

**Department of Computer Engineering**

**Faculty of Technology, Dharmsinh Desai University**

**College Road, Nadiad-387001**

**B.Tech. CE Semester – VI**

**Subject:** Software Design Practice

**Project Title:** Escape Plan – An Intelligent Tour Assistant

**Developed By**

Darji Dipesh – CE022, 17CEUBS137

Dhairya Acharya – CE026, 17CEUOS119

Viral Gajera – CE032, 17CEUOG0019

**Guided By**

Prof. Hariom A. Pandya

Department of Computer Engineering

Faculty of Technology,

Dharmsinh Desai University

****

**DHARMSINH DESAI UNIVERSITY**

CERTIFICATE

This is to certify that the project entitled “**Escape Plan – An Intelligent Tour Assistant**” is a bonafide report of the work carried out by

1) DARJI DIPESH, Roll No: CE022, Student ID No: 17CEUBS137

2) Dhairya Acharya, Roll No: CE026, Student ID No: 17CEUOS119

3) Viral Gajera, Roll No: CE032, Student ID No: 17CEUOG019

of Department of Computer Engineering, semester VI, under the guidance and supervision for the subject Software Design Practice. They were involved in Project training during academic year 2020-2021.

Prof. Hariom A. Pandya Dr. C.K. Bhensdadia

Department of Computer Engineering, Head, Department of Computer Engineering,

Faculty of Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Dharmsinh Desai University, Nadiad

Date:

**Contents**

1. Front Page...........................................................................1
2. Cretificate............................................................................2
3. Abstract...............................................................................4
4. Introduction........................................................................5
5. Software Requirement Specification...................................7
6. Design................................................................................10
7. Use Case Diagram.......................................10
8. Activity Diagram..........................................11
9. Data Dictionary………………………………………13

**5**. Implementation Detail.....................................................14

**6.** Screen-shots.....................................................................17

**7.** Conclusion........................................................................23

**8.** Limitation ........................................................................24

**9**. Future Extension……………………………………………………….. 25

10**.** Bibliography...................................................................26

**Abstract**

Our project, **Escape Plan – An** **Intelligent Tour Assistant**, is a Travel planner developed for those travelholics who love wandering and exploring new places around the globe. This is a special type of search engine which gives the users a recommendation on which are the popular places that he/she must visit at a particular destination.

It is basically a front end user interface to gather travel requirements from the user and present the proposed travel itineraries back to them, and a back end journey planning engine which performs the actual computation of the possible trip plans, prioritising these according to the user's necessities ( e.g. Number of days of the tour, number of people, Adventure filed trip for children or Pilgrims centric trip for old age people etc. )

**Introduction**

**Purpose**

This project aims to digitalize the whole aspect of trip planning, from choosing a destination to creation of itineraries. Atleast once in our lifetime, we come across this situation where we plan for months for a trip, like where to go, when, how do we go etc. and then that plans keeps on changing again and again until someone loses his/her mind. Travel Logistics can become a long, drawn-out horror story.

Thus, we come up with this idea ESCAPE PLAN, where whole process of planning and scheduling is taken care of by covering maximum famous places in and around the destination entered by the user. Hence, it is an innovative trip planning platform that enables user to plan trips across different cities of the world.

**Scope of the system**

The project targets all the travelholics who love exploring different places around the world. It is a boon for all those people and families who dream of travelling to a destination but doesn’t know other famous places around it. The main aspect of the project focuses on making the people independent from the travel agencies which demand a lot of money for the planning of a comfortable tour. There are many cases in which the travel agents cheat their clients and then the company takes no responsibility.

Hence, this project will make people self-reliable at the same time giving them the choice to plan a trip at their own pace and luxury.

