

WEATHERAPP

Group Information

Sampoorna swathi gorantla(sgpk9@mail.umkc.edu)

Kogila vani kannan(kkkz8@mail.umkc.edu)

Sirisha Valluri(valluri.sirisha03@gmail.com)

Faisal Hakami(fhhnrf@mail.umkc.edu)

I. INTRODUCTION

Weatherapp is an android application. It helps the user to find the route between the source and destination along with the flexibility to create the events or list the events on the specific date of travel

II. FRAMEWORK SPECIFICATION:

Our framework consists of three stages namely GUI, API, and Database. Database handles the storage of data like the user profile information, Events etc. The database used is Mongo DB, which is a cross platform document oriented database. Classified as a no SQL database.

GUI editor is developed as mobile application. The GUI connects to the database/API and gets the required data. The data fetched from the API's is modified and displayed in the user interface.

III. IMPORT EXISTING SERVICES/API

API to use Googlemaps and to fetch weather details:

<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

<http://api.wunderground.com/api/36b799dc821d5836/conditions/>

MongoDB API :

<https://api.mongolab.com/api/1/databases/aselab/collections//users?apiKey=gCXsqcTt-KNWvpuqEAGIdmkBAN5PhTJq>

UserStories:

As a system, I need to be able to create and retrieve the Events from Database and able to notify to the user in their homepage

As a user, I should be able to view the homepage of the application that contains buttons to create events and also, to modify and delete them.

As a user, I should be able to find the route between source and destination

As a user, I should be able to find the weather in the route.

As a user, I should be able to update or delete the events and even profile of the user.

As an admin, I need to store the details of the user in the database and also need to store the events, and notify them to the users.

WIRE FRAMES:

LOGIN PAGE:

Every user needs to have a user account in order to access the application. After clicking the Login button, the app validates the user details stored in mongo DB database and then if the user details exist it navigates to home page or throws error saying invalid credentials.

The wireframe shows a login page for 'Weatherapp'. It features a title 'Weatherapp' at the top, followed by the instruction 'Please Sign in'. Below this are two input fields: 'Email' and 'Password'. A 'Login' button is positioned below the password field. There is a checkbox labeled 'Remember me.' below the login button. At the bottom, there are two links: 'New to Weatherapp? Register Here!!!' and 'Forgot Password?? Access Here!!!'. The entire form is enclosed in a rounded rectangular border.

REGISTER PAGE:

User can register to the application by giving valid details like Email,password.

Weatherapp
Register

Enter Username

Enter Password

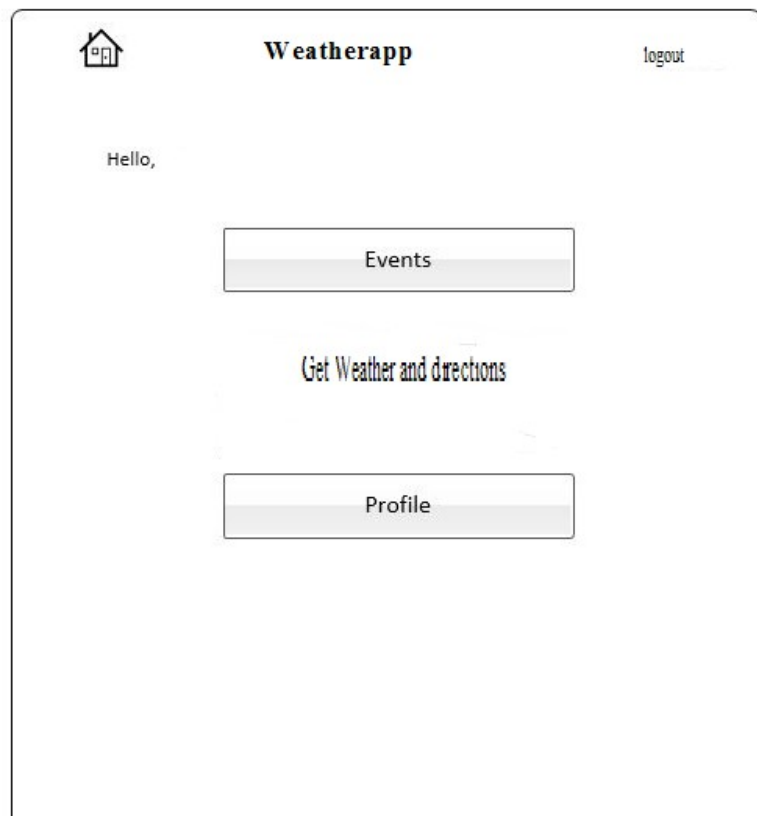
Enter Email

Register

HOME PAGE:

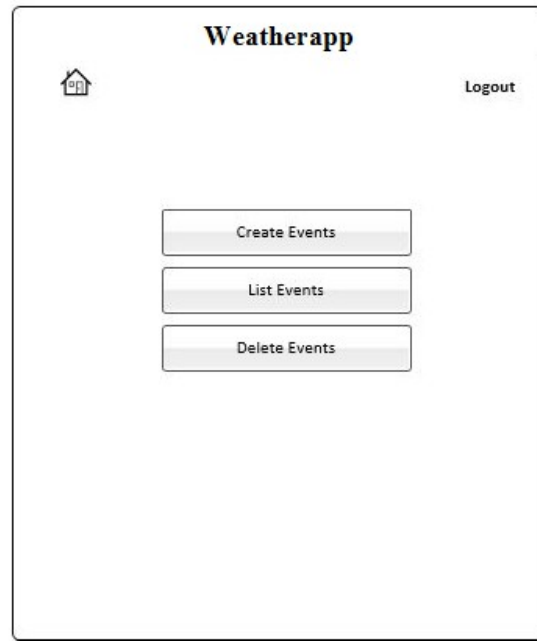
Home page consists of Events button, on clicking on which navigates to page which consists of buttons to create events, modify or delete them. Home page also consists of buttons to get weather and directions which loads directions and weather at a specified

place on clicking it.profile button helps the user to modify or delete his profile.



Events page:

User can create his desired event based on his travel schedules and can even list them and manage reminders to them. Reminder helps to get notifications to the user.



CREATE NEW EVENT PAGE:

User can create event by providing eventtype, eventname, event fromdate and todate as well as event description by setting reminder date and time.

Weatherapp

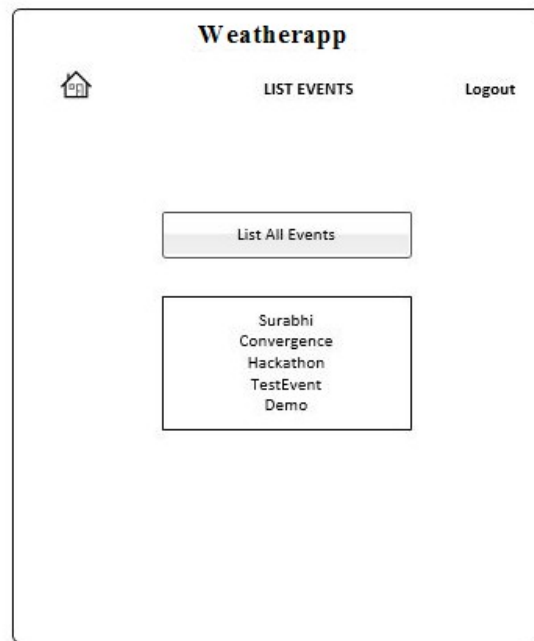
Create Event

Event Name	
Location	
Event Description	
Start Date	▼
End Date	▼

Create New Event

LIST EVENTS PAGE:

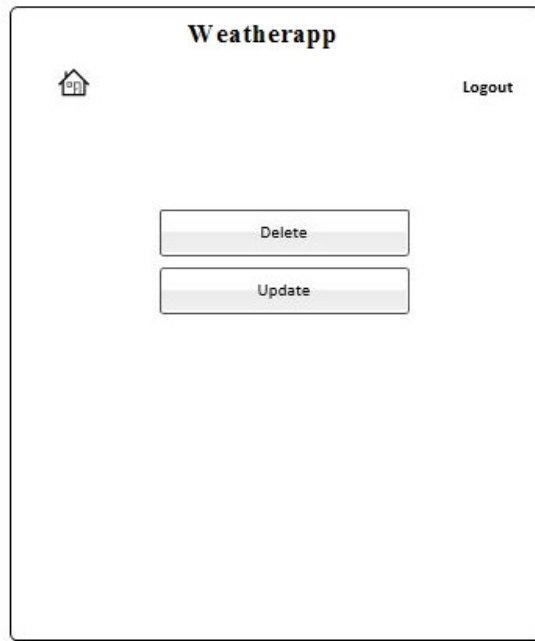
User can list down events which he had created on that particular date of travel.



Profile page:

User can update his information by clicking on the update button

User can delete his profile from the app if he wants, by clicking on the delete button.



