in collaboration with

Softwarica College of IT & E-commerce | Coventry University

SAJILOTANTRA

ST6002CEM Mobile Application Development

Submitted to

Kiran Rana

Submitted By

Nikesh Bhandari

Coventry ID: 13703679

Table of Contents

Table of Figures

Introduction

Sajilotantra is a mobile application designed to simplify the bureaucratic processes in Nepal by providing a user-friendly platform. The app allows users to share their thoughts, frustrations, and suggestions about government procedures in a social media-like environment. Additionally, it offers step by step guidance on creating official documents and a map to search and locate nearby government offices, and a calendar to track events. The name "Sajilotantra" reflects its mission which is making bureaucracy (tantra) easier (sajilo) for Nepali citizens. This project aim to address the common challenges people face, such as lack of information on costs, required documents, and procedural steps, by providing all necessary details into one accessible platform.

Aims

The primary aim of Sajilotantra is to help Nepali citizens by simplifying their interaction with government systems.

Objectives

- Providing a social platform for users to express opinions and experiences about government processes.
- Offering step-by-step guidance on creating official documents, including costs and required documents.
- Integrating location-based services to help users find government offices.
- Enhancing transparency and accessibility in bureaucratic processes.

Project Background

Citizens in rural areas and those who are not tech-savvy face major difficulties interacting with the bureaucratic system of Nepal due to its complicated nature and unclear documentation. The public faces difficulties understanding what documentation they require combined with visiting locations and acquiring procedure fees. The current governmental sites that exist lack modern features combined with poor user interfaces. Sajilotantra operates as a communication solution through mobile technology in Nepal because mobile devices reach most areas in the country. The application combines social features with useful functionalities which draw inspiration from successful social networking applications along with government service applications that exist internationally. The initiative started with the intention to increase citizen involvement and create less fearful bureaucracies.