* Technology Used
* Django Framework
* Python 3.6
* MongoDB, PyMongo, DJongo
* HTML, CSS, Bootstrap 4
* Javascript, JQuery
* Platform
* FourSquare API for finding the nearby places from a given pair of longitude and latitude.
* Google Maps API for showing the path between two places
* Google Places API
* Tools
* Visual Studio Code and Notepad++
* MongoDB Compass

**Software Requirement Specification**

* **Users of the System :**

1. People planning for a trip - TRAVELLERS

* **Functional Requirements :**

**R.1. Manage User Profile**

**Description:** User can register into the system. User can login into the system by means of user credentials. User can edit his/her account info.

**R.1.1: Login**

**Input:** Username and Password

**Output:** If user credentials are valid, then user is logged into the system, otherwise the error message is shown.

**R.1.2: Register**

**Input:** User Selection

**Output:** If all the necessary registration details are validated successfully against their corresponding criteria, then user gets register otherwise, error message is shown.

**R.1.3: Update Account Details**

**Input:** Modified Account information

**Output:** If the updated detail entered is valid against its

validation criteria, then the user details get updated, else

appropriate error message pops up.

**R.2. Generate Itinerary**

**Description:** User can generate itinerary by entering basic details of the trip and a day by day itinerary of that particular place is generated and also suggested places are displayed.

**Input:** City name, date of journey, distance, interest in that city

**Output:** day by day itinerary is generated and suggestion places are shown.

**R.3 Customize Itinerary**

**Description:** User can customize itinerary by adding or deleting the existing place in current itinerary.

**R.3.1: Add Place**

**Input:** User’s Selection

**Output:** Place added to the existing itinerary

**R.3.2: Delete Place**

**Input:** User’s Selection

**Output:** Place delete from the existing itinerary

**R.4 View places on map**

**Description:** User can view places to visit, on map day by day as per itinerary generated.

**Input:** User’s Selection

**Output:** Places on map are shown by pin.

**R.5 Save itinerary**

**Description:** Authenticated users can save the generated itinerary and use that for their future reference.

**Input:** User’s Selection

**Output:** Itinerary is saved in user’s account.

**R.6 View itineraries**

**Description:** Authenticated users can view past, current and upcoming Itineraries that he/she has saved.

**Input:** User’s Selection

**Output:** As per user’s selection saved itinerary is shown.

**R.7 Show Path between two places on Map**

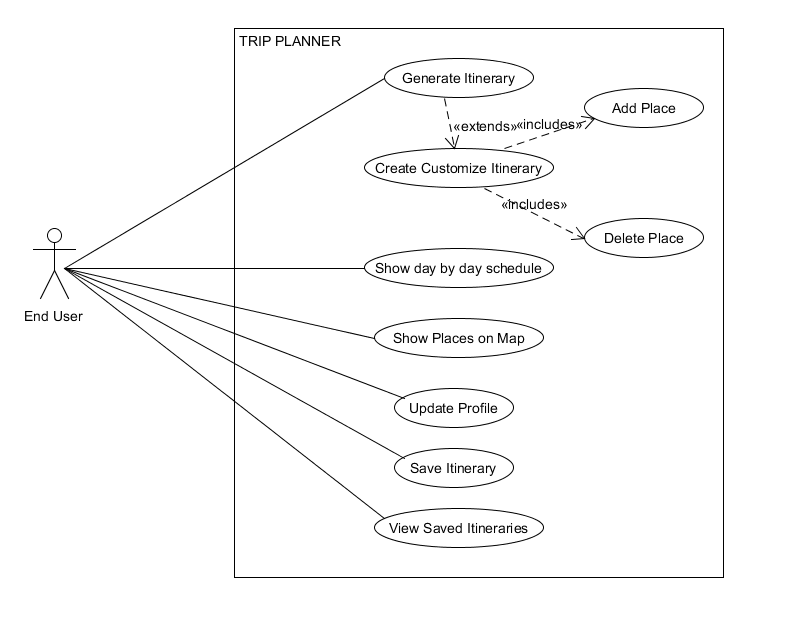
**Description:** Once the itinerary is generated, the user can see the road path between any two places in the itinerary to estimate the distance and time between two places.

