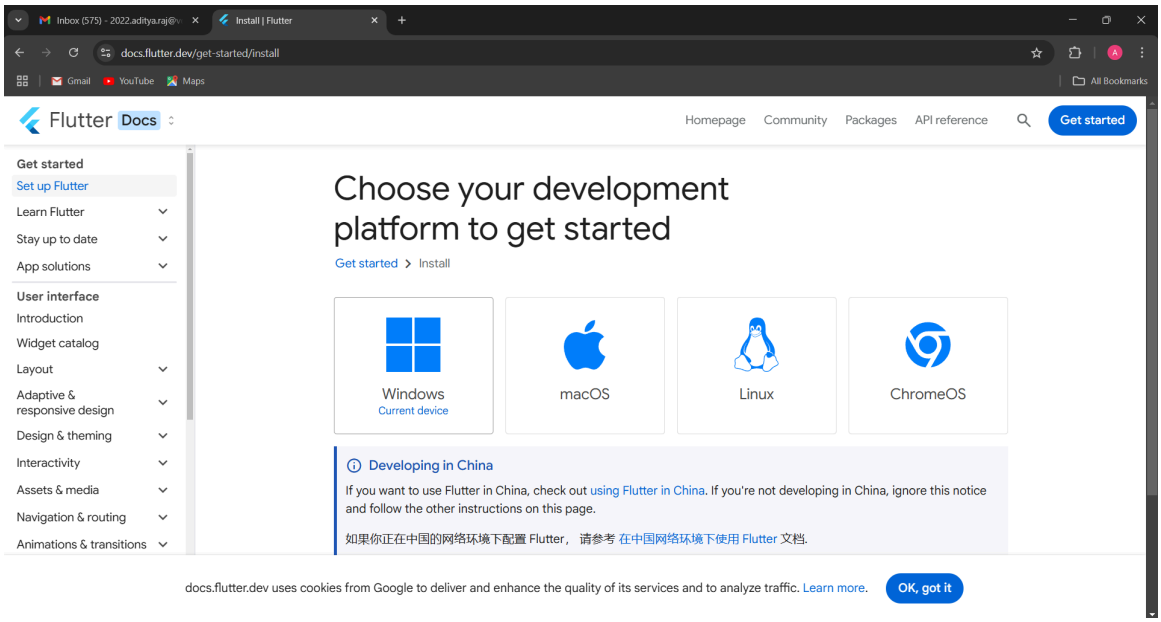
Name: Aditya Raj
Div: D15B
Roll no: 01

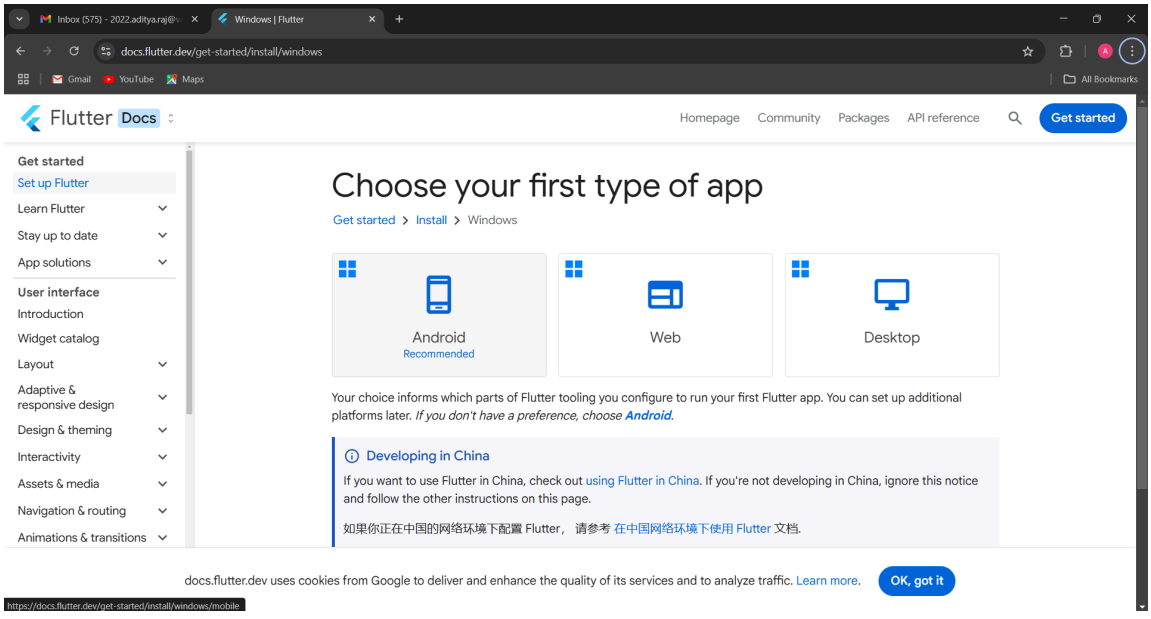
Experiment No:1

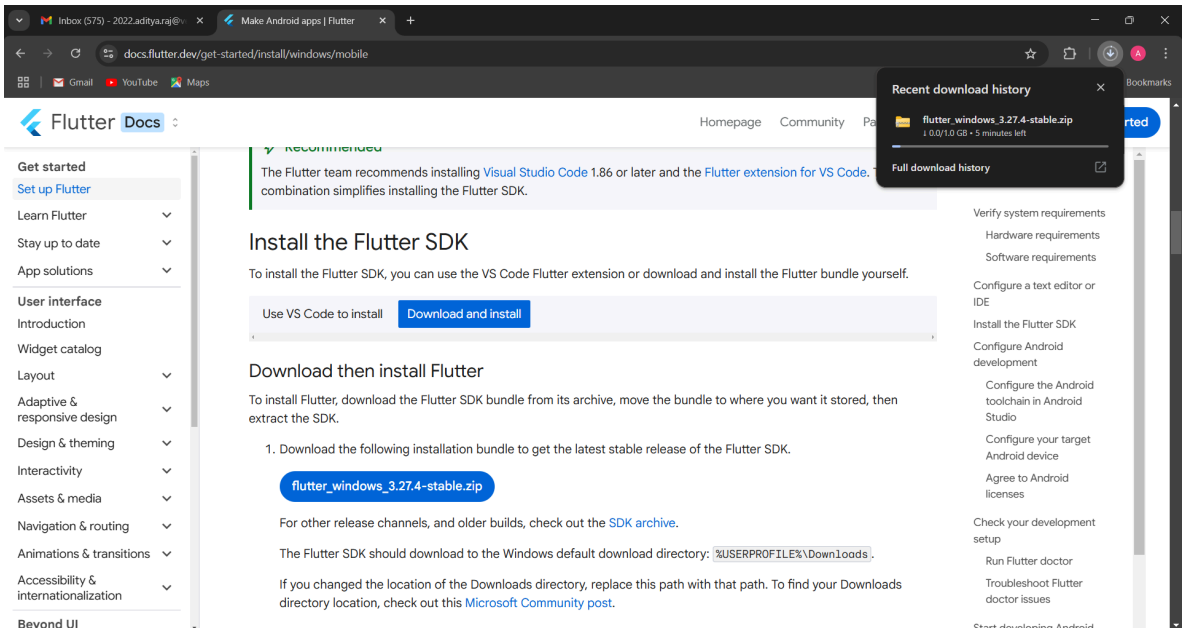
Aim:

Step 1: Download the installation bundle of the Flutter Software Development Kit for windows. To download Flutter SDK, Go to its official website https://docs.flutter.dev/get-started/install, you will get the following screen.



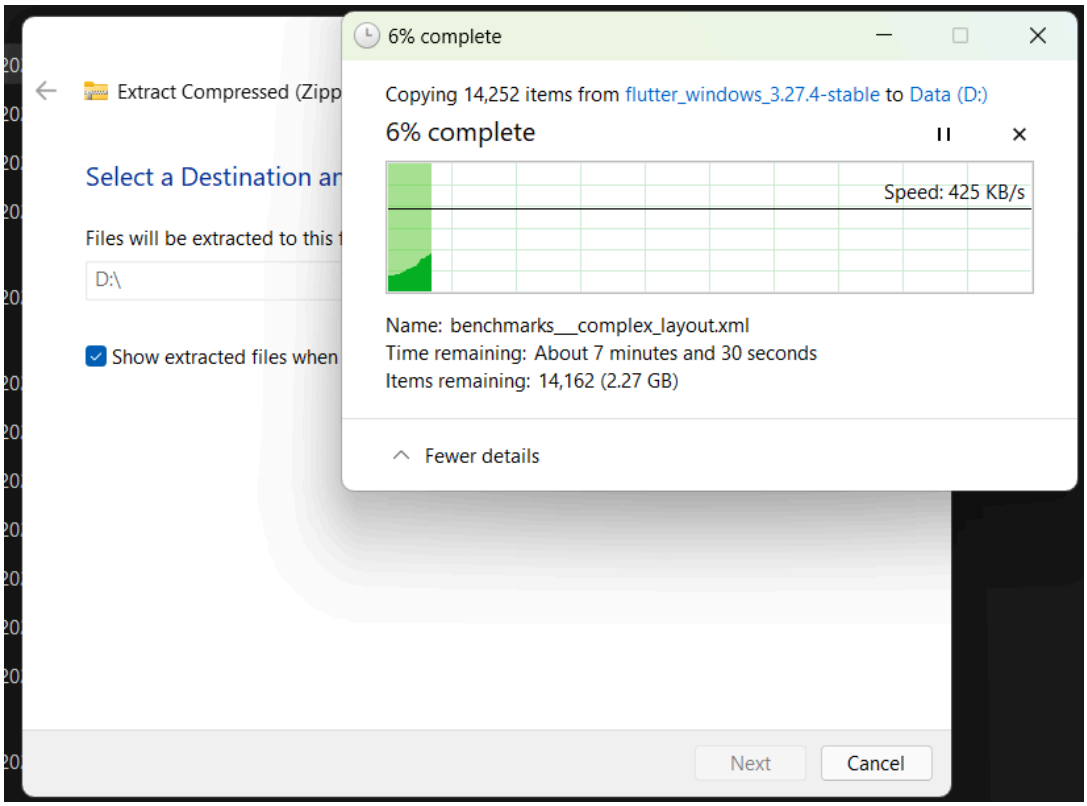
Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.



