



**Vivekanand Education Society's Institute Of Technology**  
**Department Of Information Technology**  
**Flutter**

**Title:- Tripplanner**

**Made By:**  
Priyank Kulkarni (Roll No:32)

**Mentor Name:**  
Mrs. Ravita Mishra  
(Assistant Professor, VESIT)

# Content

- Introduction to the project
- Problem Statement
- Proposed Solution
- Literature Survey
- Objectives of the project
- Functionalities
- Requirements of the system (Hardware, software)
- Conclusion
- References

# Introduction to Project

- In today's competitive job market, candidates often face challenges in preparing for interviews, especially when it comes to effectively showcasing their communication skills and expressions through video interviews. Traditional interview preparation resources lack a personalized and interactive approach to help candidates enhance their performance in this digital age. There is a need for a web application that leverages video detection technology to assist candidates in refining their interview skills, focusing specifically on facial expressions and non-verbal communication cues.

# Problem Statement

Travelers often struggle to efficiently plan trips, needing to juggle researching destinations, creating itineraries, and managing budgets. This can lead to a stressful and time-consuming planning process.

# Proposed Solution

Develop a mobile application built with Flutter that combines trip planning and destination search functionalities. Users can create personalized trip plans by specifying destinations, travel dates, budget, and trip duration. The app will also offer a search feature to explore different city options, providing details like attractions, activities, and recommendations.

# Objectives

**The Objectives for our project are:**

1. Simplify Trip Planning: To develop a user-friendly mobile application that streamlines the trip planning process, making it less time-consuming and more efficient for travelers.
2. Empower Personalized Itineraries: To equip users with the tools to create custom trip plans tailored to their specific destinations, travel dates, budget constraints, and desired activities.
3. Enhance Destination Discovery: To provide a comprehensive search feature that allows users to explore new cities, uncover hidden gems, and gain valuable insights into popular attractions, recommended activities, and local recommendations.
4. Increase User Engagement: To foster a user-centric experience by potentially incorporating functionalities like budget tracking, user reviews, and visually-appealing elements (photos, virtual tours) to keep users engaged throughout the planning process.
5. Promote Informed Travel Decisions: To empower users with the necessary information to make informed travel decisions that align with their interests and budget.

# Functionalities

## The functionalities for our project are:

The application offers a two-pronged approach to streamline your travel planning journey:

### 1. Personalized Trip Planning:

- Effortless Itinerary Creation: Effortlessly build custom itineraries by specifying your desired destination, travel dates, budget, and trip duration.
- Smart Planning Tools: (Optional) Leverage integrations with mapping services to explore potential routes, discover transportation options, and estimate travel times.
- Budget Management Assistant: (Optional) Keep your spending in check with a budget tracking feature. Set a budget for the entire trip and allocate funds for specific categories like accommodation, activities, and transportation.

### 2. Comprehensive Search Feature:

- Intuitive Search Bar: Uncover the perfect travel destination with a user-friendly search bar at your fingertips.
- Destination Details at a Glance: Dive deep into each city with comprehensive information, including popular attractions, must-do activities, and local recommendations.

# Requirements of the system (Hardware, software)

Sr. No.	Name Of Resource / Material	Specifications
1	Computer system	Android 10 or above
2	Tech stack used	<ul style="list-style-type: none"><li>◦ Flutter</li><li>◦ Openai</li></ul>