**Input:** User’s Selection

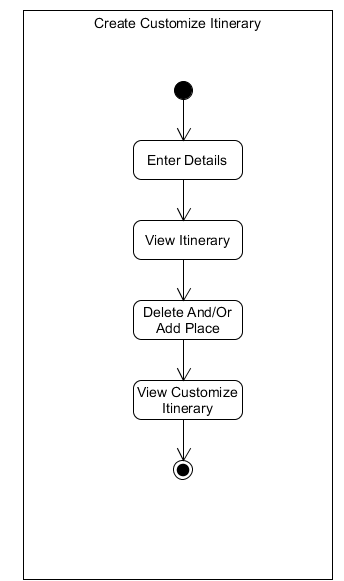
**Output:** Path is shown in the Map along with distance and Journey time

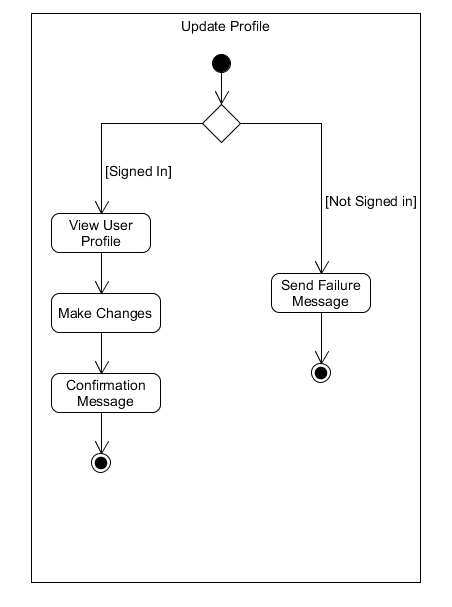
**Design**

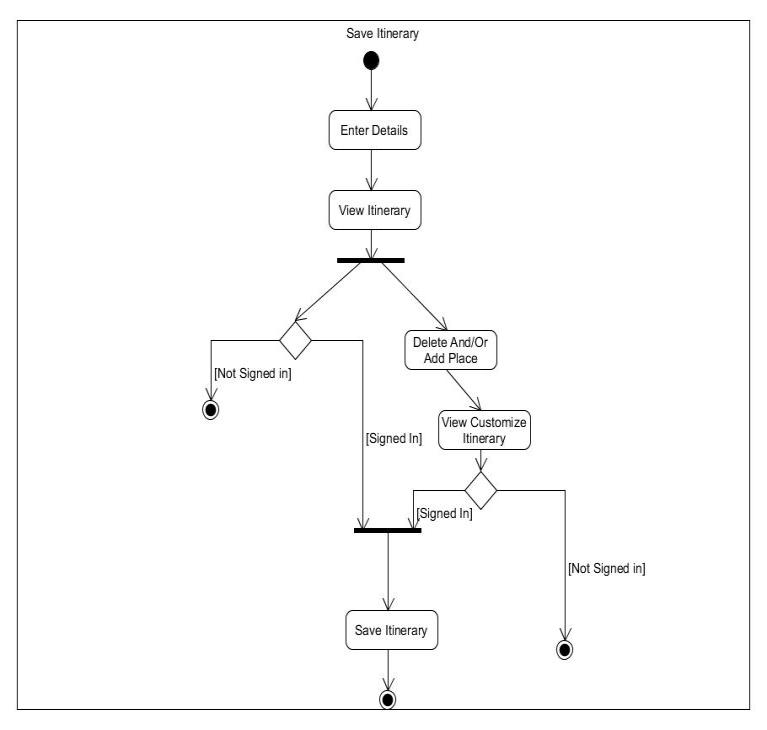
1. **Use Case Diagram**



1. **Activity Diagram**

****

****

****

1. **Data Dictionary**

| **1.User Details** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Username | Varchar | 15 | Yes | Yes | PK |  |  |
| 2. | First Name | Varchar | 20 | Yes | No |  |  |  |
| 3. | Last Name | Varchar | 18 | Yes | No |  |  |  |
| 4. | Email | Email | 50 | Yes | Yes |  |  |  |
| 5. | Password | Varchar | 20 | Yes | No |  |  |  |
| 6. | Mobile No | Number | 10 | Yes | Yes |  |  |  |