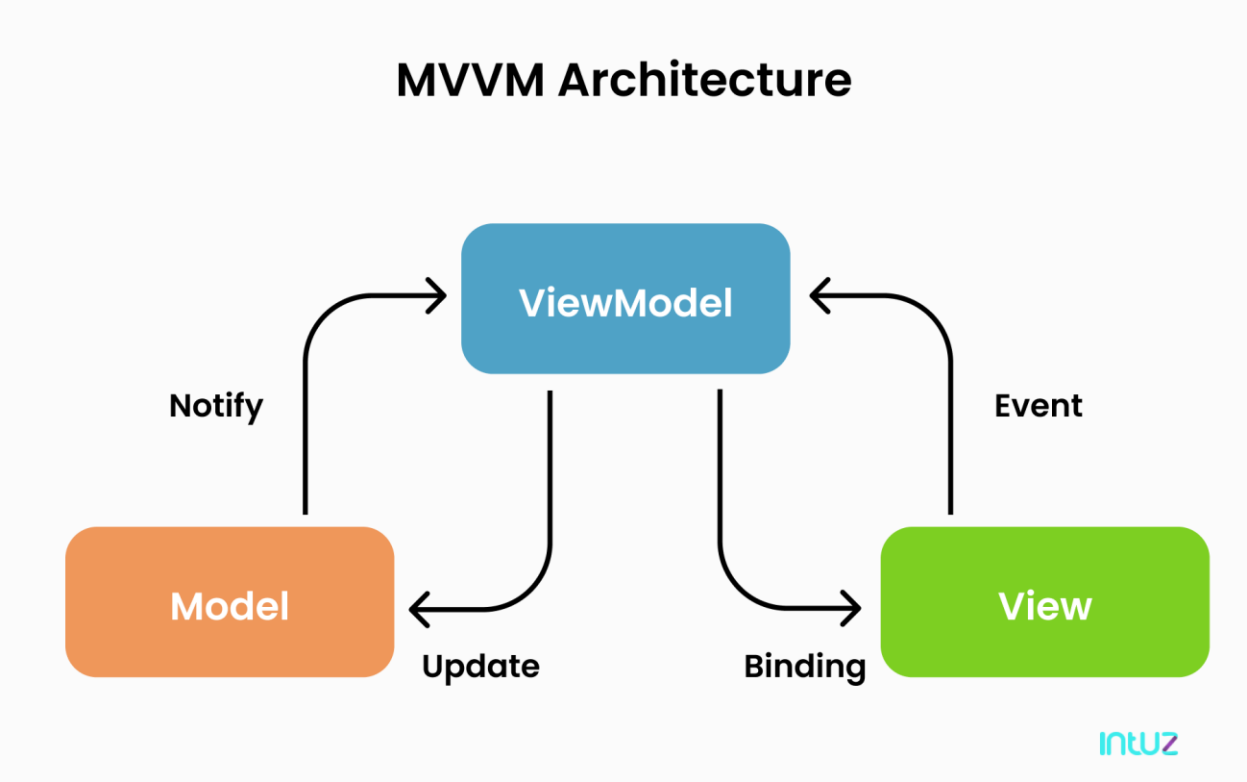
Design

Design Pattern (MVVM)

Sajilotantra uses the Model-View-ViewModel (MVVM) design pattern to ensure a clean, scalable, and maintainable codebase. MVVM separates the app into three components:

- Model: Represents the data (e.g., user posts, document templates, or API responses).
- View: The user interface (e.g., screens displaying guidance or maps).
- ViewModel: Acts as a mediator, handling business logic and preparing data for the View.

Figure 1: MVVM Architecture



(William, 2023b)

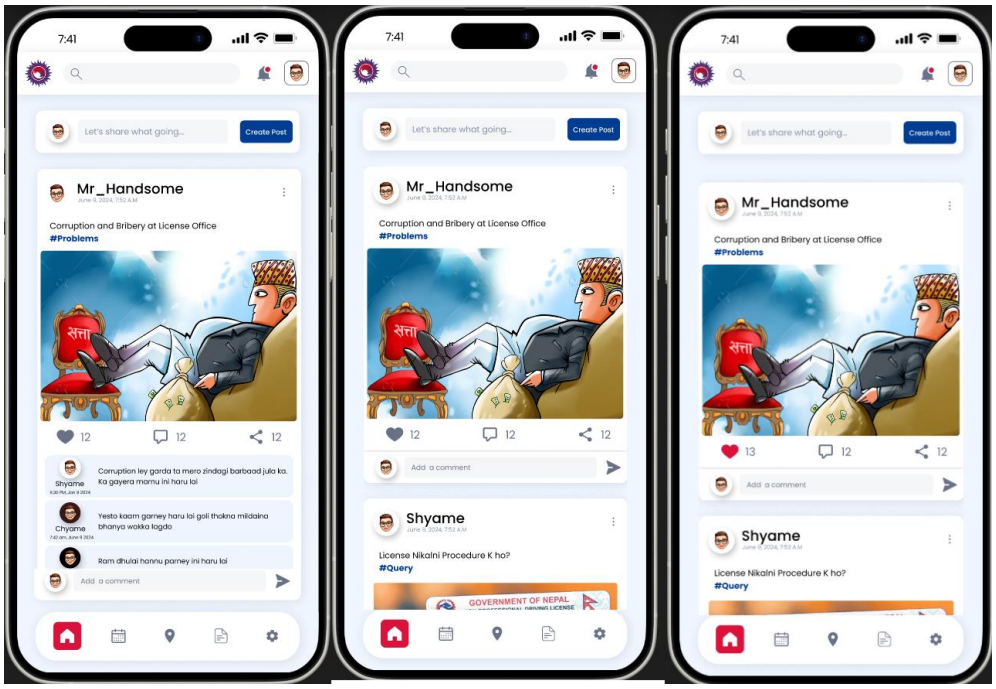
This pattern enhances testability and allows developers to update the UI without altering the underlying logic. For example, in Sajilotantra, the ViewModel fetches government office locations and passes them to the View for display on a map.

Wireframes

The app's user interface was designed using Figma, a collaborative design tool. Wireframes were created to visualize the layout of key screens, such as the social feed, guidance section, map, and calendar. The design prioritizes simplicity and accessibility, with Nepali language support and intuitive navigation.