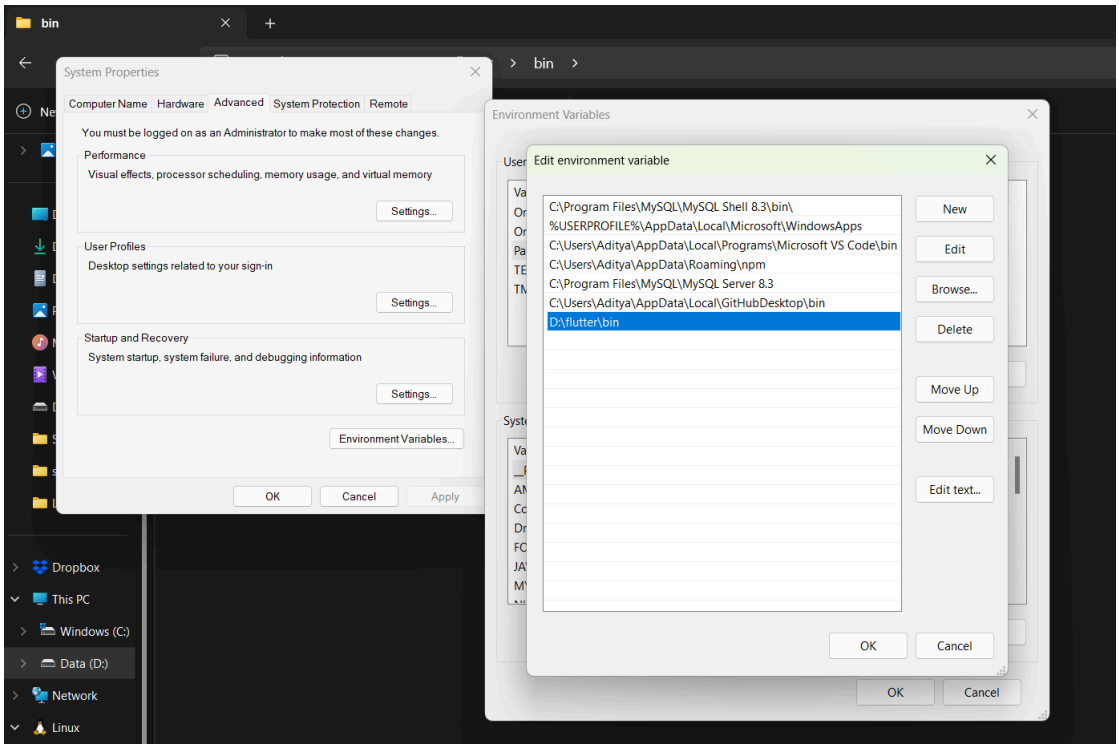


Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter

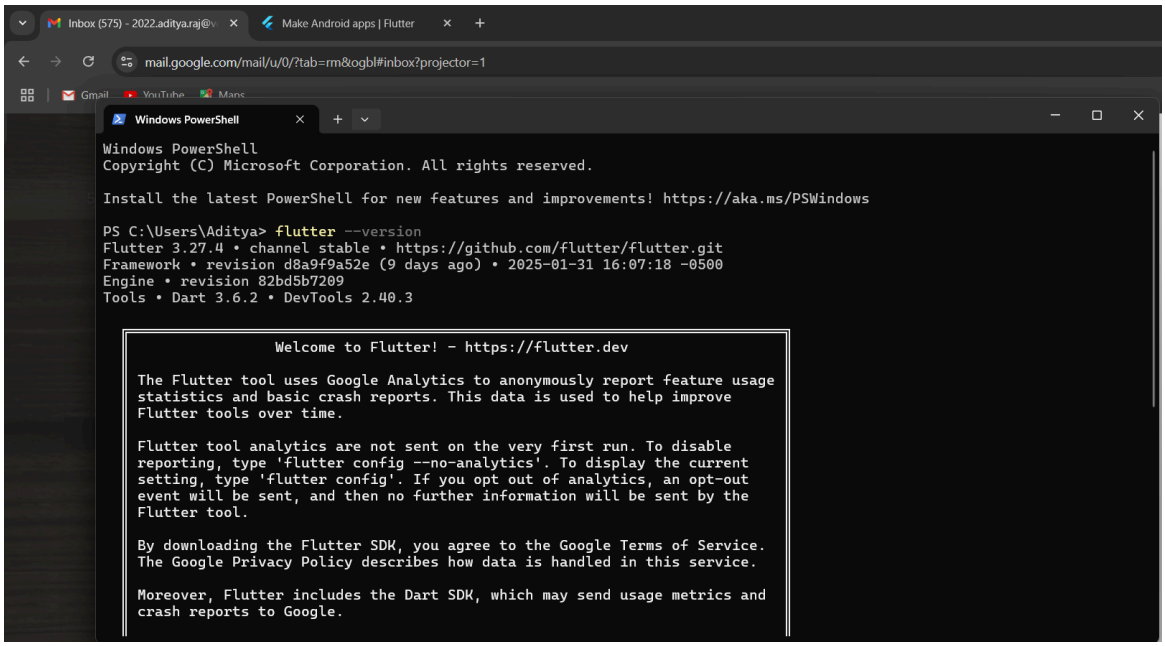
Step 4: To run the Flutter command in the regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:
Step 4.1: Go to MyComputer properties -> advanced tab -> environment variables. You will get the following screen.



Step 4.2: Now, select path -> click on edit. The following screen appears
Step 4.3: In the above window, click on New->write path of Flutter bin folder in variable value -> ok -> ok -> ok

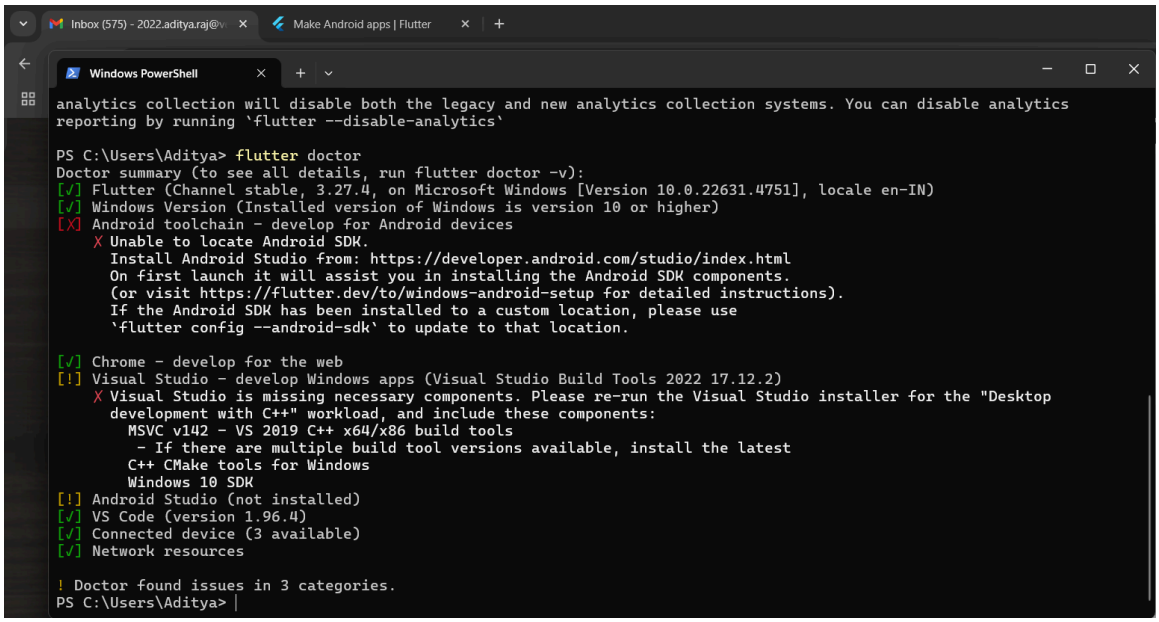


Step 5: Now, run the $ flutter command in the command prompt.



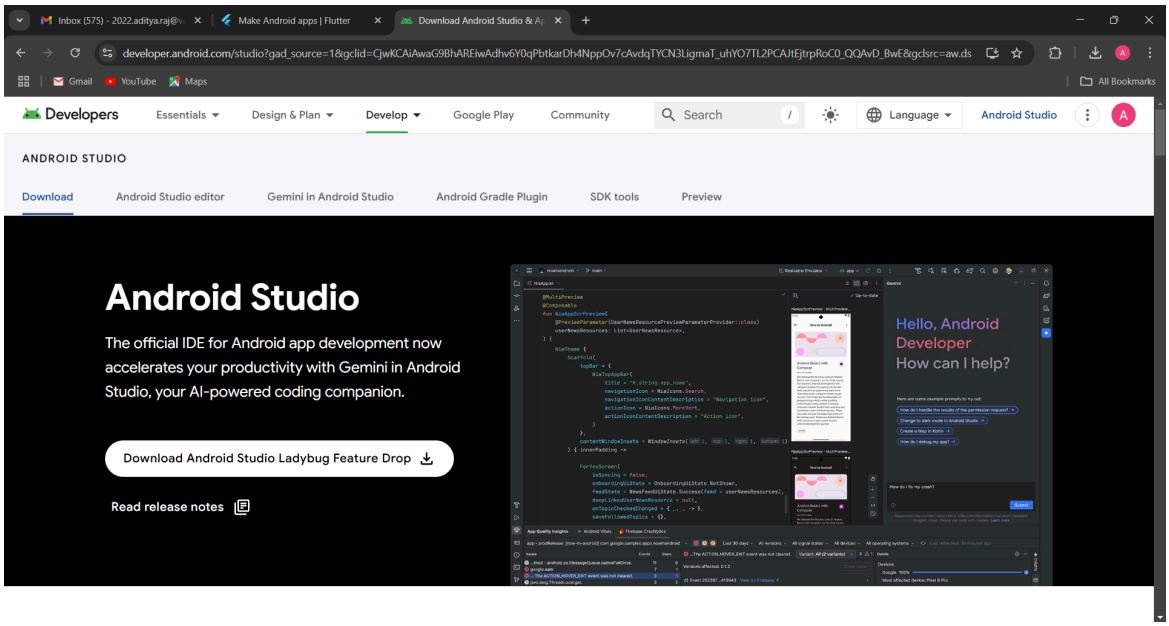
Now, run the $ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

Step 6: When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which are required to run Flutter as well as the development tools that are available but not connected with the device.