| **2.Trip Details** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Username | Varchar | 15 | Yes | Yes |  |  |  |
| 2. | City | Varchar | 30 | Yes | No |  |  |  |
| 3. | Category | Varchar | 10 | Yes | No |  |  |  |
| 4. | Start Date | Datetime | - | Yes | No |  |  |  |
| 5. | End Date | Datetime | - | Yes | No |  |  |  |
| 6. | Pack Days | Number | 2 | Yes | No |  |  |  |
| 7. | Radius | Number | 10 | Yes | No |  |  |  |

**Implementation Detail**

**Project Modules:**

* **End User Modules**
* **User Account Management Module**: User must provide all the necessary details and register himself/herself into the system provided that all details are valid against validation criteria. Using these credentials he must login into the system to further use its functionalities.

Once authenticated, the user can update the details if he has done a mistake in entering the credentials. This can be done by navigating to the View Profile page and changing the details. We are currently using MongoDB technology to authenticate and register users.

* **Generate Itinerary Module**: Whenever user wishes to schedule a trip, he/she uses this module. By entering the destination and journey dates, an itinerary is generated based on his/her preferences of places.

This is one of the main modules of our project. We have used Travelling Salesman Problem Algorithm among the most popular places we got using the FourSquare API. The list of all the places we get, is divided into the parts depending on number of days, and then we apply TSP algorithm to get the most optimized itinerary. The TSP algorithm needs to be used we need to cover all the places such that all nearby places are visited together.So that the user doesn’t have to come to the same place the next day. Hence TSP algorithm gives us the most optimized itinerary.

Apart from that, we fetch the details of the places using the FourSquare API like address, place type( garden, worship etc. ), images, longitude and latitude, and so on.

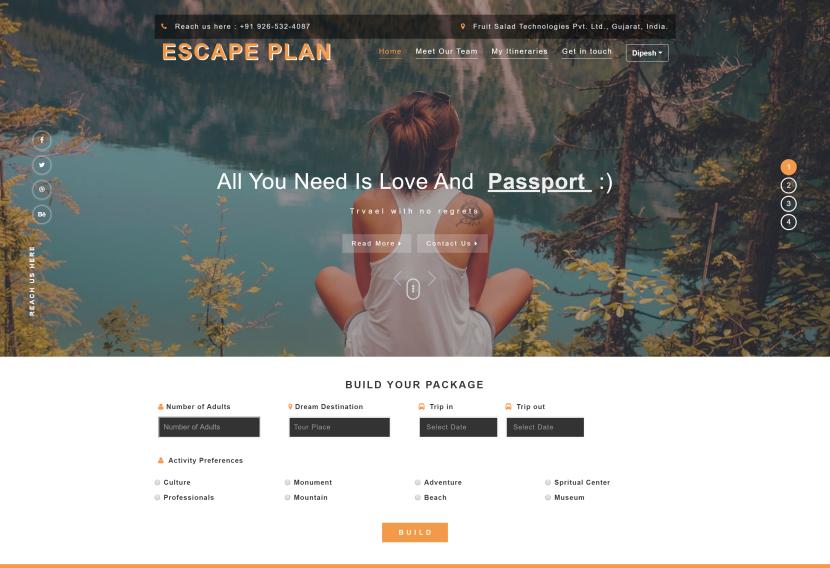
* **Place Recommendation Module**: If the user wishes not to go on some of the places in the itinerary, he/she can customize the same and delete the places that he/she doesn’t want to visit. User is also provided a feature of adding new places other than the ones shown in the current itinerary. After doing so, a new itinerary is generated based on the new places added by the user.
* **Map Plotter Module**: The places shown in the itinerary are plotted on the map using longitudes and lattitudes. Day By Day view of the map is shown to the user. Map contains marked places the user is expected to visit on that day. Longitudes and Lattitudes are fetched using the FourSquare API. The places are marked on the map using the Google Places API, which takes the lat, long pair as input and gives us the location of that place on the map.

