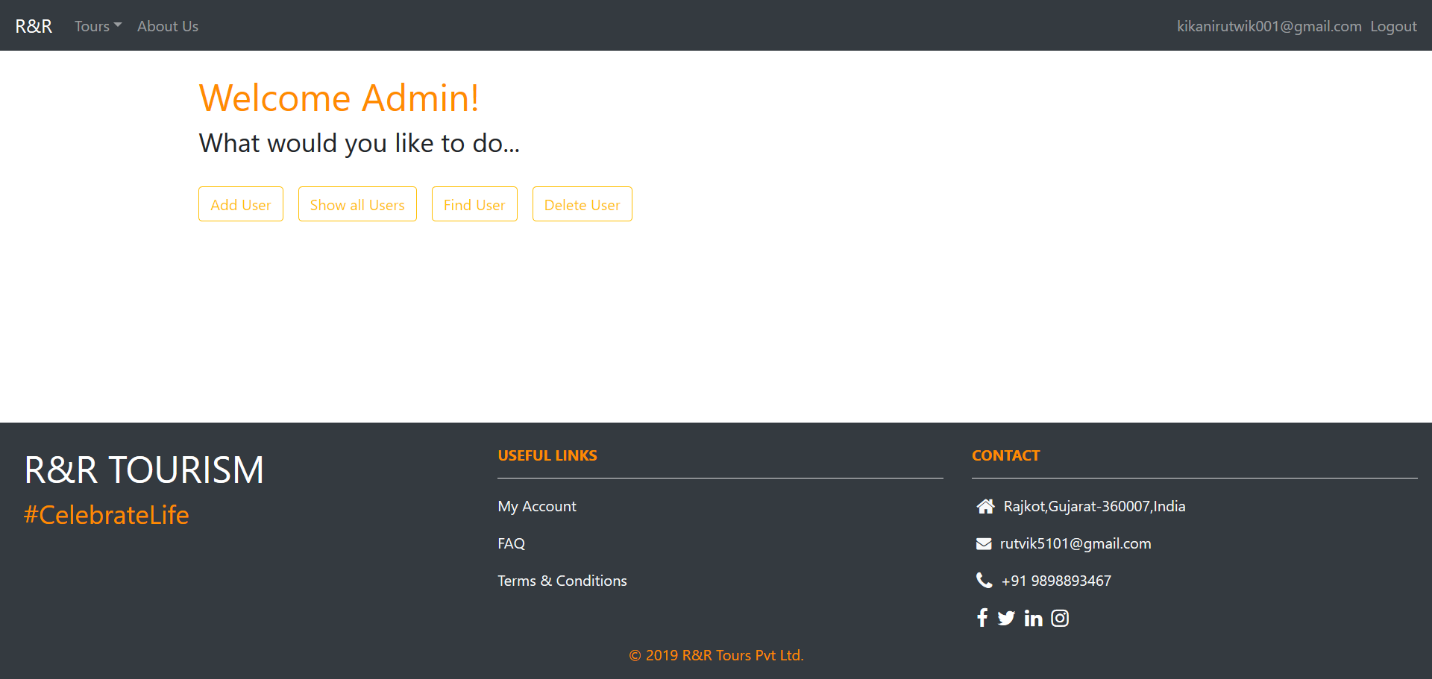
Africa Tours Package



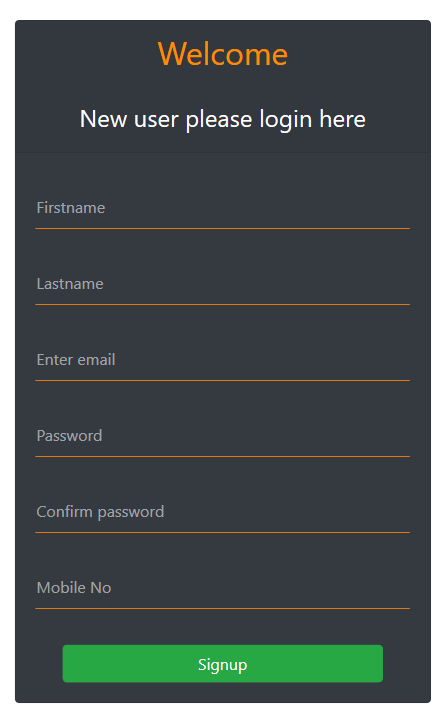
Tour Details

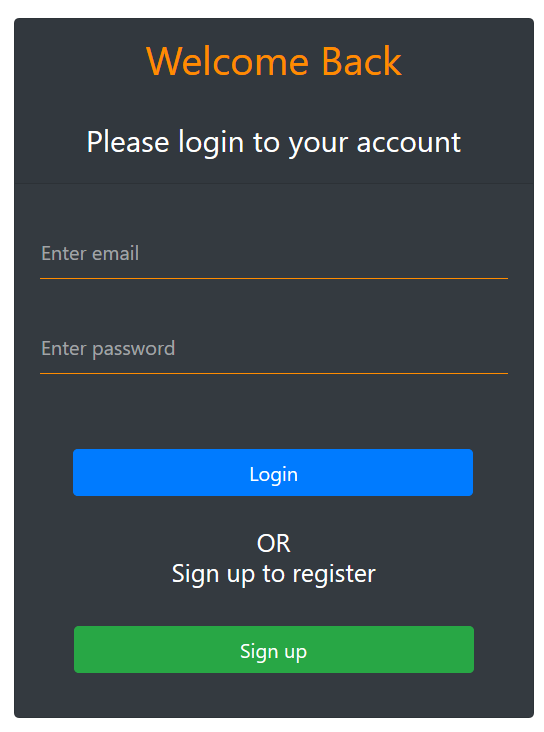


Request Callback model



Admin page





Login Page

Signup page

# Conclusion

In this project we have successfully implemented the signup and login functionality, which will check all the constraints occur during the process. Also, the designing and styling part was successfully implemented with the help of CSS, Bootstrap and JavaScript. Admin side is successfully implemented in which admin can add user, view all the registered users, find specific user and can delete the user. User update profile is also implemented successfully.

# Limitation and Future Extension

Limitations:

We have not implemented the booking functionality. Instead we have provided the request callback facility. Details of the packages are static. Profile page of the user is still remaining to implement fully.

Future Extension:

* All the package details will be provided dynamically.
* Booking functionality will be provided.
* Payment functionality will be implemented.
* User authentication will be done using OTP at time of registration (Email Verification).
* Feedback system will be implemented.

# Bibliography

<https://www.w3schools.com/html/default.asp>

<https://www.w3schools.com/css/default.asp>

<https://www.w3schools.com/bootstrap4/default.asp>

<https://getbootstrap.com/>

<https://fontawesome.com/>

<https://www.typescriptlang.org/>

<https://www.tutorialspoint.com/typescript/index.htm>

<https://angular.io/>

<https://nodejs.org/en/>

<https://www.mongodb.com/>

<https://expressjs.com/>