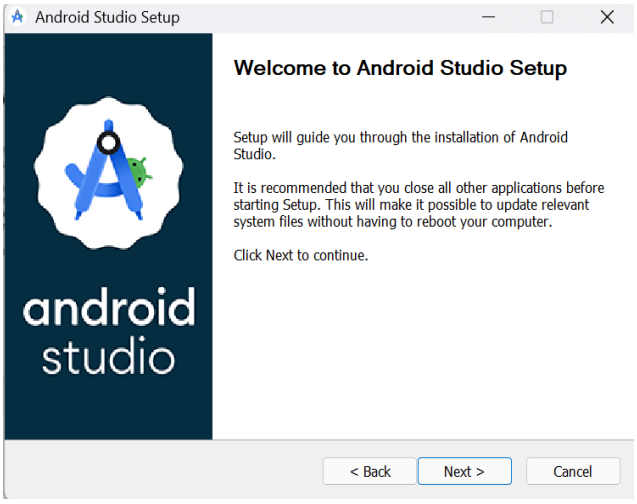


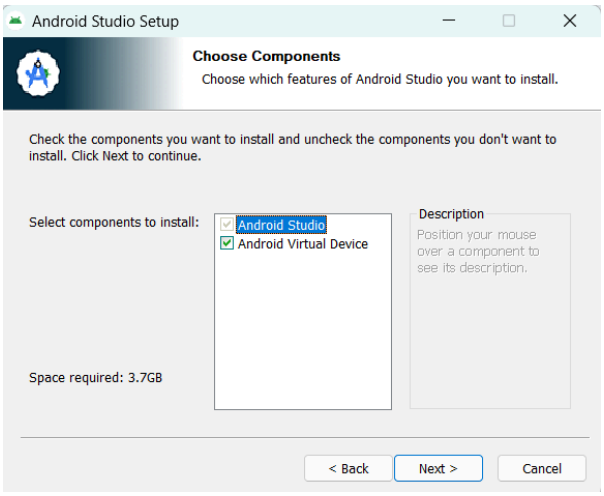
Step 7: Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

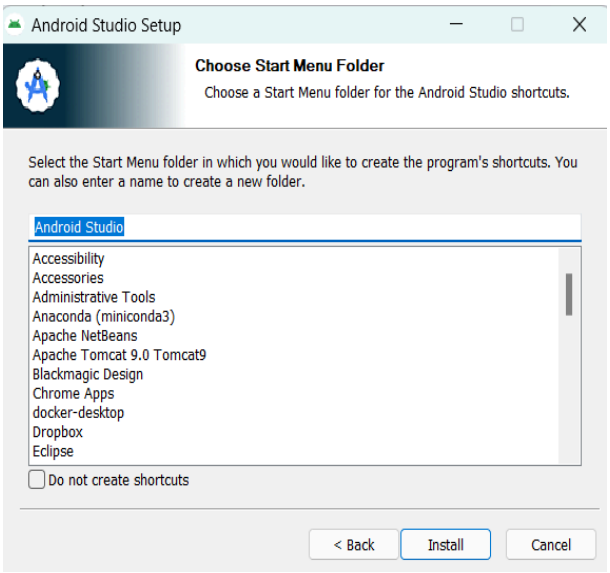
Step 7.1: Download the latest Android Studio executable or zip file from the official site.

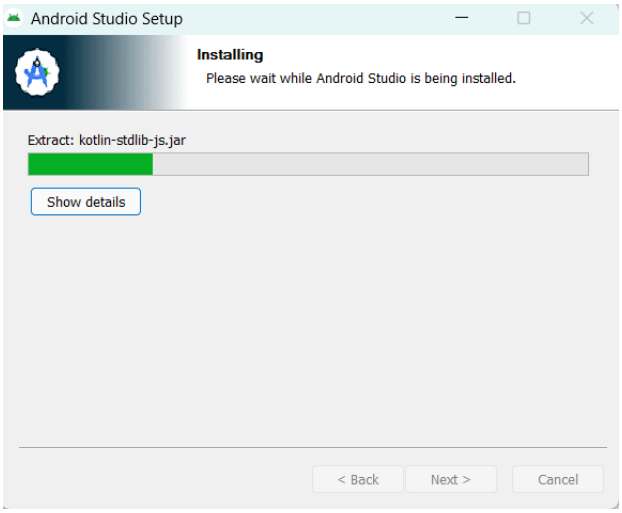


Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box





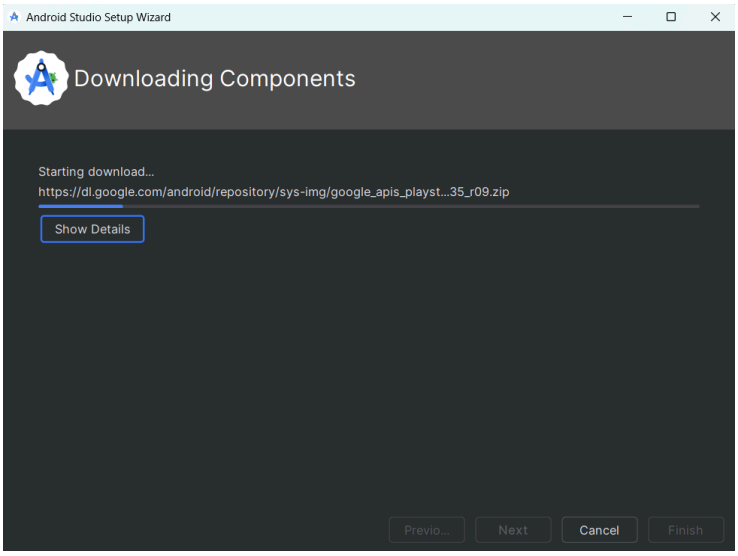


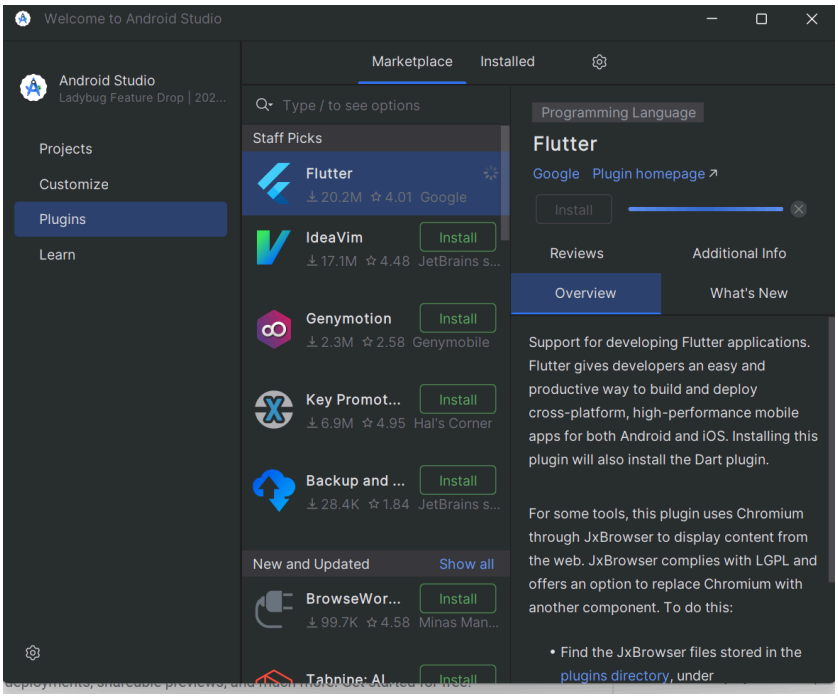


Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.

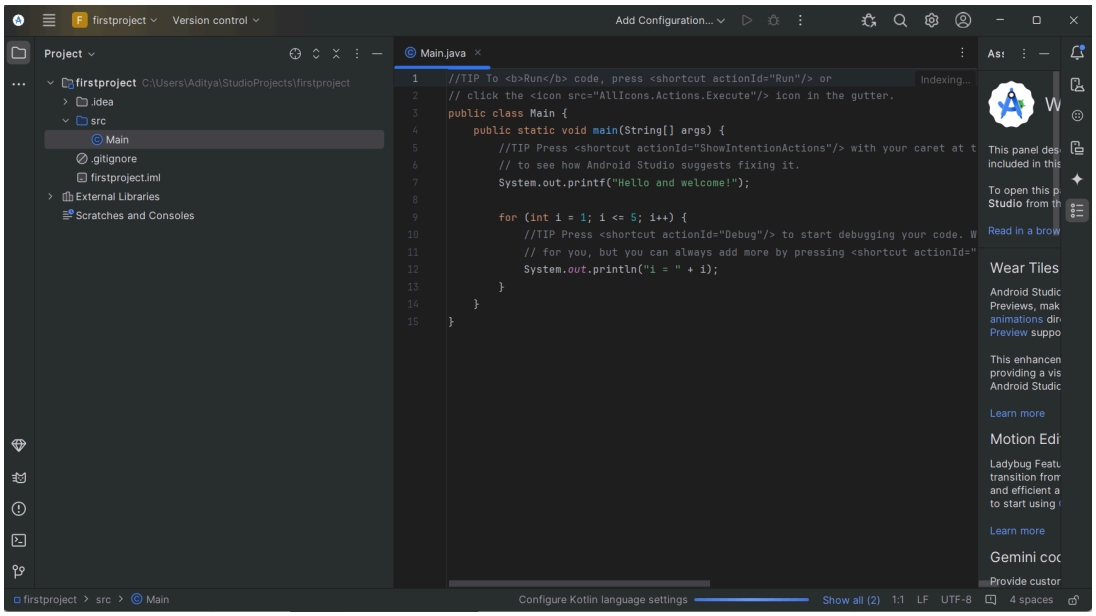
Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the 'Don't import Settings option' and click OK. It will start the Android Studio.