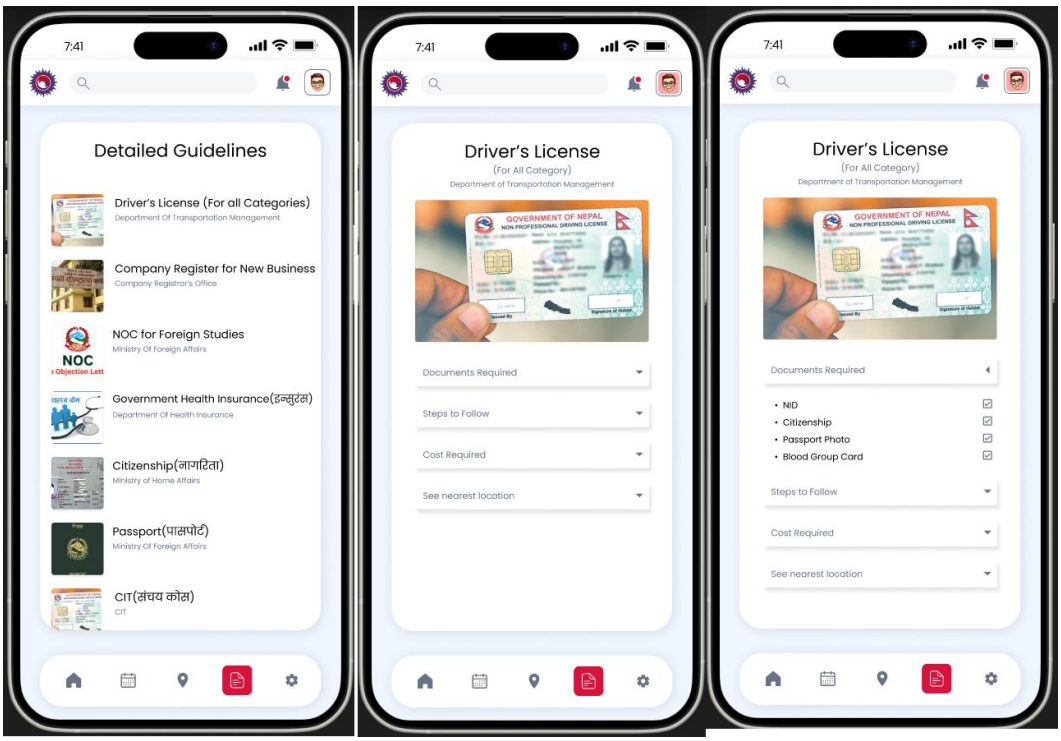
- Social Feed: A scrolling list of user posts with like/comment options.

Figure 2: Dashboard Feed



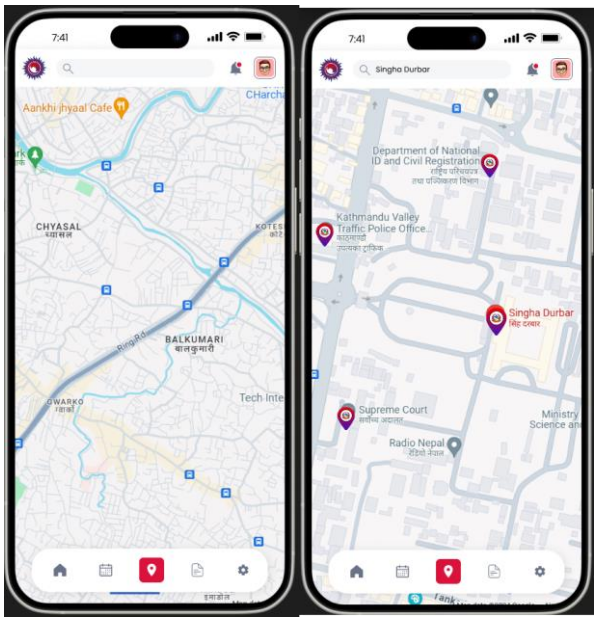
- Guidance Section: Step-by-step document creation guides with expandable sections.

Figure 3: Guidance Page



- Map: Interactive map with pinned government offices.

Figure 4: Map



- Calendar: Monthly view of events with clickable details.