This module is also used for displaying and finding the distance and path between the two places on the generated itinerary. The Google Places API used is provided the (lat,long) pair of the two places and accordingly the places are marked on the map. It also gives us the distance between two places and the route between them.

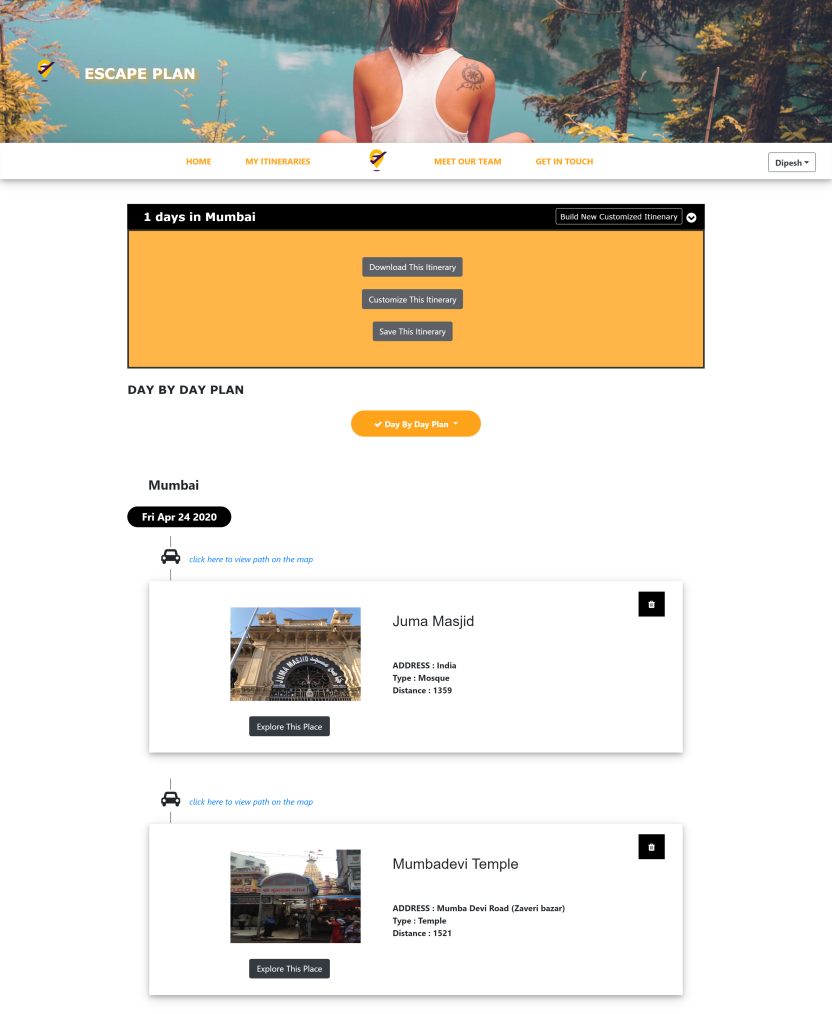
* **Storage Module for the itinerary**: The itinerary generated finally after all the customizations, can be saved by the user for future references.

Based on the journey date we classify the schedule as Ongoing, Upcoming or Past and accordingly display the itinerary in different panels. The itinerary is stored in the MongoDB tables as objects. Each object of the itinerary contains all the necessary details related to that particular place. We have used this schemaless db since we needed to store the itineraries in different format for some of the places. So Mongo Db was choosen as the backend technology.