Step 8: Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.
Step 8.1: To set an Android emulator, go to Android Studio > Tools > Android > AVD Manager and select Create Virtual Device. Or, go to Help->Find Action->Type Emulator in the search box. You will get the following screen.

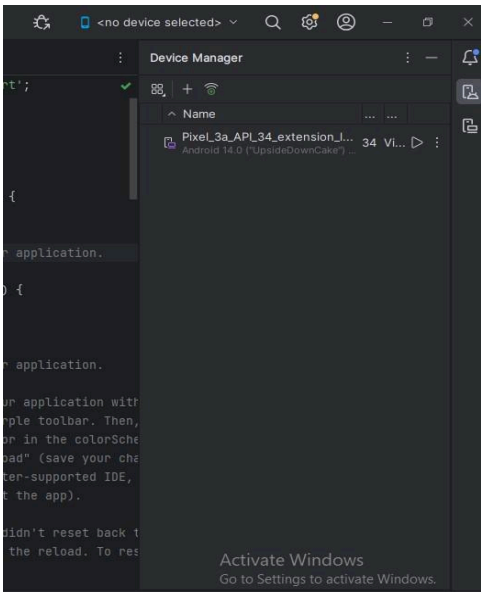




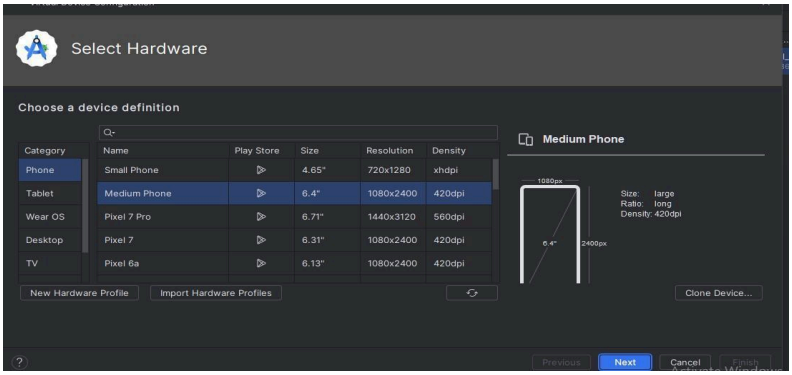
Step 8.2: Choose your device definition and click on Next.



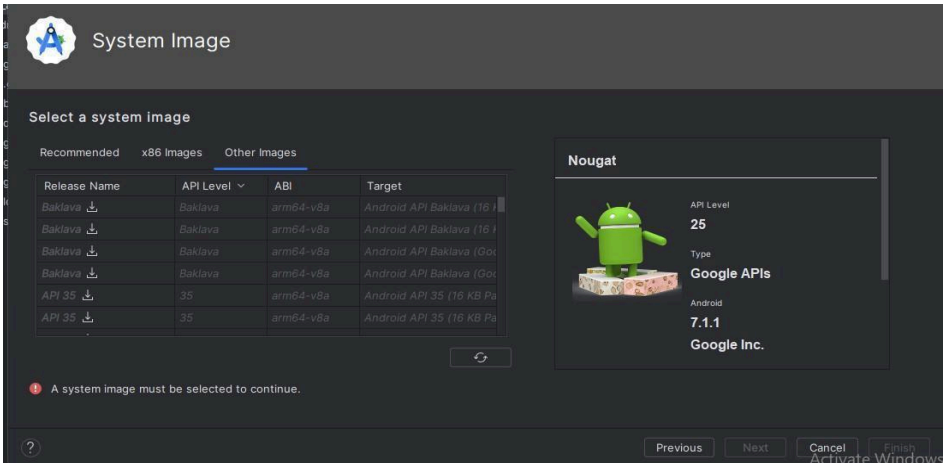
Step 8.3: Select the system image for the latest Android version and click on Next.
Step 8.4: Now, verify the all AVD configuration. If it is correct, click on Finish. The following screen appears.



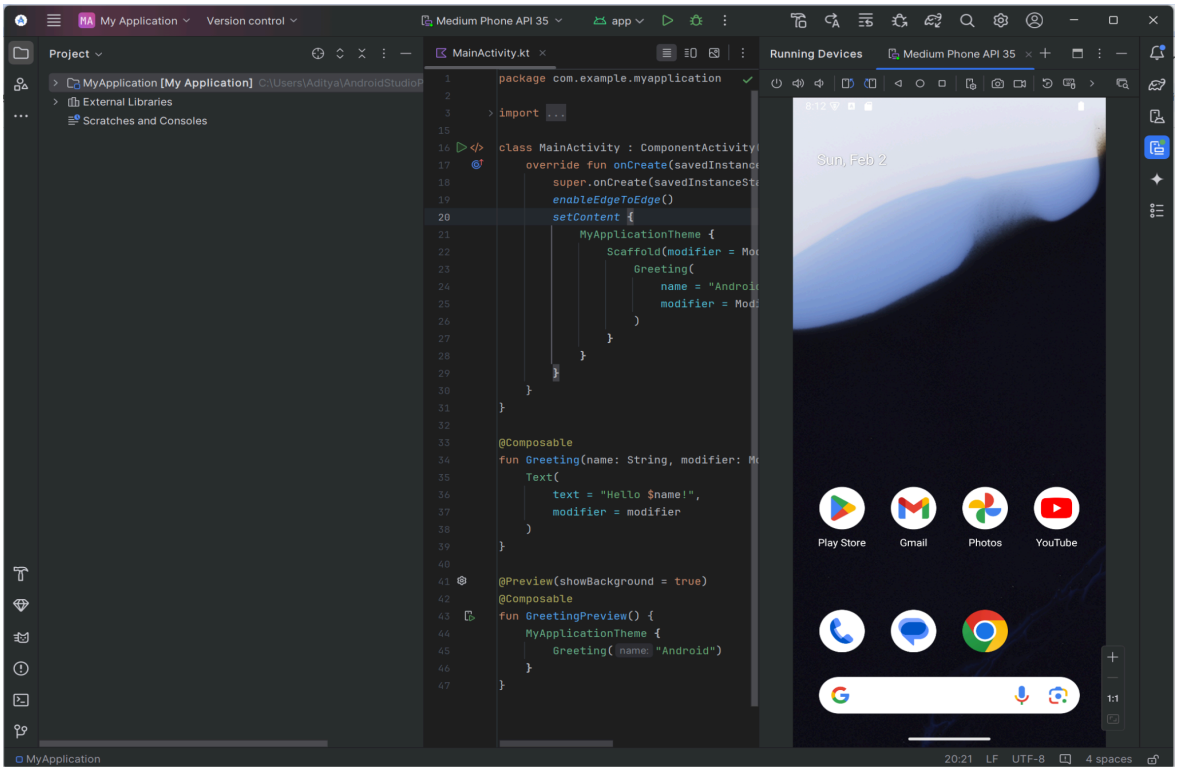
Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator



Step 9: Now, install Flutter and Dart plugin for building Flutter applications in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself. Do the following steps to install these plugins.
Step 9.1: Open the Android Studio and then go to File->Settings->Plugins.