Bloc Integration

Bloc (Business Logic Component) pattern for state management was implemented to handle the app's dynamic data efficiently. Bloc separates the UI from business logic, making it easier to manage states like loading, success, or error when fetching API data or user inputs. For example, when a user searches for a guidance details, Bloc manages the state transitions (e.g., "loading" to "data loaded") and updates the UI accordingly.

Figure 5: Folder Structure

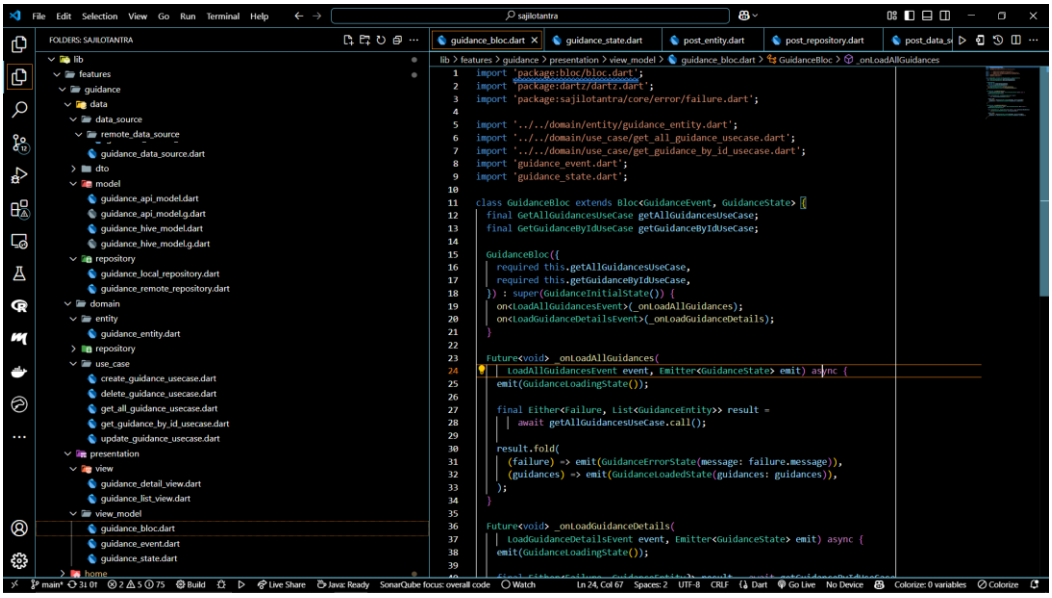
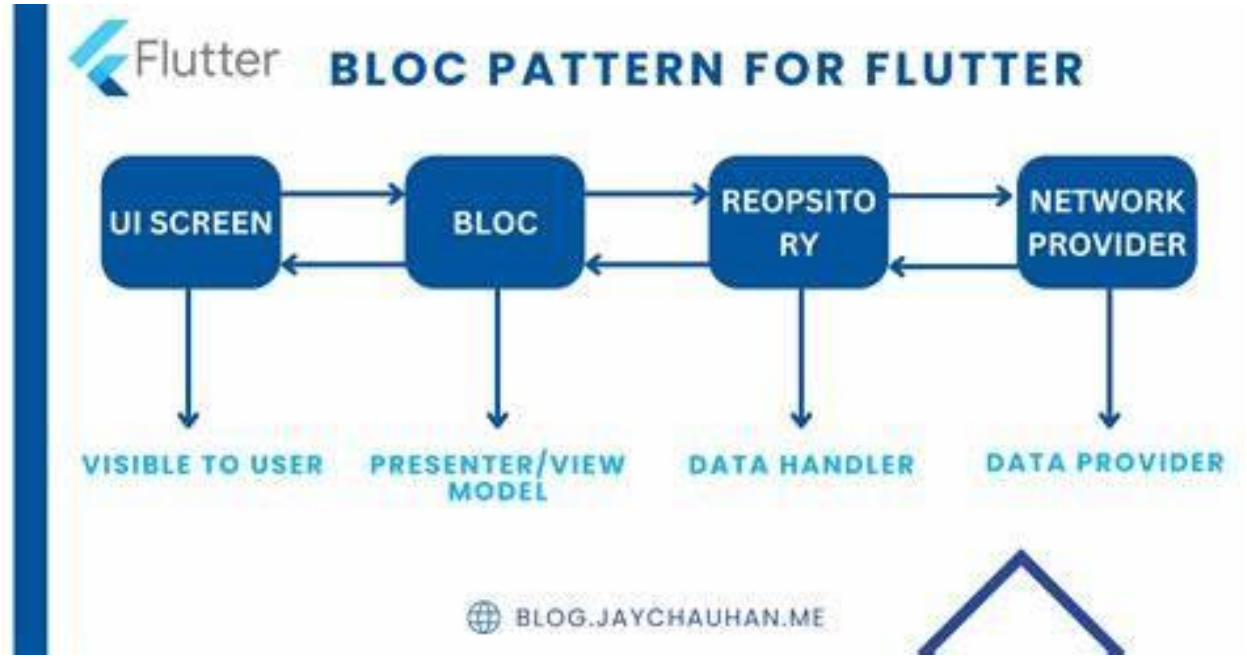