MOCKUPS:

LOGINPAGE:

User needs to login to access the application.

Student Corner x

127.0.0.1:49793/www/index.html#/login

Weatherapp

Please sign in

Login

☐ Remember me.


New to weatherapp? [Register Here!!!](#)

Forgot password? [Access Here!!!](#)

Register:

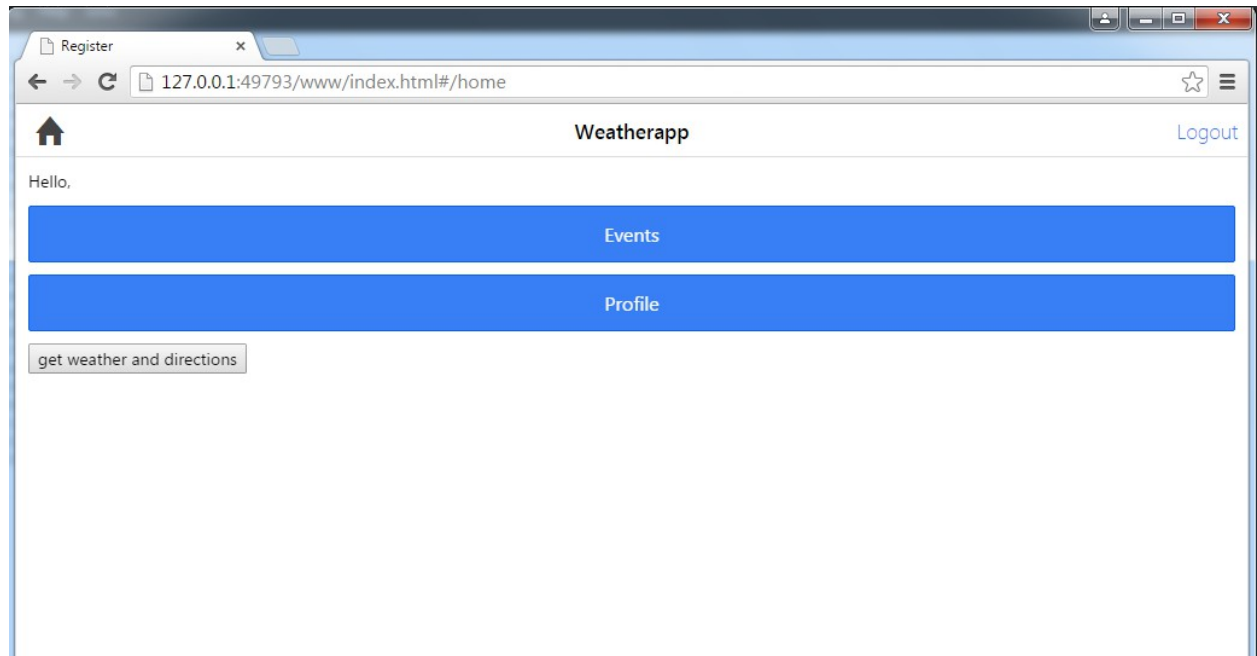
Register x

127.0.0.1:49793/www/index.html#/register

 Register [Logout](#)

Register

Home page:



To get weather and directions:

Mention the starting city and ending city to get weather and also directions

Home x

127.0.0.1:49793/www/templates/dir.html

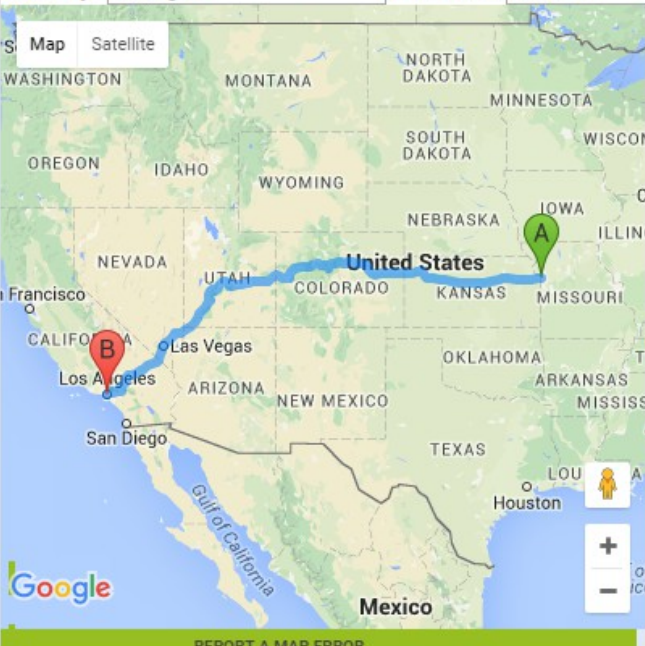
Logout

Hello

Start City: Start State:

End City: End State:

Map Satellite



United States

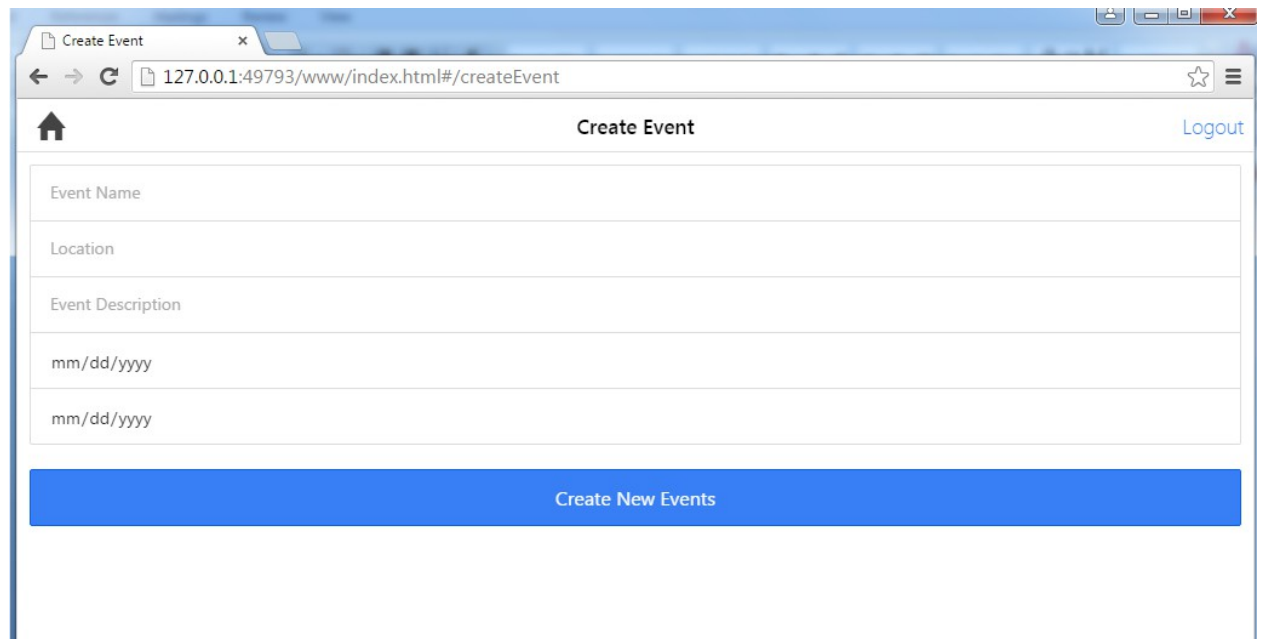
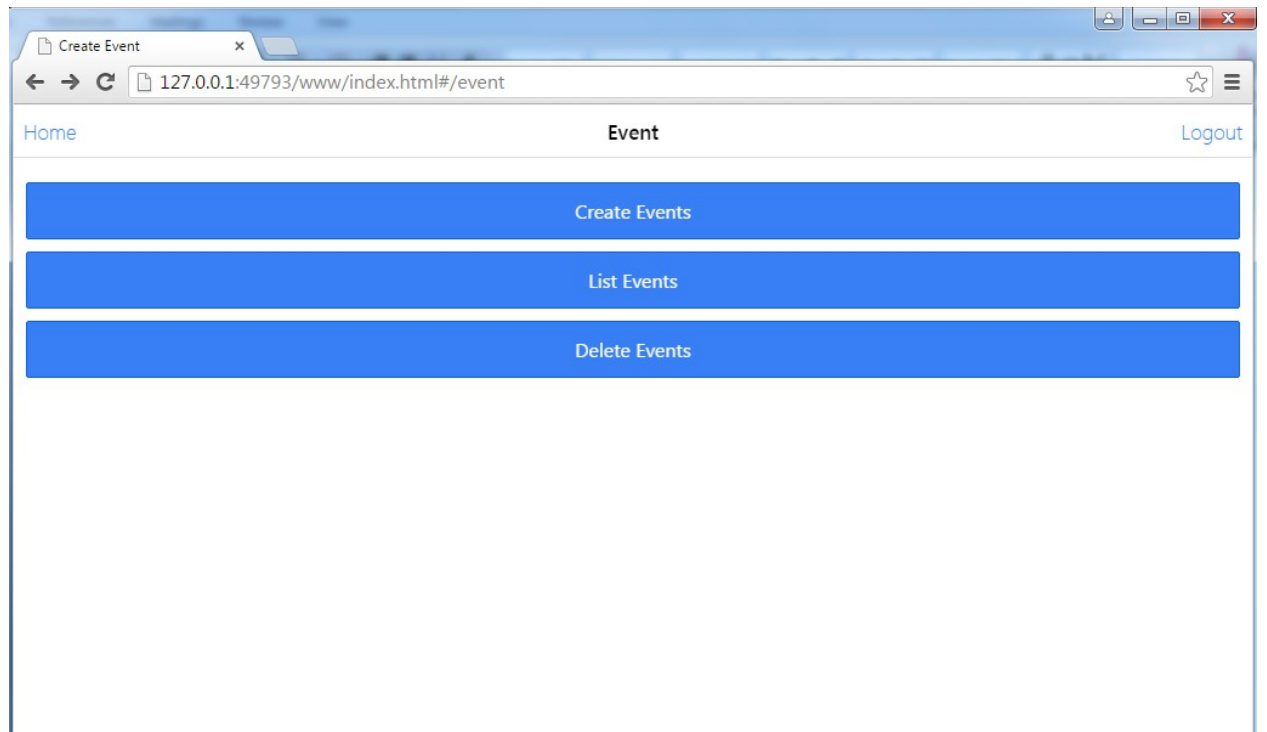
Currently 77 ° F and Partly Cloudy

