



Arva: Smart Agricultural Assistant



Project Structure

```
lib/
  └── core/ # Global app configurations
    |   ├── di/
    |   |   └── injection_container.dart # GetIt setup for BLoC and Repositories
    |   ├── network/
    |   |   └── api_client.dart # Retrofit/Dio configuration
    |   ├── theme/
    |   |   └── app_theme.dart # Theme for Mobile and Web compatibility
    |   └── utils/
    |       └── app_constants.dart # Requirement IDs and API endpoints
  └── features/ # Feature-based folders
    └── auth/ # Login & Registration
      |   ├── data/
      |   |   ├── datasources/ # Remote auth calls
      |   |   └── repositories/ # Auth repository implementation
      |   ├── domain/
      |   |   ├── entities/ # User/Admin entity
      |   |   └── usecases/ # Login/Register use cases
      |   └── presentation/
      |       ├── bloc/ # auth_bloc.dart
      |       ├── pages/ # login_page.dart, registration_page.dart
      |       └── widgets/
    └── dashboard/ # User & Admin Dashboards
      |   ├── presentation/
      |   |   ├── bloc/ # dashboard_bloc.dart
      |   |   └── pages/ # user_dashboard.dart, admin_dashboard.dart
    └── soil_analysis/ # Dynamic Page
      |   ├── data/

```

```

|   |   |
|   |   |   └── models/ # soil_data_model.dart (NPK, pH, Moisture)
|   |   |   └── repositories/ # Sensor data repository implementation
|   |   └── domain/
|   |   |   └── entities/ # Soil reading entity
|   |   |   └── usecases/ # get_sensor_readings.dart
|   |   └── presentation/
|   |   |   └── bloc/ # soil_bloc.dart (Handles real-time updates)
|   |   |   └── pages/ # soil_analysis_page.dart
|   |   |   └── widgets/ # npk_gauges.dart
|
|   |
|   └── crop_recommendation/ # Static Page
|       └── data/
|           |   └── models/ # recommendation_model.dart
|           └── presentation/
|               └── bloc/ # recommendation_bloc.dart
|                   └── pages/ # crop_recommendation_page.dart
|
|   |
|   └── plant_monitoring/ # Disease & Pest Detection
|       └── domain/
|           |   └── entities/ # Disease entity (Name, ID, Confidence)
|           └── presentation/
|               └── bloc/ # monitoring_bloc.dart
|                   └── pages/ # plant_health_page.dart, pest_detection_page.dart
|
|   |
|   └── treatment/ # Irrigation & Fertilization
|       └── presentation/
|           └── bloc/ # treatment_bloc.dart
|               └── pages/ # irrigation_fertilization_page.dart
|
└── shared/ # Reusable UI components
    └── widgets/ # Custom buttons, text fields, cards
|
└── main.dart

```

Dependencies to Install

- flutter pub add dio easy_localization firebase_core flutter_bloc flutter_native_splash
flutter_screenutil flutter_svg freezed_annotation get_it json_annotation pretty_dio_logger retrofit

- flutter pub add --dev build_runner freezed json_serializable retrofit_generator flutter_lints
- flutter pub add syncfusion_flutter_gauges

Dependency Explanation

- **flutter_bloc**: Excellent for managing the state of your app, such as switching between User and Admin views or handling the dynamic Soil Analysis state.
- **get_it**: A service locator used for Dependency Injection. It allows you to swap your "Static Data Source" for a "Real Sensor Data Source" later without changing your UI.
- **freezed & json_serializable**: These automate the creation of Data Models. They are perfect for handling the complex Soil Analysis report objects (NPK, pH, moisture, fertility score).
- **dio & retrofit**: These are used to make HTTP requests to your backend. You'll use these when you move away from static data to get real readings from the Data Warehouse.
- **pretty_dio_logger**: A great tool for debugging the data being sent between your app and the API. # App entry point with ScreenUtil setup
- **flutter_screenutil**: Since you want compatibility for both mobile and website view, this helps you create a responsive UI that scales across different screen sizes.
- **flutter_svg**: Best for agricultural icons (plants, insects, droplets) as they stay sharp at any size.
- **flutter_native_splash**: Used to create the "Arva" logo splash screen when the app starts.
- **easy_localization**: If you plan to support both English and Arabic (highly relevant for a project aimed at Egyptian agriculture)