Figure 6: Bloc



(Rashid, 2023)

Data and Security

For the time being Sajilotantra is running locally. It uses hive and shared preference to store data. The mobile application is interactive with backend server with remote data. Cloudinary is being used now and will be using cloud for automatic backup in future

Cloud Computing

Local storage makes up the core operational principle of Sajilotantra to ensure device-based data processing which gives better access for rural areas with restricted internet access. Sajilotantra currently uses Cloudinary as a cloud-based media storage solution for managing efficient delivery of images and documents and multimedia content. The implementation of Cloudinary at Sajilotantra leads to increased performance and diminishes storage requirement on local devices together with quick media asset retrieval.

Data Handling

Local Data

Sajilotantra use the Hive database system to store information about user preferences together with browsing activities and stored government service content. The application depends on Shared Preferences to manage lightweight data storage such as authentication tokens and app settings to keep users automatically logged in without wasting API requests. The system implements data protection through encryption of sensitive information together with local device storage based on essentials only to lower possible security risks.

Remote Data

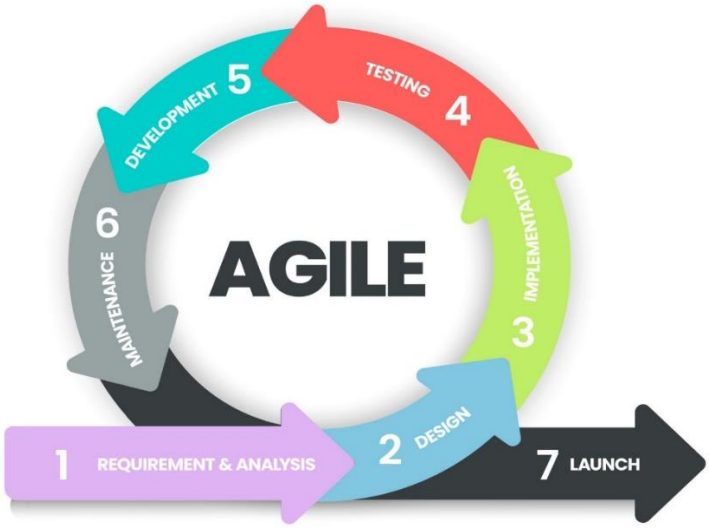
Dynamic content in Sajilotantra it fetched by using secure APIs to talk with a backend system built on Node.js and MongoDB. Through secured APIs the system enables different operations for acquiring government office information and processing document guidance procedures while handling user responses and managing notification functionality. User information receives protection against unauthorized access by implementing security protocols which include JWT authentication and encrypted data transactions and HTTPS enforcement and role-based access control. Sajilotantra unites local and remote data processing to provide users with an efficient and protected platform experience that enhances access and performance across all application features.

Methodology and Tools

Agile

The project sticked to Agile methodology, focusing on iterative development and adaptability. Sprints were conducted to plan features, develop prototypes, and incorporate feedback. This methodology prioritized essential features, such as guidance and maps evolving requirements effectively.