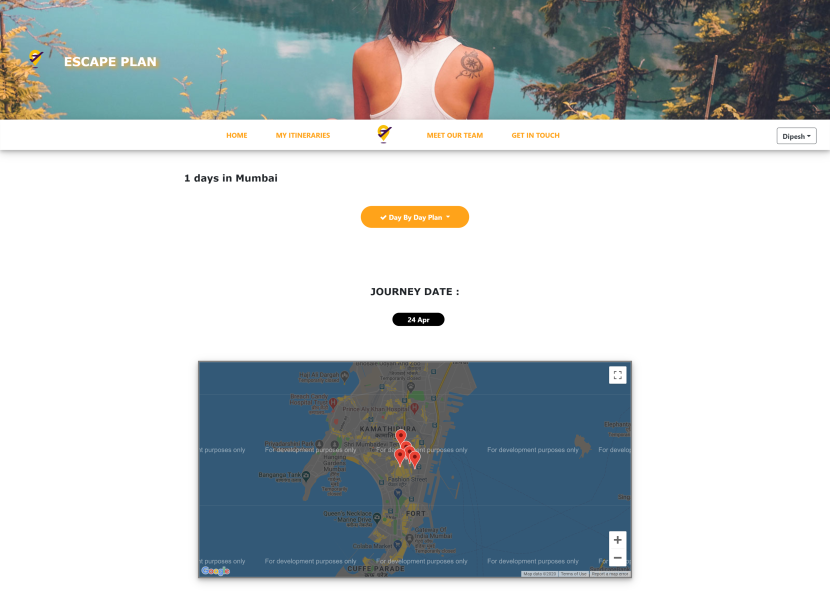
**Screen-shots**



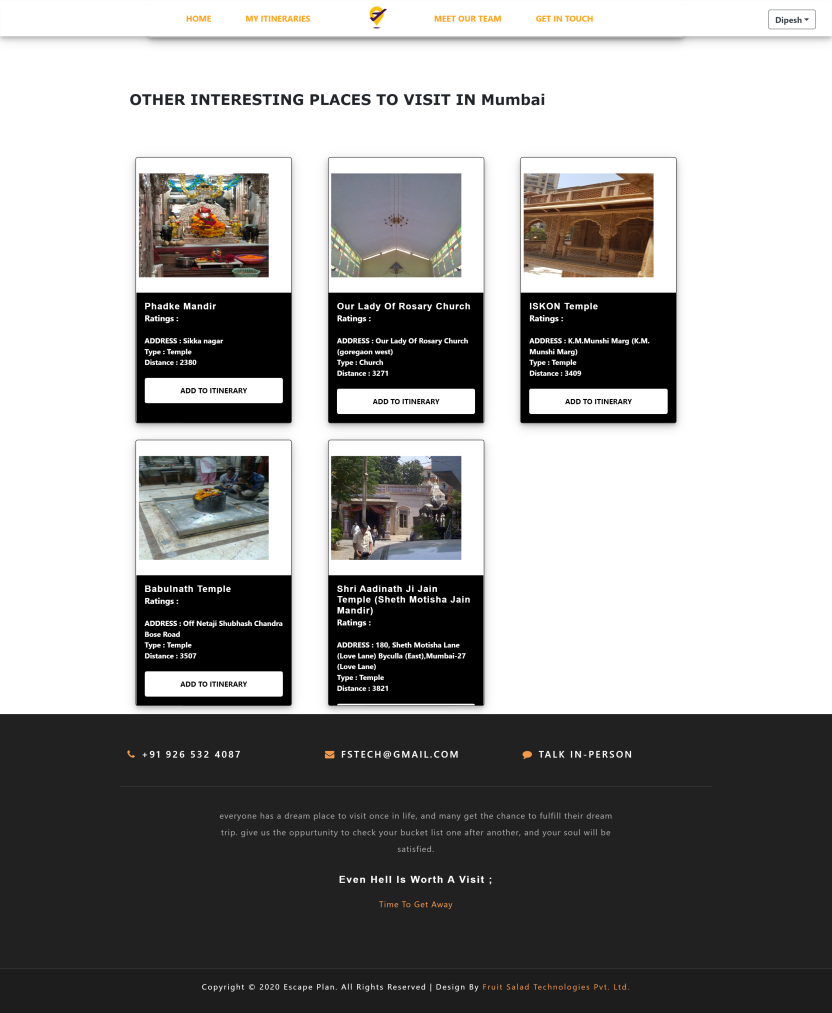
1. Home Page of Escape Plan



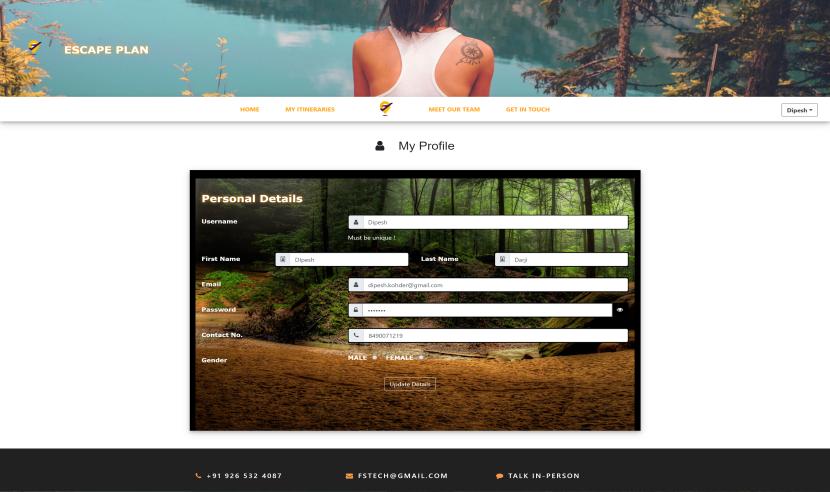