# Code Snippets

```
Future<void> _callOpenAI() async {
  try {
    OpenAI.apiKey = 'sk-y1Z2CVZxe01JLqzrzBVkT3B1bkFJUzjPcv7dMcG09tkNhGFD';
    String prompt =
      'Generate a Day-wise trip plan for a $_days day trip to $_city under $_budget';

    OpenAICompletionModel response = await OpenAI.instance.completion.create(
      model: 'davinci-002',
      prompt: prompt,
      maxTokens: 700,
      n: 1,
      stop: null,
      temperature: 0.3,
    );

    setState(() {
      _openAIResponse = response.choices.first.text;

      _openAIError = '';
    });
  } catch (error) {
    setState(() {
      _openAIError = 'Error: ${error.toString()}';
      _openAIResponse = '';
    });
    print(error);
  }
}
```

```
class SearchPage extends StatefulWidget {
  const SearchPage({Key? key}) : super(key: key);

  @override
  State<SearchPage> createState() => _SearchPageState();
}

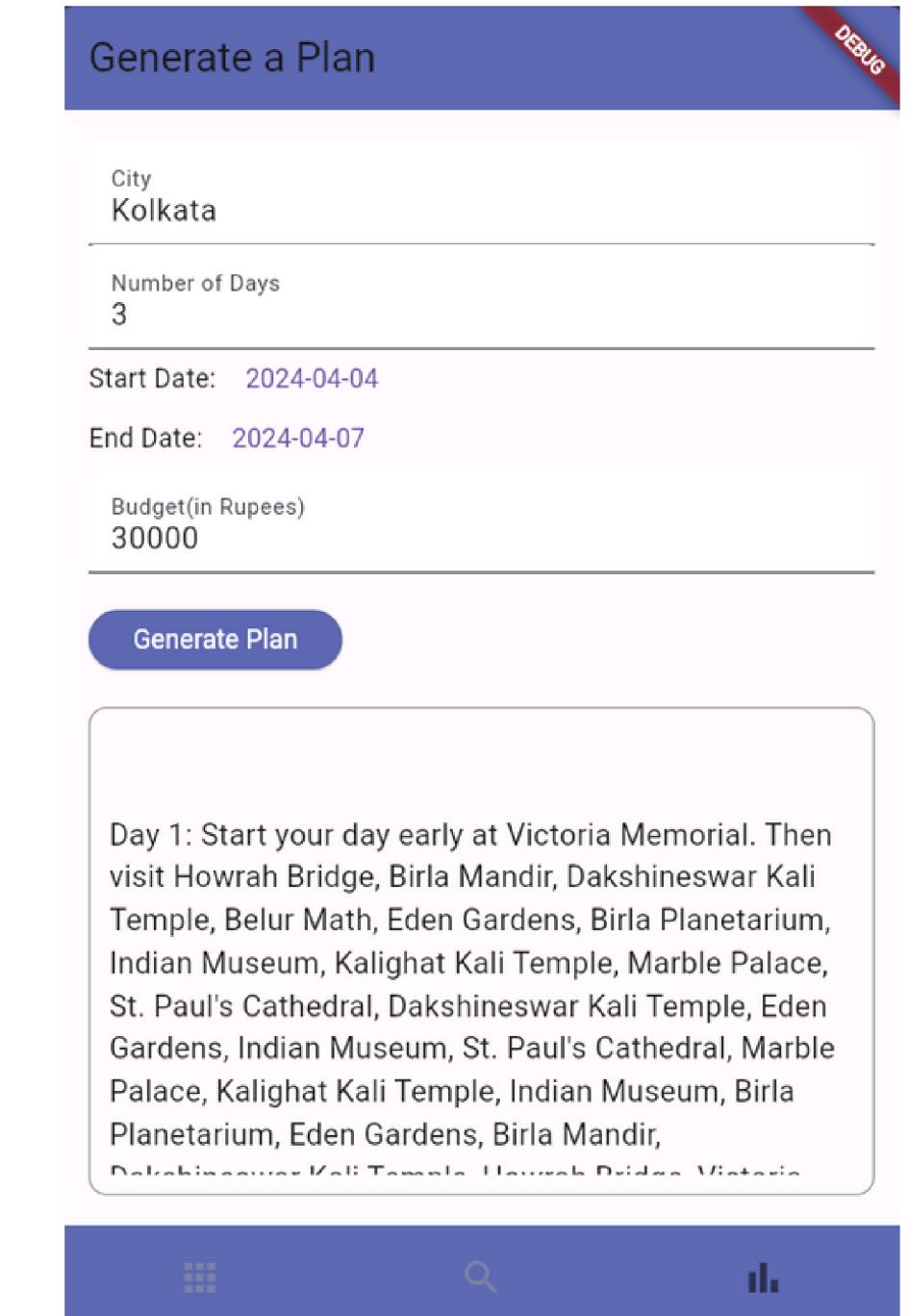
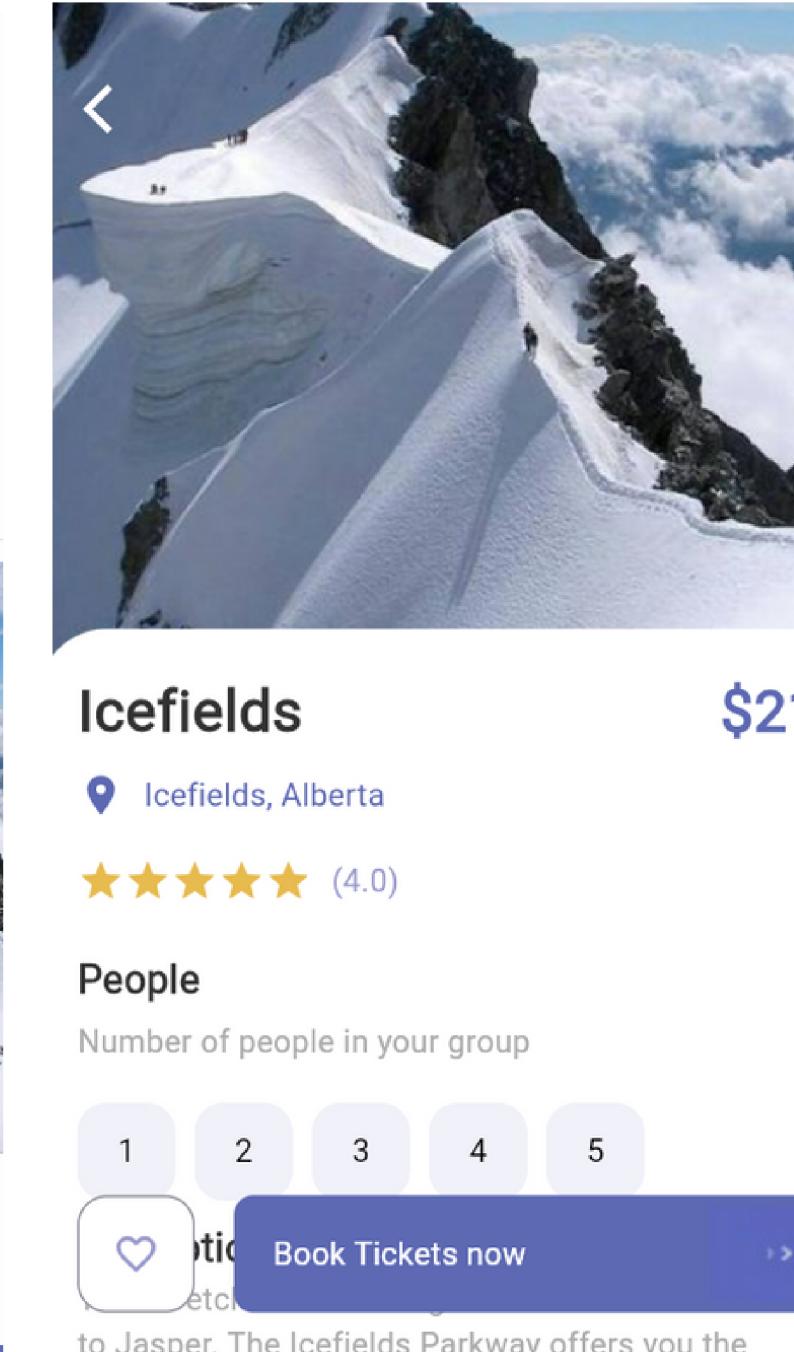
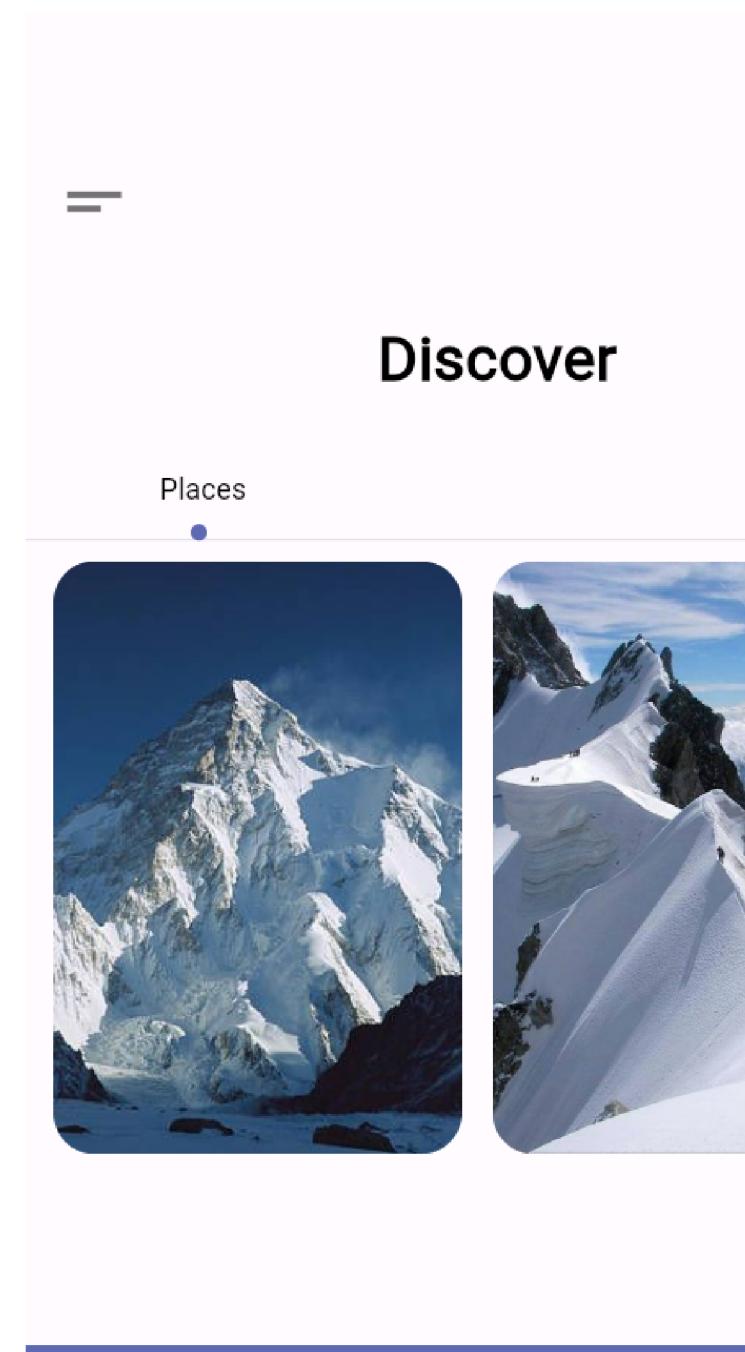
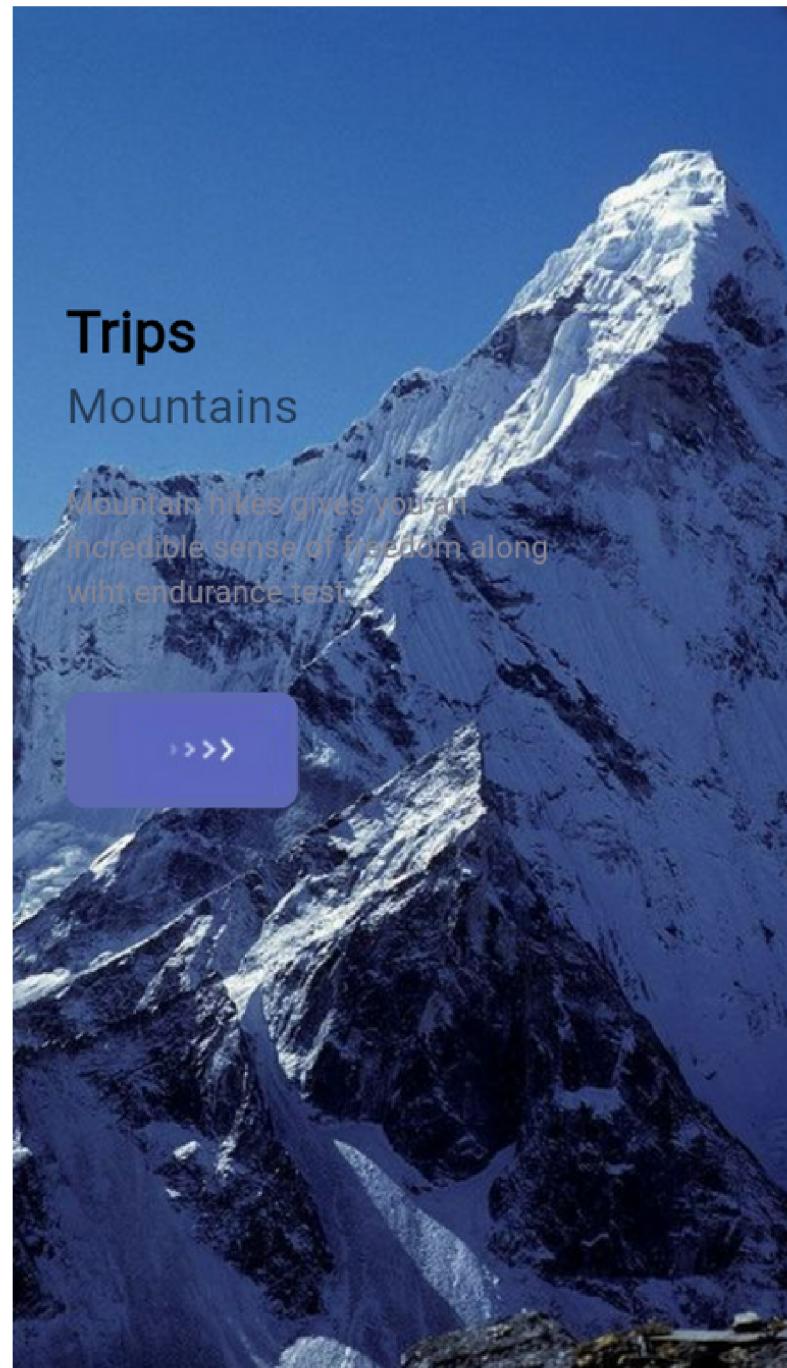
class _SearchPageState extends State<SearchPage> {
  final _formKey = GlobalKey<FormState>();
  final _cityController = TextEditingController();
  String _openAIResponse = '';

  Future<void> _makeOpenAIRequest(String city) async {
    const String openAIUrl = 'https://api.openai.com/v1/completions';
    const String openAIKey = 'sk-y1Z2CVZxe01JLqzrzBVkT3B1bkFJUzjPcv7dMcG09tkNhGFD';

    final Map<String, String> headers = {
      'Content-Type': 'application/json',
      'Authorization': 'Bearer $openAIKey',
    };

    final Map<String, dynamic> body = {
      'model': 'davinci-002',
      'prompt': "Suggest tourist places and activities for a visit to $city",
      'max_tokens': 150,
      'temperature': 0.7,
    };
  }
}
```