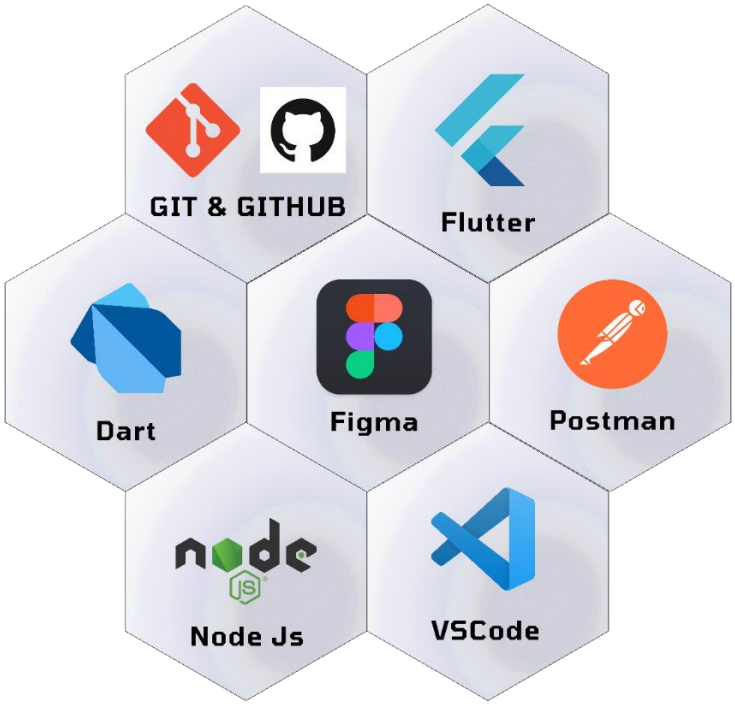
Figure 7: Agile Methodology



(GeeksforGeeks, 2022)

Tools

Figure 8: Tools and Technology



App Monetization

The revenue generation model of Sajilotantra combines free accessibility with premium services to sustain operations into the long term. Users benefit from unlimited free accessibility together with premium features that maintain the platform's effectiveness and continuing service value to citizens.

Non-intrusive advertisements based on government services and legal documentation assistance and local businesses will be displayed as banners and pop-ups in Sajilotantra. The advertisers will receive placement points that maintain user flow and deliver free usage for most platform users.

Premium users who choose the subscription plan can access offline document templates and guides through downloads which benefit subscribers without continuous internet access. Subscription members who use the premium service will get immediate priority support as well as specialized help with their questions about government service requirements.

Paid membership allows users to obtain realistic government-issued forms which are structured professionally and closely match official documents. Users who access this special feature can correctly complete their applications independently, so they do not have to bother anyone else while reducing errors and shortening the completion time.

Sajilotantra implements a freemium business approach by providing free essential services to users who need to pay for more elaborate options. Free services on Sajilotantra include general guidance and maps and discussion forums but premium tools such as interactive document creation and AI assistance and real-time consultancy require payment from users.

Conclusion

Sajilotantra is a digital system which simplifies Nepalese bureaucracy through practical tools combined with social features. The system provides clear guidance for accessing government services together with office location tracking features through which users can connect in social networking spaces to share experiences. The platform unites these features to provide citizens with real-time information that remains simple to understand thereby decreasing both confusion and bureaucratic inefficiency levels.

Three core functionalities found in the software system serve to enhance engagement system operations. The social interaction system features a platform which enables users to publish inquiries along with bureaucratic obstacles and participate in exchanges with people experiencing comparable situations. The detailed document guidance module guides users through every stage of government document acquisition by showing them step-by-step procedures and necessary costs together with procedure timelines. The location-oriented service of this system helps users discover nearby government offices simultaneously offering access through a graphical display of local facilities.

Cloud service integration represents a future development priority because it enables real-time data sharing while providing access to multiple devices. User testing will be carried out to enhance the interface while guaranteeing smooth user experiences for every demographic group. Sajilotantra continues to develop as it moves toward transforming Nepal's bureaucratic system by improving transparency and efficiency and enhancing accessibility.