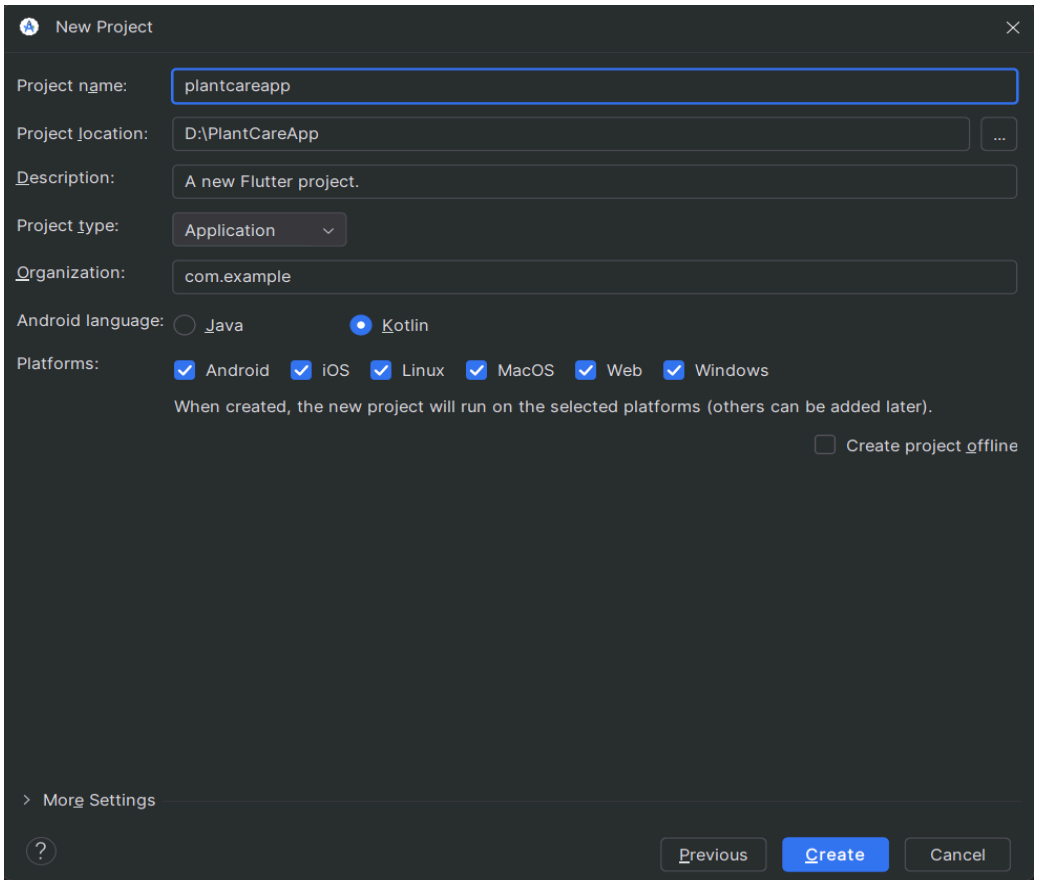


Step 9.2: Now, search the Flutter plugin. If found, select Flutter plugin and click install. When you click on install, it will ask you to install the Dart plugin as shown below screen. Click yes to proceed.
Step 9.3: Restart the Android Studio.

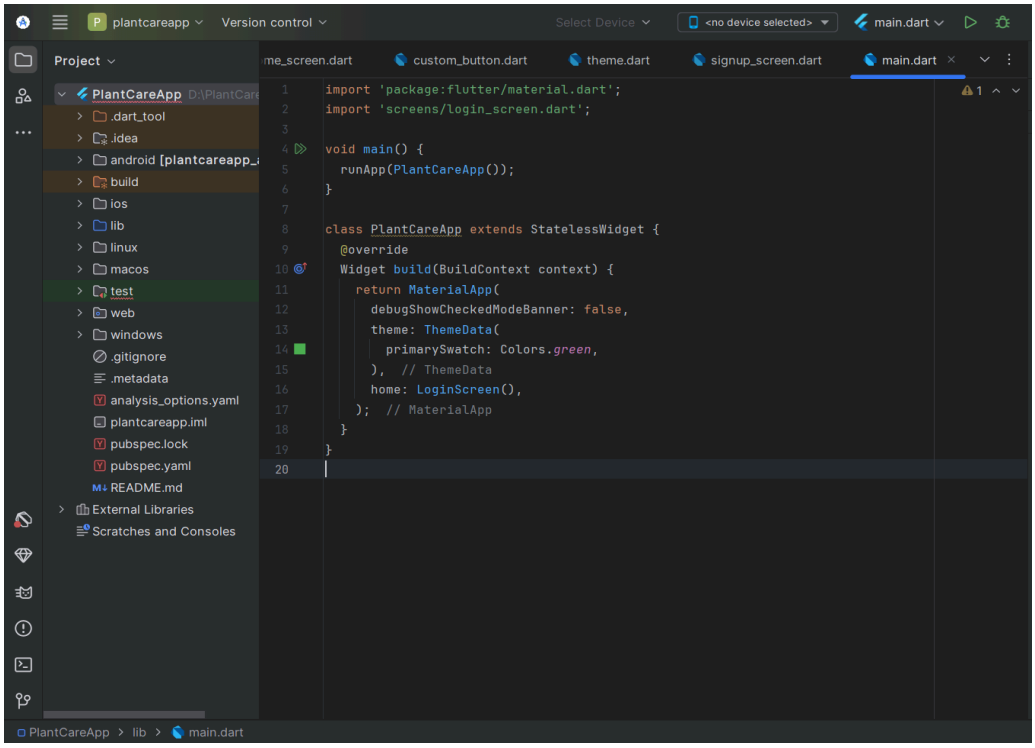


Aim : To design flutter UI by including common widgets.

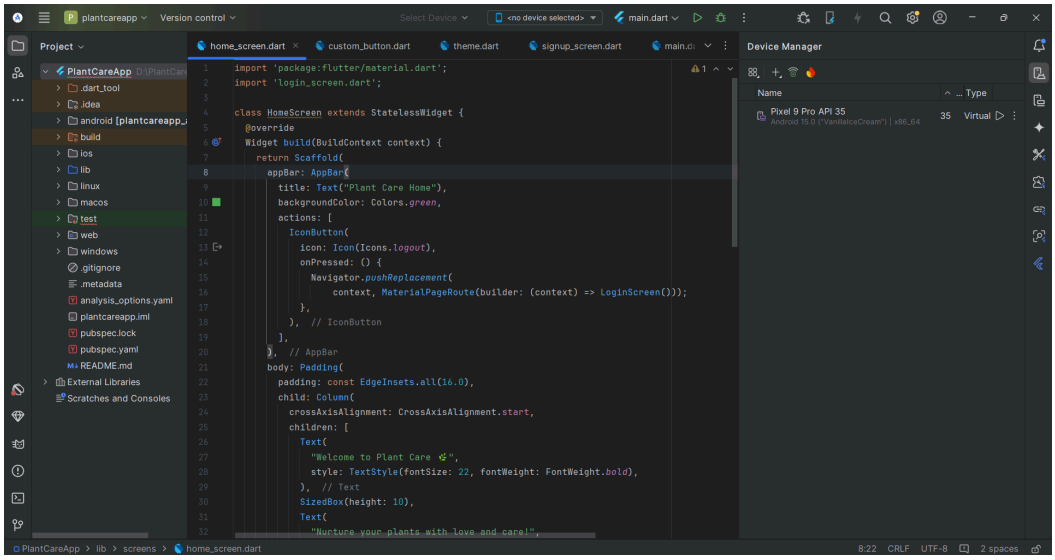
Creating a new flutter project



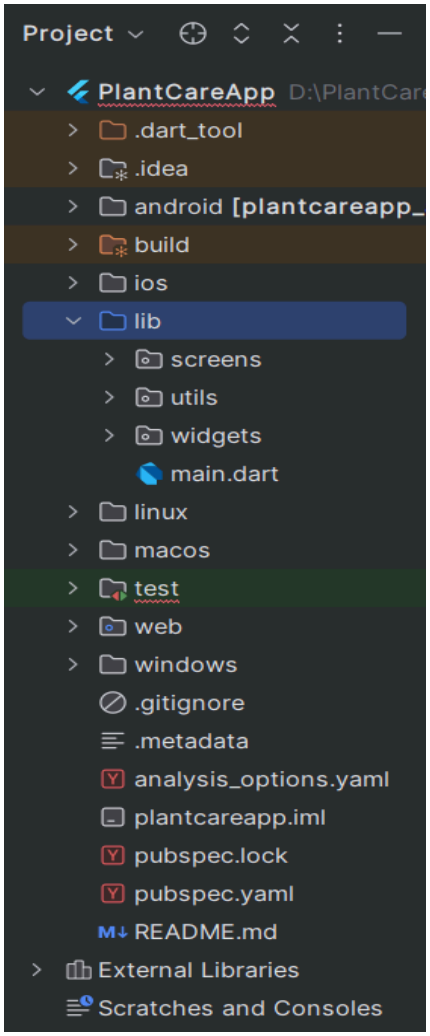
main.dart



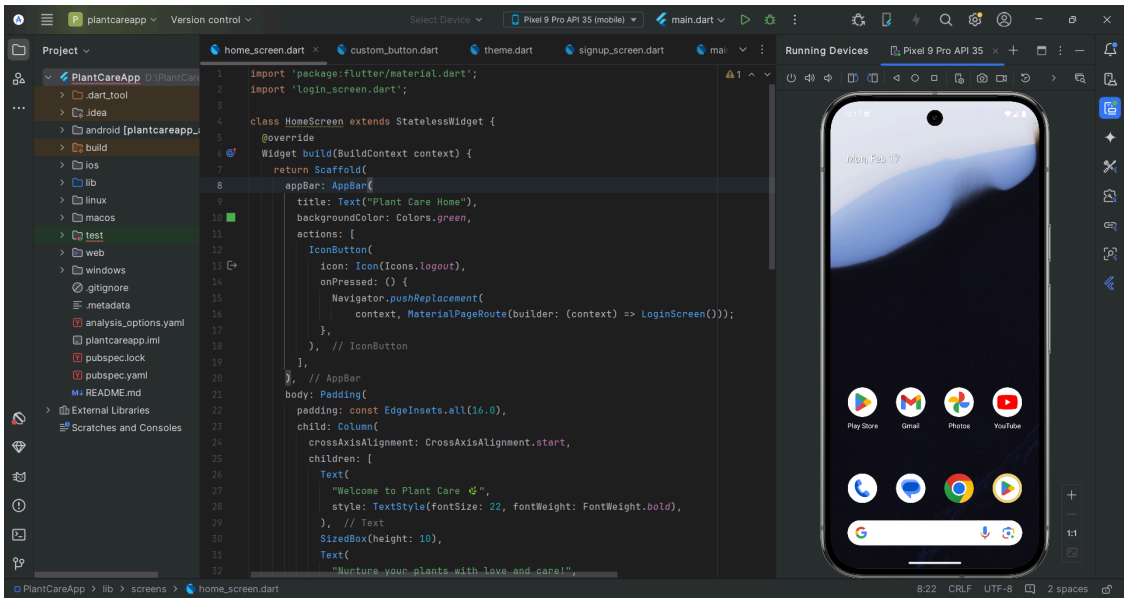
Adding virtual device: Configuring a virtual device (emulator) for testing the Flutter app.