# Screenshots



# Conclusion

Interview Buddy bridges the gap between interview prep and real-world confidence. It personalizes the mock interview experience, breaking down barriers to effective skill development. Its user-friendly platform empowers users of all experience levels to practice and refine their communication, leading to impactful interview performances. Interview Buddy fosters the appreciation of one's own interview skills.



**Vivekanand Education Society's Institute Of Technology**  
**Department Of Information Technology**  
**Progressive Web App**

**Title:- Redstore e commerce website**

**Made By:**

Priyank Kulkarni (Roll No:32)

**Mentor Name:**

Mrs. Ravita Mishra  
(Assistant Professor, VESIT)

# Progressive Web Application

RedStore is a Progressive Web App (PWA) designed to provide a seamless e-commerce experience for buying Clothes, Shoes etc. With its user-friendly interface and robust features, Woodex aims to revolutionize the way users shop for furniture online.

## Features:

- Fast loading times.
- Offline access to previously viewed products.
- Add to home screen feature for easy access.

Service worker events for improved performance.

# Problem Statement

**Problem:** Traditional e-commerce websites often suffer from slow loading times and poor user engagement, especially on mobile devices.

Challenges:

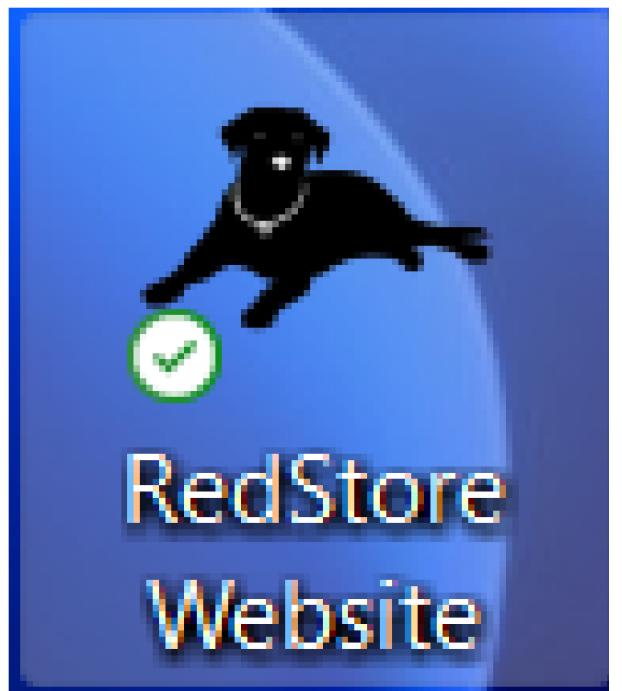
- Limited offline functionality.
- Lack of native app-like experience.
- Poor performance on slower networks.

**Solution:** Develop a PWA that offers fast loading times, offline access, and a responsive design for a seamless user experience.

# Objectives

1. Develop an e-commerce Progressive Web App (PWA) for selling furniture products.
2. Implement PWA features to enhance user experience and performance.
  - A) Write metadata of E-commerce PWA and add to home screen feature.
  - B) Code and register service worker and complete install and activation.
  - C) Implement service worker events like fetch, sync and push for E-commerce.
3. Deploy the PWA on GitHub Pages for easy access and testing.
4. Use Google Lighthouse PWA analysis tool to evaluate and optimize the PWA's performance.

# Implementation



```
4 Fetch successful!                                serviceworker.js:9
Sync successful!                                 serviceworker.js:14
Push notification sent                         serviceworker.js:21
```

# Cache

#	Name	Respo...	Conte...	Conte...	Time ...	Vary H...
0	/	basic	text/h...	15,901	04/04/...	Origin
1	/images/android-chrome-192x192.p...	basic	image...	6,997	04/04/...	Origin
2	/images/app-store.png	basic	image...	8,974	04/04/...	Origin
3	/images/cart.png	basic	image...	2,403	04/04/...	Origin
4	/images/category-1.jpg	basic	image...	226,052	04/04/...	Origin
5	/images/category-2.jpg	basic	image...	146,279	04/04/...	Origin
6	/images/category-3.jpg	basic	image...	166,233	04/04/...	Origin
7	/images/exclusive.png	basic	image...	205,943	04/04/...	Origin
8	/images/image1.png	basic	image...	2,592,...	04/04/...	Origin
9	/images/logo-coca-cola.png	basic	image...	9,664	04/04/...	Origin
10	/images/logo-godrej.png	basic	image...	16,763	04/04/...	Origin

# Lighthouse Report