Reference

William. (2023, November 14). 62 essential What is MVVM Architecture Recomended Post. *Ultimate Android Apps Collection*. https://droidchip.github.io/android-tablet/what-is-mvvm-architecture/

GeeksforGeeks. (2022, December 19). What is Mobile Cloud Computing? GeeksforGeeks. https://www.geeksforgeeks.org/what-is-mobile-cloud-computing/

GeeksforGeeks. (2024, August 1). *Shared preferences in Flutter*. GeeksforGeeks. https://www.geeksforgeeks.org/shared-preferences-in-flutter/

Palma, G. (2024, February 14). Dio interceptors in flutter - flutter community - medium. *Medium*. https://medium.com/flutter-community/dio-interceptors-in-flutter-17be4214f363

Rashid. (2023, September 22). Mastering State Management in Flutter with Bloc: A Comprehensive Guide. *Medium*. https://medium.com/@dihsar/mastering-state-management-in-flutter-with-bloc-a-comprehensive-guide-1d03319ba7df

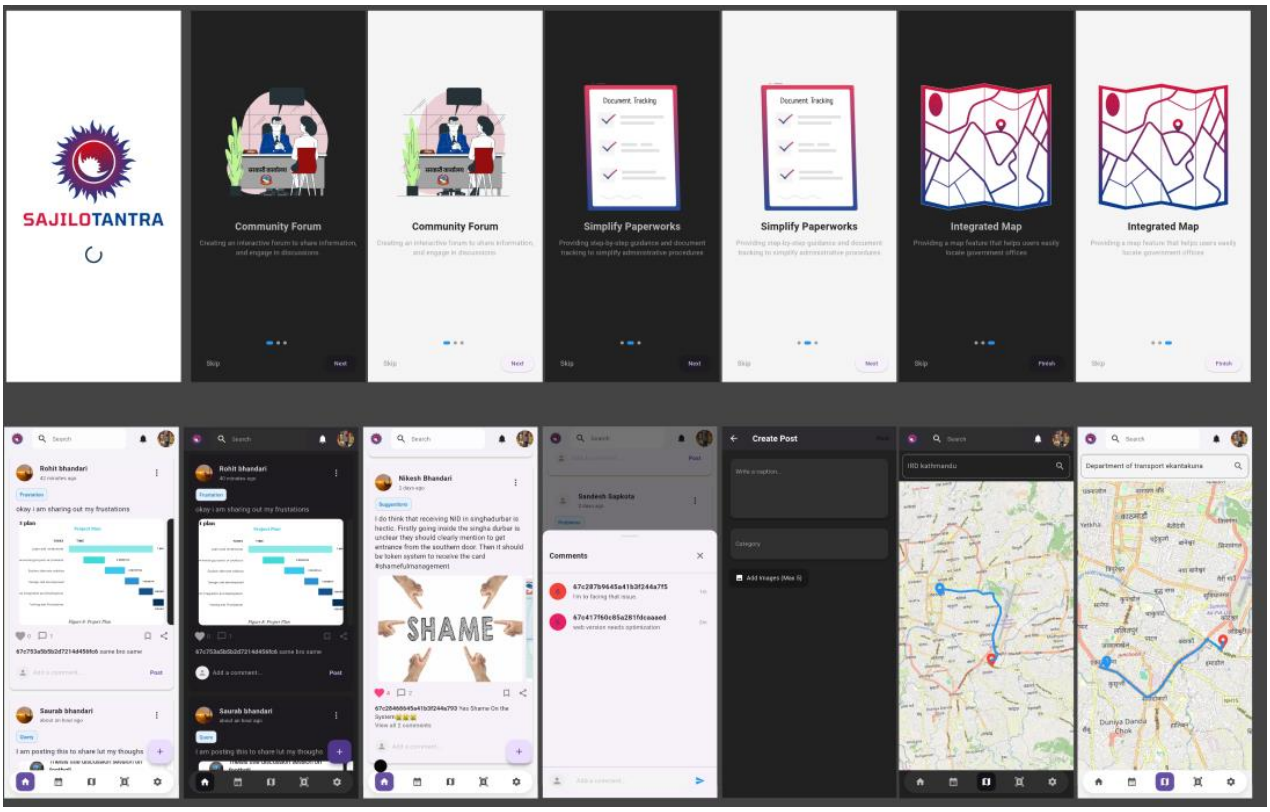
Rajput, K. (2023, September 24). Connecting MongoDB to a Flutter App Using mongo_dart. Medium. https://kailash8799.medium.com/connecting-mongodb-to-a-flutter-app-using-mongo-dart-f154aae66a4d