2. Page which shows Itinerary searched by user.



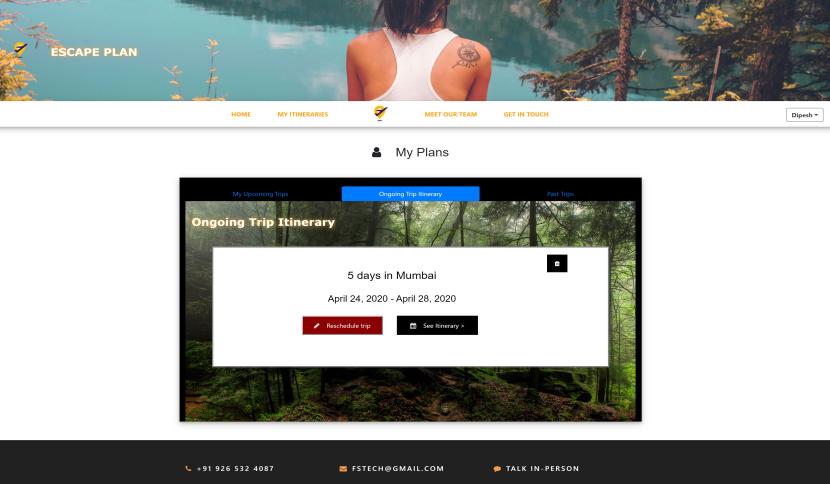
3. Page which shows plotted places on map for Individual days