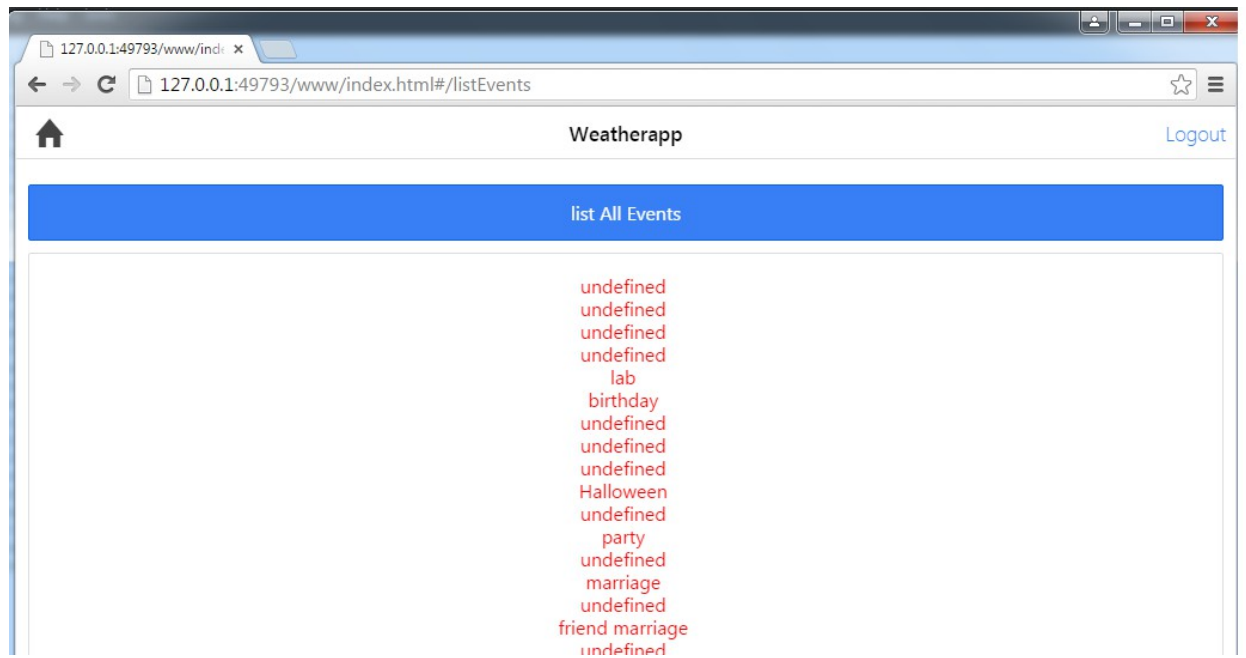
REPORT A MAP ERROR

Currently 57 ° F and Overcast

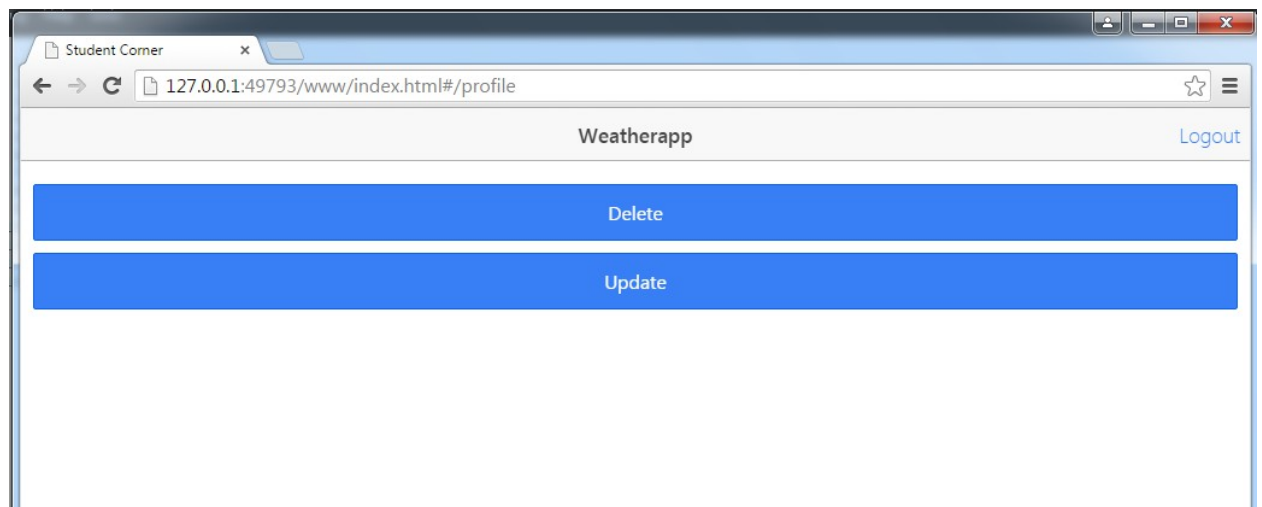
Events page:

To create, modify or delete the events

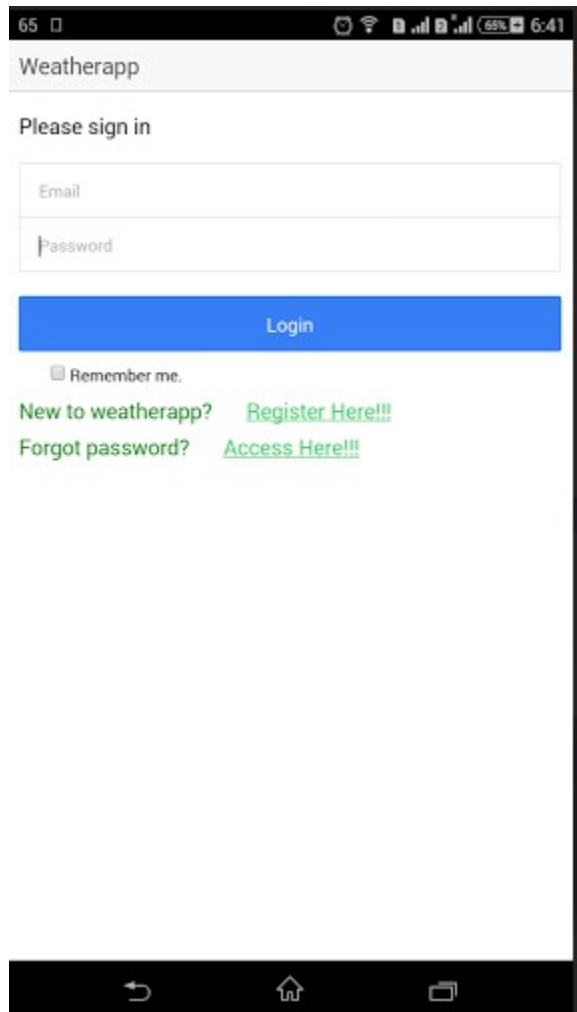




Profile page:



Deployment:



The screenshot shows a mobile application interface for 'Weatherapp'. At the top, there is a status bar with the number '65', a battery icon, and the time '6:41'. Below the status bar, the app title 'Weatherapp' is displayed in a light gray header. The main content area has a white background with the text 'Please sign in' in bold. Below this text are two input fields: 'Email' and 'Password'. A blue 'Login' button is positioned below the input fields. Under the button, there is a checkbox labeled 'Remember me.'. At the bottom of the form, there are two lines of text: 'New to weatherapp?' followed by a green link 'Register Here!!!', and 'Forgot password?' followed by a green link 'Access Here!!!'. The bottom of the screen features a black navigation bar with three white icons: a back arrow, a home icon, and a recent apps icon.