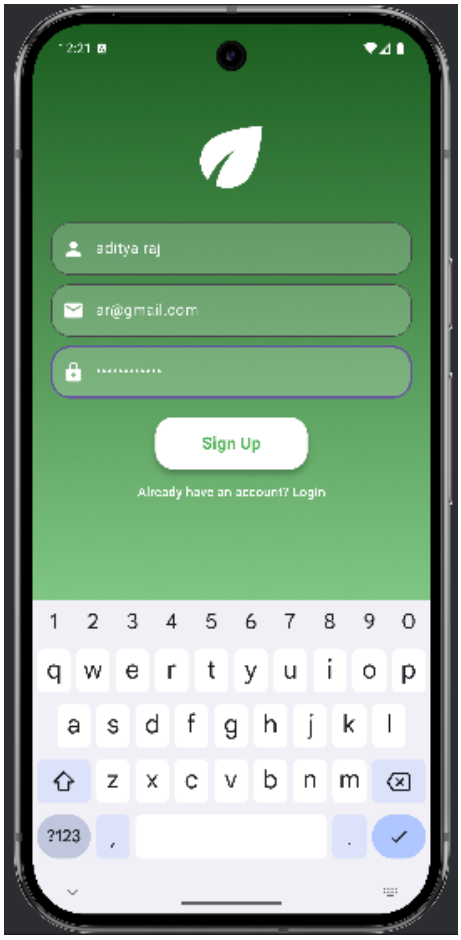
File structure: Displaying the organized directory structure of the Flutter project.



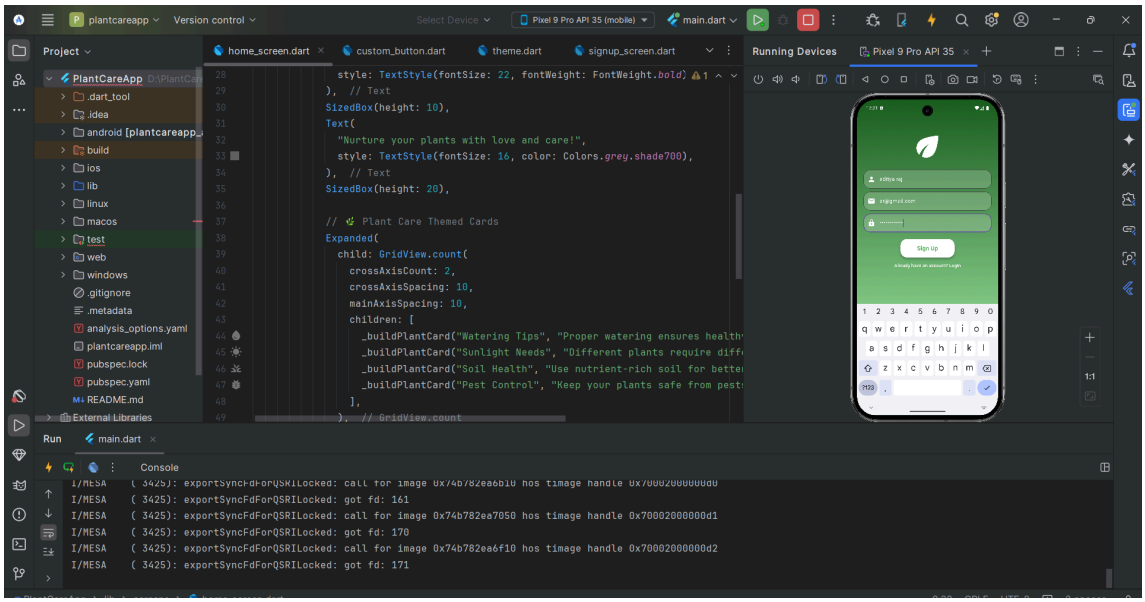
Start the emulator: Running the emulator to test and visualize the Flutter UI.



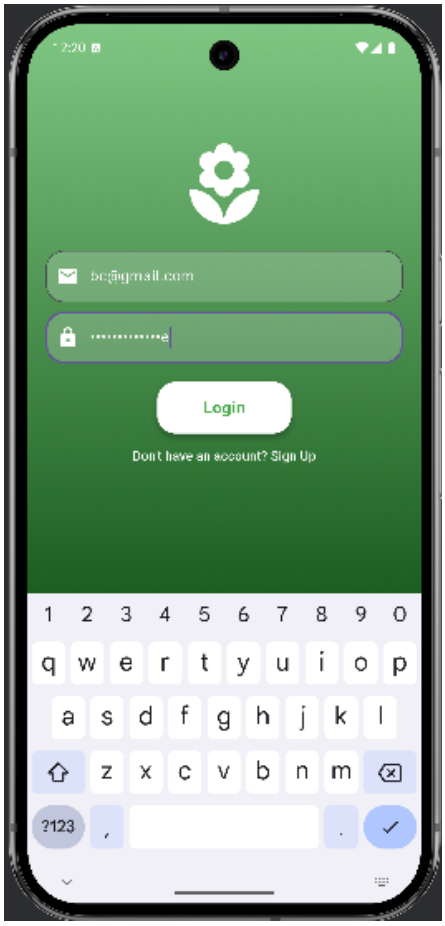
Sign-up page with widgets and buttons: User-friendly signup with text fields and buttons.



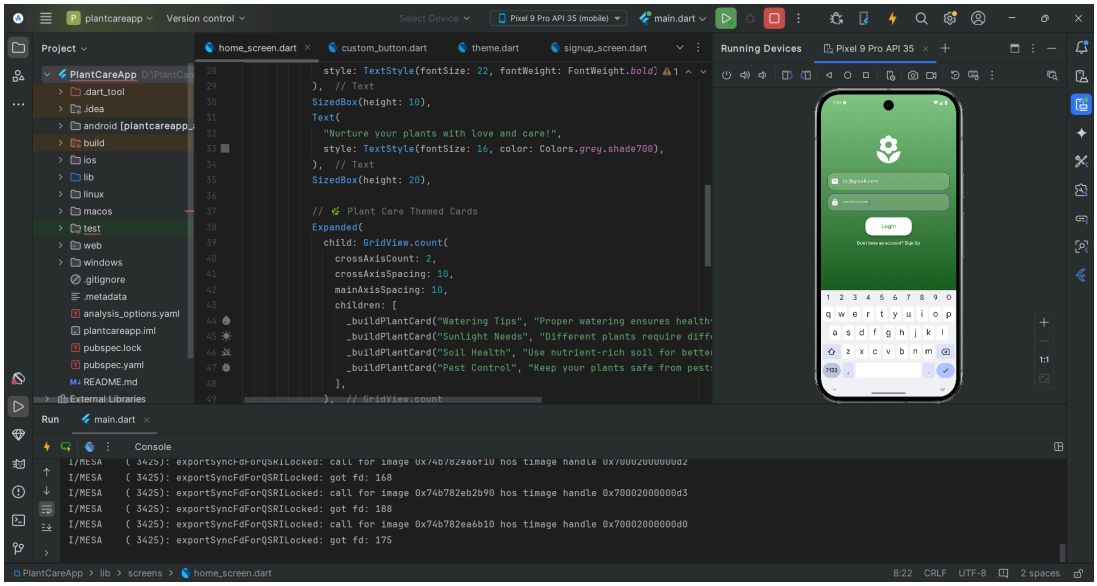
Adding icons in UI: Enhancing the interface with icons for better user experience



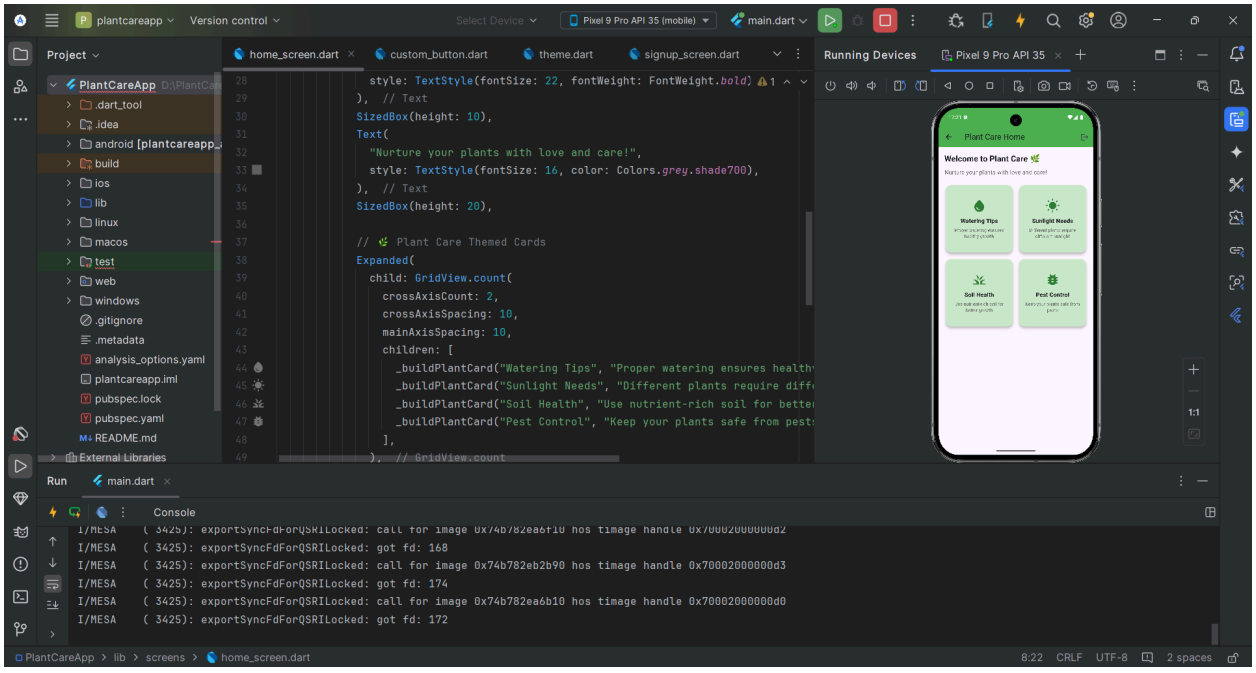
The **login page** features input fields, a submit button, and form validation, providing a seamless user authentication experience.