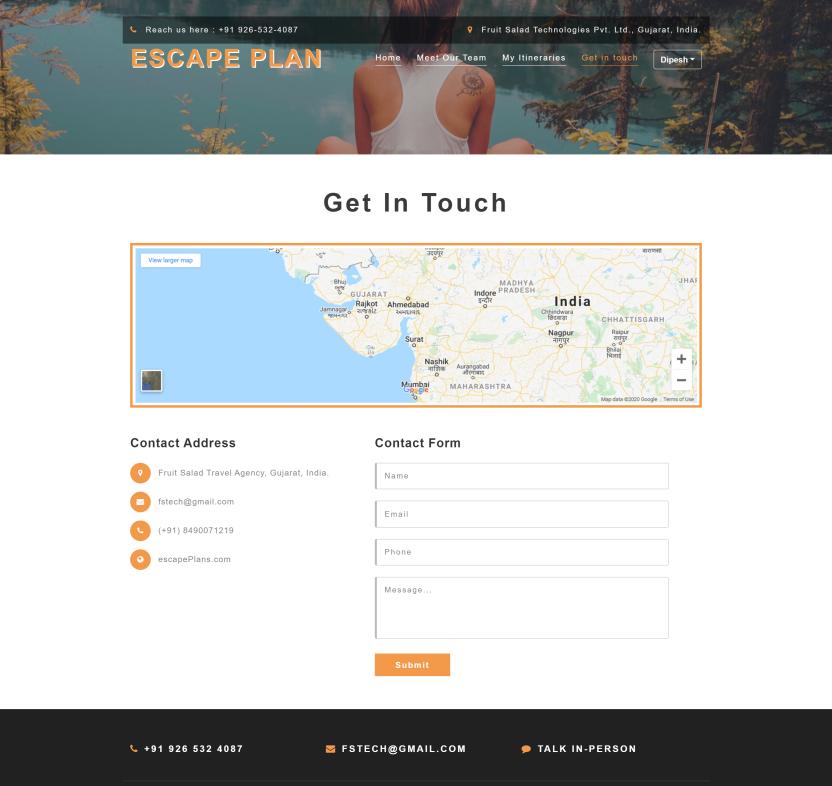
1. Page which shows Suggestion place list for customized Itinerary



1. My Profile page for user



1. My Plans page for the user



1. Get in touch page

**Conclusion**

This project is successfully implemented with the functionalities such as generation of itinerary, string this itinerary, customizing it, displaying the places on the map, etc. This application focuses on making showing the users best possible routes and plan in and around a destination entered by the user. Hence, all the users and travel agencies can use this application for reference before their trip so that they can get a brief idea on what all places to explore around that destination., and accordingly plan their own trip.

**Limitation**

* Currently, we have not used any cryptographic algorithm to secure the passwords and other credentials of the users. Thus, security remains as a limitation.
* We have not used any clustering algorithms for generating the optimized itinerary. The use of TSP algorithm downgrades the response time of the system as it is an exponential time algorithm. Hence, when the itinerary is to be generated for many days, the application takes more time to respond.
* The places fetched from the FourSquare API are not the most popular ones. So sometimes we are not able to generate the itineraries which covers all the famous places around a destination. So this is another limitation.
* Currently, we are not considering the budget of the trip from user. And the fact is not all people across the globe have equal financial status. So, we are not differentiating the luxury of the trip for richer and comparatively poorer people.

**Future Extension**

* This project currently lacks the **Budget Module,** which takes care of the budget of user while planning a trip for him. We look forward to implement that module in the near future.
* We are planning to develop a place recommendation model using machine learning to suggest places to the users based on his past searches and other similar searches of different users. Hence concept of Cosine similarity is to be used in this project.

**BIBLIOGRAPHY**

For the successful implementation of this project we referred to many websites and books.

**Reference Websites:**

* [**https://foursquare.com/**](https://foursquare.com/)
* [**https://cloud.google.com/maps-platform/**](https://cloud.google.com/maps-platform/)
* [**http://stackoverflow.com/**](http://stackoverflow.com/)
* [**https://www.tutorialspoint.com/django/index.htm**](https://www.tutorialspoint.com/django/index.htm)
* [**https://www.w3schools.com/**](https://www.w3schools.com/)

**\*\*\*\*\***