65 6:41

Weatherapp

Please sign in

Email

Password






Login


☐ Remember me.

New to weatherapp? [Register Here!!!](#)

Forgot password? [Access Here!!!](#)

65



 Register




Logout

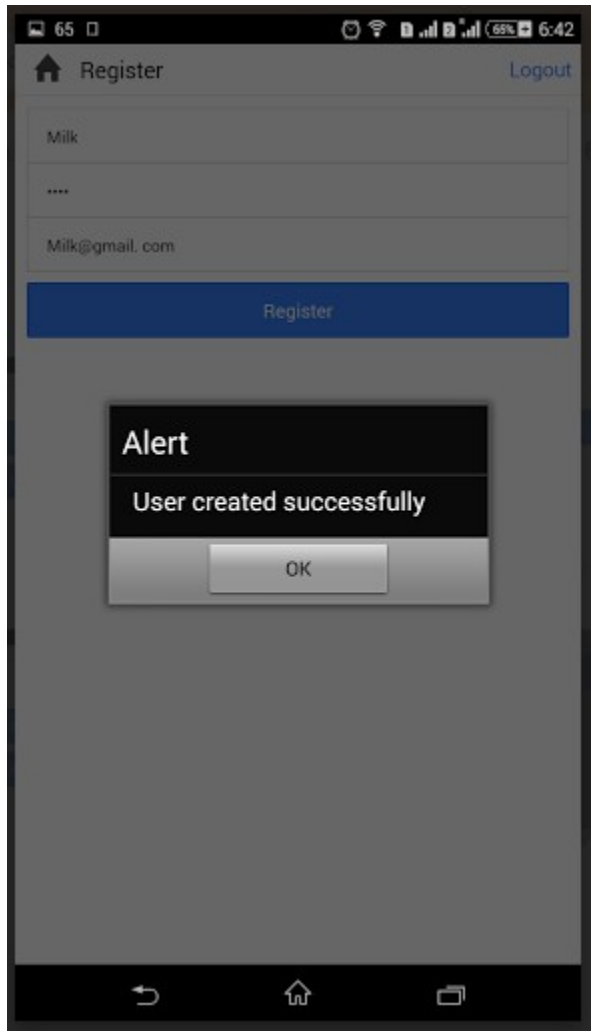
Enter Username