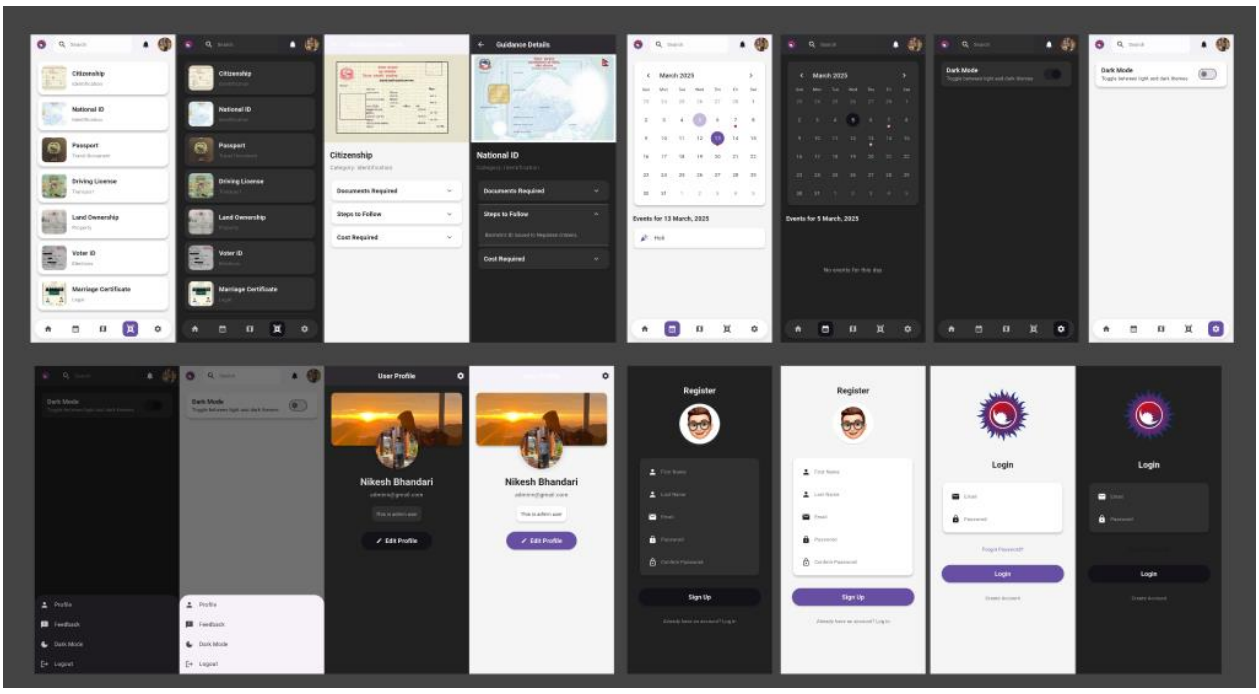
Flutter & Hive Database: CRUD Example (updated) - KindaCode. (n.d.). https://www.kindacode.com/article/flutter-hive-database

Flutter: How to like and unlike a post using like button package. (n.d.). Stack Overflow. https://stackoverflow.com/questions/74761787/flutter-how-to-like-and-unlike-a-post-using-like-button-package

Tarapara, K. (2023, June 26). *Implementing Flutter with NodeJS | Bosc Tech Labs*. BOSC Tech Labs. https://bosctechlabs.com/how-to-implement-flutter-with-node-js/

Appendix


SAJILOTANTRA
Community Forum
Community Forum
Simplify Paperworks
Simplify Paperworks
Integrated Map
Integrated Map
Create Post
Comments
SHAME


Citizenship
National ID
Passport
Driving License
Land Ownership
Voter ID
Marriage Certificate
Guidance Details
Documents Required
Steps to Follow
Cost Required
User Profile
Nikesh Bhandari
Edit Profile
Register
Sign Up
Login