Login page code integrates **TextFields** for input, an **ElevatedButton** for submission, and form validation, ensuring a smooth user authentication experience.

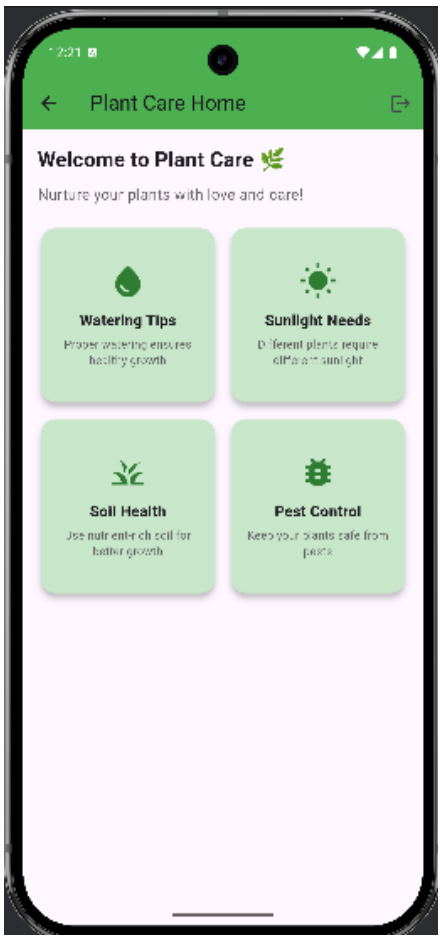


The **home page** code implements a GridView/ListView, Card widgets, and an AppBar with icons, creating an interactive and visually appealing plant care interface.



The home page of the Plant Care App features an intuitive layout with plant images, Card widgets for plant details, a search bar, and interactive buttons. It uses **GridView**/ListView for

plant display and a BottomNavigationBar for navigation with icons for a seamless user experience.



Conclusion : The experiment successfully demonstrated Flutter UI design by implementing common widgets, icons, images, and fonts. It provided hands-on experience in creating interactive and visually appealing mobile interfaces.

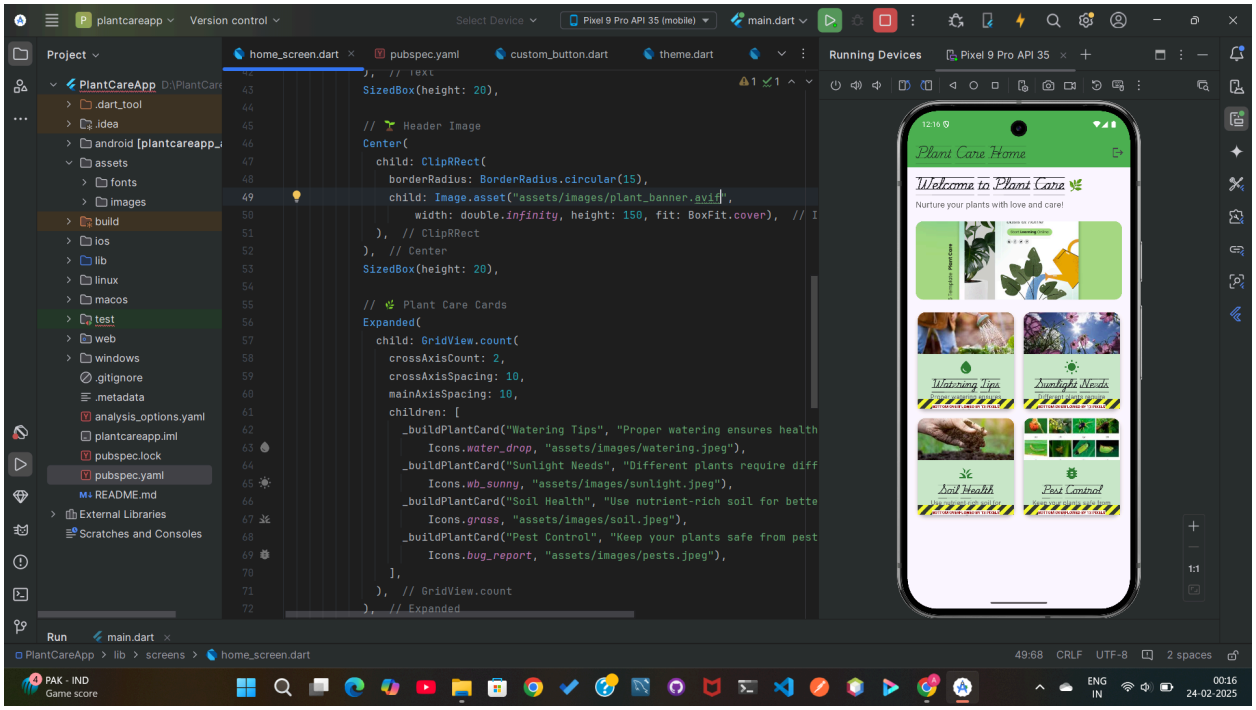
Aim : To include icons, images, fonts in Flutter app

