

Tirtha

A Flutter Based Pilgrimage app

Documentation

Group: S2T2

Mentor: Yash Anand Giri

Team members:

Abhibhav Chaudhary Subhag Sharma Suman Saurabh Manya Verma Rahil Saini

Problem Statement

The ultimate goal is to craft a technological solution that elevates the pilgrimage experience for every devotee visiting Shri Mata Vaishno Devi, ensuring it's spiritually enriching, safe, and hassle-free. The solution should be user-friendly, accessible, and considerate of the diverse demographics of the pilgrims.

Description: Every year, millions of pilgrims embark on a spiritual journey to Shri Mata Vaishno Devi, a revered temple located in the challenging terrains of Jammu and Kashmir. The path spans approximately 14 km and is often trodden by devotees who are unfamiliar with the region, its customs, and the many facilities available on route.

Challenge: Develop a software application that:

- **Guides the Devotee:** Many first-time visitors are unfamiliar with the path, from where to go where, sim card problem and what not. How can we ensure they are guided every step of the way, know where they are, and how much distance remains? Informs about Cultural Significance: How can we provide real-time, contextual information about the various points of significance, rituals, and stories associated with the temple?
- **Aids in Emergencies:** In the hilly terrains and vast crowds, emergencies can occur. How can we ensure that a devotee in distress can quickly get the help they need?
- **Facilitates Basic Needs:** Accommodation and food are primary concerns. How can we guide devotees to appropriate facilities based on their needs, preferences, and budgets?
- **Encourages Community Interaction:** How can we provide a platform for devotees to share their experiences.

Tech Stack Used: Flutter (Dart), Firebase, Riverpod

*Depending on the specific requirements of the app the tech stack is subject to change.

- **State management:** Flutter does not come with a built-in state management solution, so we will need a third-party library. Popular options include Riverpod.
- **Networking:** To communicate with a backend server, we will need to use a networking library. (Preferred: flutter_websockets)
- **Testing:** It is important to test app thoroughly to ensure its quality and reliability. For this we'll use **Hot reload**.

Here are some additional details:

Flutter + Riverpod + Firebase: This is a popular tech stack for developing simple to medium-complexity Flutter apps.

Flutter: Flutter is an open-source UI toolkit for building beautiful, native-looking apps for mobile, web, and desktop from a single codebase. Flutter is preferred for a number of reasons, including:

- **Cross-platform development:** Flutter allows developers to build native apps for multiple platforms, such as iOS, Android, web, and desktop, from a single codebase. This can save a lot of time and money, as developers do not need to write and maintain separate codebases for each platform.
- **High performance:** Flutter apps are known for their high performance and smooth animations. This is because Flutter renders the UI directly to the native canvas, without the need for a bridge.
- **Expressive UI:** Flutter provides a rich set of widgets that allow developers to create expressive and customizable user interfaces.
- **Hot reload:** Flutter's hot reload feature allows developers to see changes to their code reflected in the running app almost instantly. This can make development much faster and more efficient.

Riverpod: Riverpod is a reactive caching and data-binding framework for Flutter/Dart. It can automatically fetch, cache, combine and recompute network requests, while also taking care of errors for you. It is a reconstruction of the Provider package; meaning it reduces dependencies.

Riverpod provides a number of features that make it a popular choice for state management in Flutter apps, including:

- **Reactivity:** Riverpod is reactive, meaning that it automatically updates widgets when the underlying state changes. This makes it easy to create dynamic and responsive user interfaces.
- **Caching:** Riverpod can cache expensive state computations, which can improve performance and reduce battery drain.
- **Dependency injection:** Riverpod makes it easy to inject dependencies into widgets, which can improve code readability and maintainability.
- **Testing:** Riverpod is easy to test, as each provider can be overridden to behave differently during a test.

Firebase: Firebase is a mobile platform that helps developers build high-quality apps and grow their business. It provides a suite of tools and services to help developers build, deploy, and scale their apps, including:

- **Authentication:** Firebase provides a variety of authentication methods, such as email/password, phone number, and social sign-in.
- **Real-time database:** Firebase provides a real-time database that allows developers to store and sync data between their users in real time.
- **Cloud messaging:** Firebase provides a cloud messaging service that allows developers to send push notifications to their users' devices.
- **Crashlytics:** Firebase provides a crash reporting service that helps developers identify and fix crashes in their apps.
- **Analytics:** Firebase provides an analytics service that helps developers track the usage of their apps and gain insights into their users.

Key features of the app:

GPS navigation: The app should be able to track the user's location and guide them on the pilgrimage path.

Cultural information: The app should provide information about the various points of significance on the pilgrimage path, such as temples, shrines, and natural landmarks. It can also provide information about the rituals and stories associated with these locations.

Emergency assistance: The app should have a panic button that the user can press in the event of an emergency. This will send an alert to the nearest emergency services, providing them with the user's location and other relevant information.

Basic needs: The app should list nearby accommodation and food options, along with their prices and ratings. It can also allow the user to book accommodation and meals in advance.

Community interaction: The app should include a forum or chat feature where users can share their experiences, ask questions, and offer support to each other.

To implement these features, you can use the following Flutter plugins:

Flutter Google Maps: This plugin provides the necessary functionality for GPS navigation.

Flutter Wikipedia API: This plugin can be used to retrieve information about the various points of significance on the pilgrimage path.

Flutter Emergency Contact: This plugin can be used to send an alert to the nearest emergency services in the event of an emergency.

Flutter Booking: This plugin can be used to book accommodation and meals in advance.

Flutter Forum: This plugin can be used to implement a forum or chat feature.

Here are some algorithms that can be used in building the app:

Routing algorithm: This can be used to calculate the shortest or fastest route between the user's current location and the next point of significance on the pilgrimage path.

Landmark recognition algorithm: This algorithm can be used to identify landmarks on the pilgrimage path, such as temples, shrines, and natural landmarks. The app can then provide information about these landmarks to the user, such as their history and significance.

Navigation algorithm: This algorithm can be used to guide the user on the pilgrimage path. The app can use the user's GPS location and the calculated route to provide turn-by-turn directions.

Recommendation algorithm: This algorithm can be used to recommend accommodation and food options to the user based on their needs, preferences, and budget. The app can consider factors such as the user's location, price range, and dietary restrictions when making recommendations.

Natural language processing (NLP) algorithm: This algorithm can be used to process user queries and provide relevant information, such as information about the pilgrimage path or the various points of significance on the path. For example, the user could ask the app "What is the next point of significance on the pilgrimage path?" or "What is the history of this temple?"

UI/UX

A well-designed UI/UX will make the app easy to use and navigate, while also providing a visually appealing and engaging experience for users.

There are a number of factors we will consider when designing the UI/UX, including:

- **User-friendliness:** The app should be easy to use and navigate, even for users with limited technical expertise. This means using clear and concise language, avoiding jargon and technical terms, and providing clear instructions.
- **Accessibility:** The app should be accessible to all users, regardless of their age, gender, or disability. This means using large font sizes, high-contrast colors, and alternative input methods, such as voice input.
- **Cultural sensitivity:** The app should be respectful of the cultural and religious sensitivities of the pilgrims. This means using appropriate language and imagery, and avoiding any content that could be considered offensive or disrespectful.