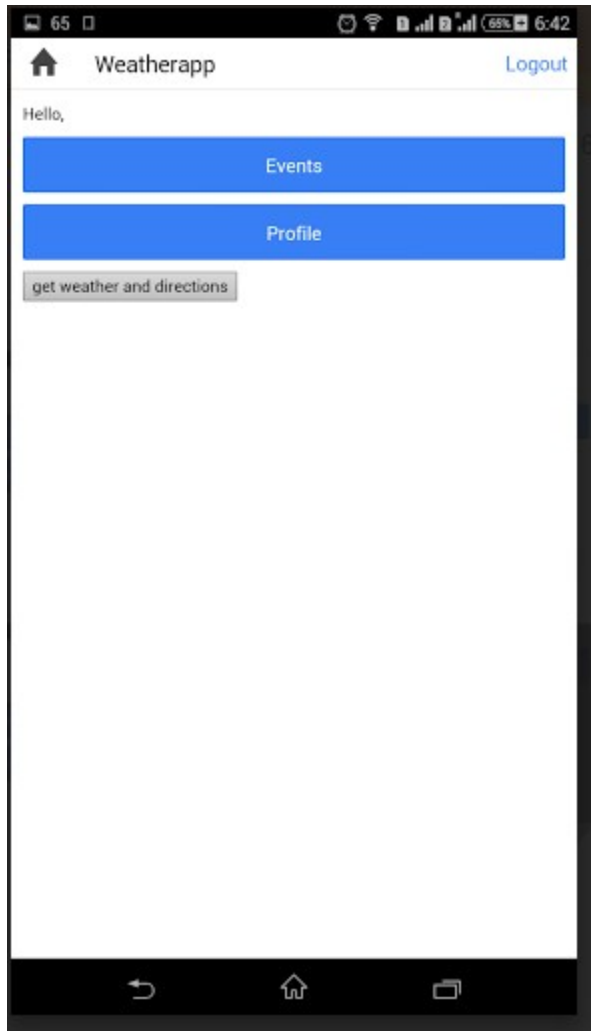
Enter Password

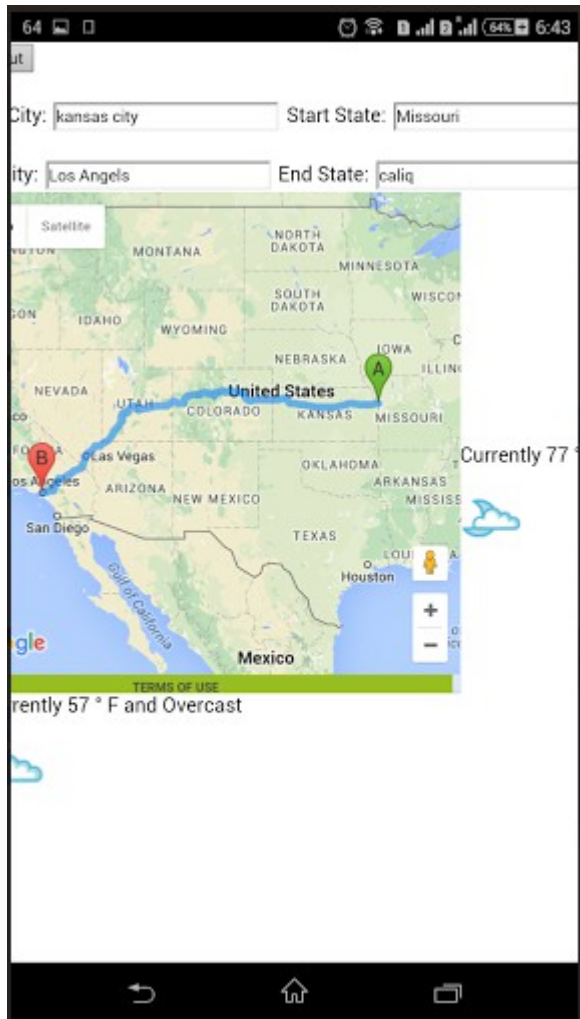
Enter Email

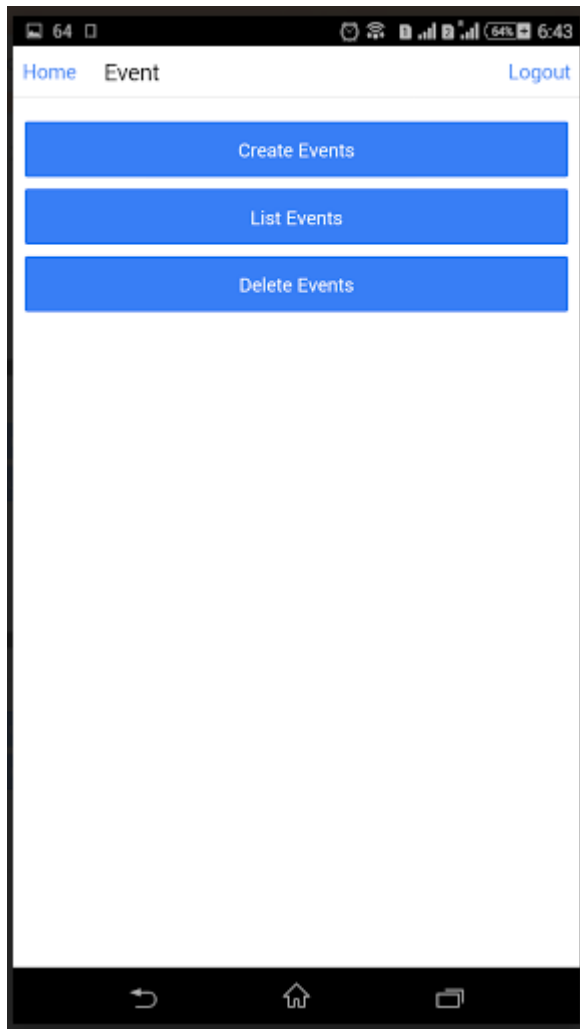
Register













 Saving screenshot...

 Create Event

[Logout](#)

Event Name

Location

Event Description

Create New Events