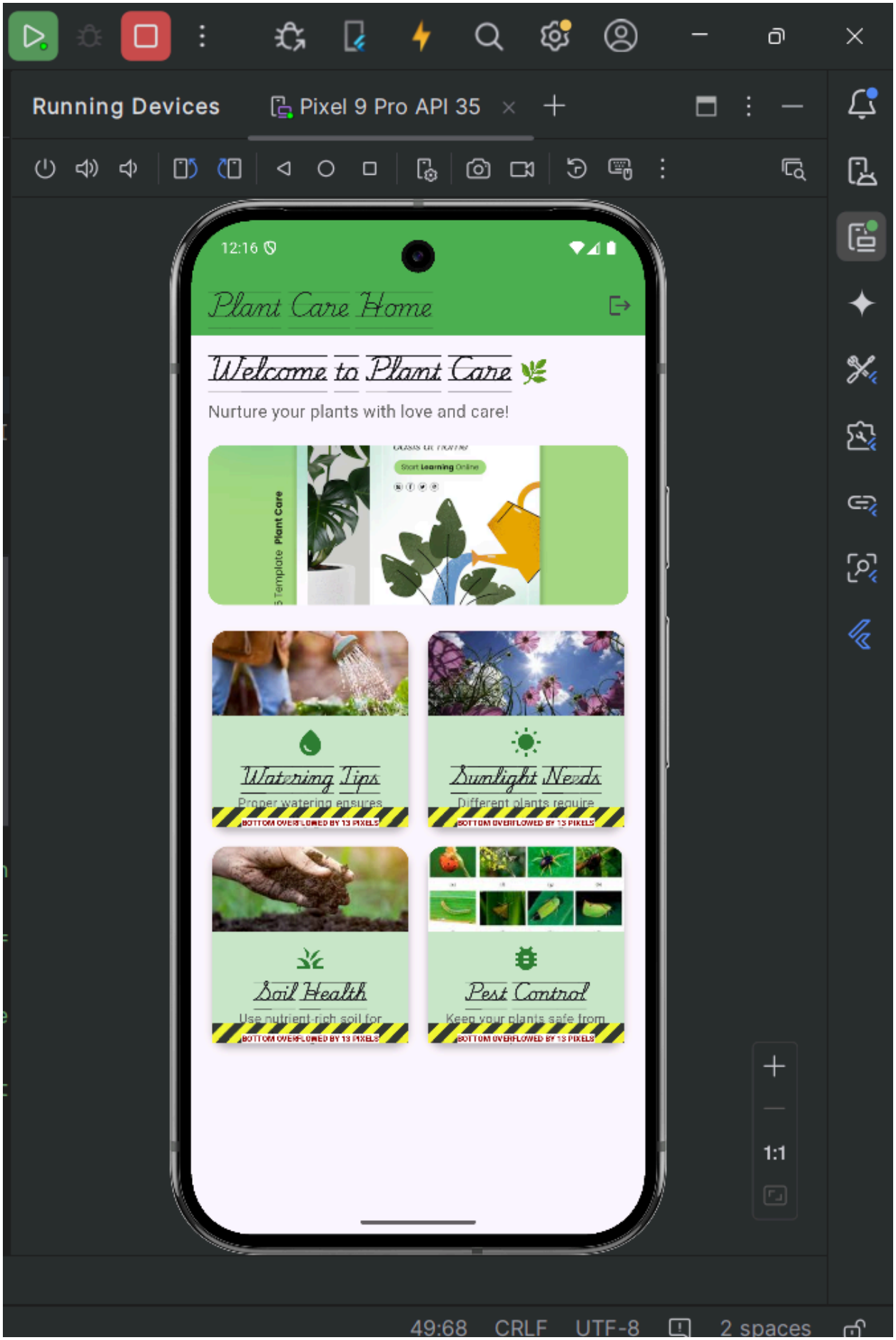
Code :

```
import 'package:flutter/material.dart';
import 'login_screen.dart';

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Plant Care Home",
          style: TextStyle(fontFamily: 'CustomFont', fontSize: 22),
        ),
        backgroundColor: Colors.green,
        actions: [
          IconButton(
            icon: Icon(Icons.logout),
            onPressed: () {
              Navigator.pushReplacement(
                  context, MaterialPageRoute(builder: (context) => LoginScreen()));
            },
          ),
        ],
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // 🌿 Welcome Section
            Text(
              "Welcome to Plant Care 🌿",
              style: TextStyle(
                fontSize: 22,
                fontWeight: FontWeight.bold,
                fontFamily: 'CustomFont',
              ),
            ),
            SizedBox(height: 10),
            Text(
              "Nurture your plants with love and care!",
```

```
            style: TextStyle(fontSize: 16, color: Colors.grey.shade700),
          ),
          SizedBox(height: 20),

          // 🌱 Header Image
          Center(
            child: ClipRRect(
              borderRadius: BorderRadius.circular(15),
              child: Image.asset("assets/images/plant_banner.avif",
                  width: double.infinity, height: 150, fit: BoxFit.cover),
            ),
          ),
          SizedBox(height: 20),

          // 🌿 Plant Care Cards
          Expanded(
            child: GridView.count(
              crossAxisCount: 2,
              crossAxisSpacing: 10,
              mainAxisSpacing: 10,
              children: [
                _buildPlantCard("Watering Tips", "Proper watering ensures healthy growth",
                    Icons.water_drop, "assets/images/watering.jpeg"),
                _buildPlantCard("Sunlight Needs", "Different plants require different sunlight",
                    Icons.wb_sunny, "assets/images/sunlight.jpeg"),
                _buildPlantCard("Soil Health", "Use nutrient-rich soil for better growth",
                    Icons.grass, "assets/images/soil.jpeg"),
                _buildPlantCard("Pest Control", "Keep your plants safe from pests",
                    Icons.bug_report, "assets/images/pests.jpeg"),
              ],
            ),
          ),
        ],
      ),
    ),
  );
}

//  Reusable Plant Info Card with Image
Widget _buildPlantCard(String title, String description, IconData icon, String imagePath) {
  return Card(
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
    elevation: 5,
    color: Colors.green.shade100,
```

```
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          ClipRRect(
            borderRadius: BorderRadius.vertical(top: Radius.circular(15)),
            child: Image.asset(imagePath, height: 80, width: double.infinity, fit: BoxFit.cover),
          ),
          Padding(
            padding: EdgeInsets.all(10),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Icon(icon, size: 30, color: Colors.green.shade800),
                SizedBox(height: 5),
                Text(title,
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold, fontFamily:
'CustomFont')),
                SizedBox(height: 5),
                Text(description,
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 12, color: Colors.grey.shade700)),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```


plantcareapp
Version control
Pixel 9 Pro API 35 (mobile)
main.dart
Project
PlantCareApp
home_screen.dart
pubspec.yaml
custom_button.dart
theme.dart
Running Devices
Pixel 9 Pro API 35
SizedBox(height: 20),
// Header Image
Center(
child: ClipRRect(
borderRadius: BorderRadius.circular(15),
child: Image.asset("assets/images/plant_banner.avif",
width: double.infinity, height: 150, fit: BoxFit.cover),
), // ClipRRect
), // Center
SizedBox(height: 20),
// Plant Care Cards
Expanded(
child: GridView.count(
crossAxisCount: 2,
crossAxisSpacing: 10,
mainAxisSpacing: 10,
children: [
_buildPlantCard("Watering Tips", "Proper watering ensures health
Icons.water_drop, "assets/images/watering.jpeg"),
_buildPlantCard("Sunlight Needs", "Different plants require diff
Icons.wb_sunny, "assets/images/sunlight.jpeg"),
_buildPlantCard("Soil Health", "Use nutrient-rich soil for bette
Icons.grass, "assets/images/soil.jpeg"),
_buildPlantCard("Pest Control", "Keep your plants safe from pest
Icons.bug_report, "assets/images/pests.jpeg"),
],
), // GridView.count
), // Expanded
Plant Care Home
Welcome to Plant Care
Nurture your plants with love and care!
Watering Tips
Sunlight Needs
Soil Health
Pest Control
Run
main.dart
PlantCareApp > lib > screens > home_screen.dart
49:68 CRLF UTF-8 2 spaces
PAK - IND
Game score
ENG
IN
00:16
24-02-2025

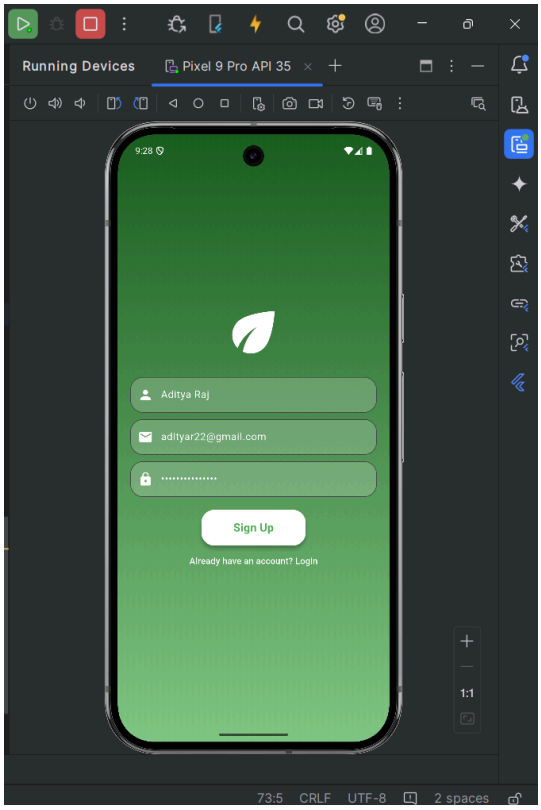

Running Devices
Pixel 9 Pro API 35
12:16
Plant Care Home
Welcome to Plant Care
Nurture your plants with love and care!
Watering Tips
Proper watering ensures
BOTTOM OVERFLOWED BY 13 PIXELS
Sunlight Needs
Different plants require
BOTTOM OVERFLOWED BY 13 PIXELS
Soil Health
Use nutrient-rich soil for
BOTTOM OVERFLOWED BY 13 PIXELS
Pest Control
Keep your plants safe from
BOTTOM OVERFLOWED BY 13 PIXELS
1:1
49:68
CRLF
UTF-8
2 spaces

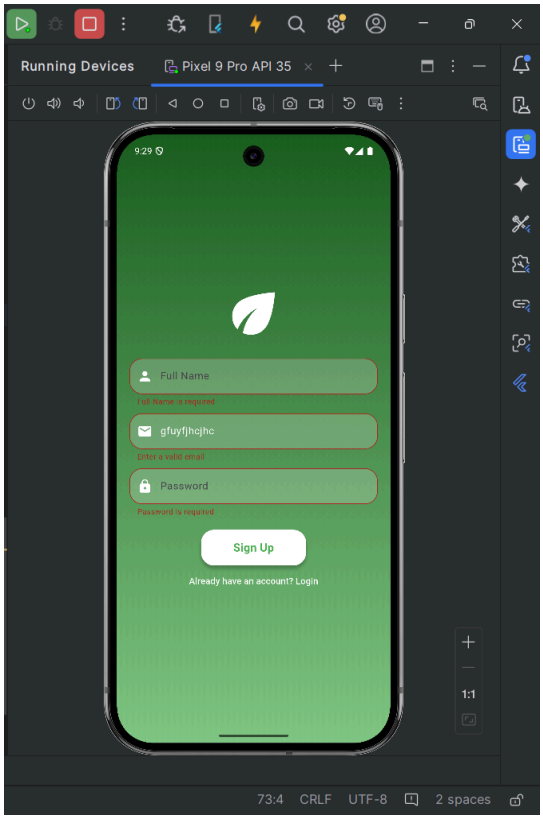
Aim : To create an interactive Form using form widget

Signup_screen.dart

```
import 'package:flutter/material.dart';
import 'login_screen.dart';
import 'home_screen.dart';
import '../widgets/custom_button.dart';

class SignupScreen extends StatelessWidget {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController nameController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: EdgeInsets.all(20),
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.green.shade900, Colors.green.shade300],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisSize: MainAxisSize.min,
                children: [
                  Icon(Icons.eco, size: 100, color: Colors.white),
                  SizedBox(height: 20),
                  _buildTextField(nameController, "Full Name", Icons.person, validator: _validateName),
                  SizedBox(height: 10),
                  _buildTextField(emailController, "Email", Icons.email, validator: _validateEmail),
                  SizedBox(height: 10),
                  _buildTextField(passwordController, "Password", Icons.lock, obscureText: true, validator:
_validatePassword),
                  SizedBox(height: 20),
                  CustomButton(
                    text: "Sign Up",
                    onPressed: () {
                      if (_formKey.currentState!.validate()) {
                        Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) =>
HomeScreen()));
                      }
                    },
                  ),
                  TextButton(
```

```
                onPressed: () {
                  Navigator.pop(context);
                },
                child: Text("Already have an account? Login", style: TextStyle(color: Colors.white)),
              ),
            ],
          ),
        ),
      ),
    ),
  );
 }

  Widget _buildTextField(TextEditingController controller, String hint, IconData icon, {bool obscureText =
false, String? Function(String?)? validator}) {
    return TextFormField(
      controller: controller,
      obscureText: obscureText,
      validator: validator,
      decoration: InputDecoration(
        hintText: hint,
        prefixIcon: Icon(icon, color: Colors.white),
        filled: true,
        fillColor: Colors.white.withOpacity(0.2),
        border: OutlineInputBorder(borderRadius: BorderRadius.circular(20)),
      ),
      style: TextStyle(color: Colors.white),
    );
  }

  String? _validateName(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "Full Name is required";
    }
    return null;
  }

  String? _validateEmail(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "Email is required";
    }
    final emailRegex = RegExp(r'^[^@]+@[^@]+\.[^@]+$');
    if (!emailRegex.hasMatch(value)) {
      return "Enter a valid email";
    }
    return null;
  }

  String? _validatePassword(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "Password is required";
    }
    return null;
  }
}
```


Running Devices
Pixel 9 Pro API 35
9:28
Aditya Raj
adityar22@gmail.com
Sign Up
Already have an account? Login
1:1
73:5 CRLF UTF-8 2 spaces

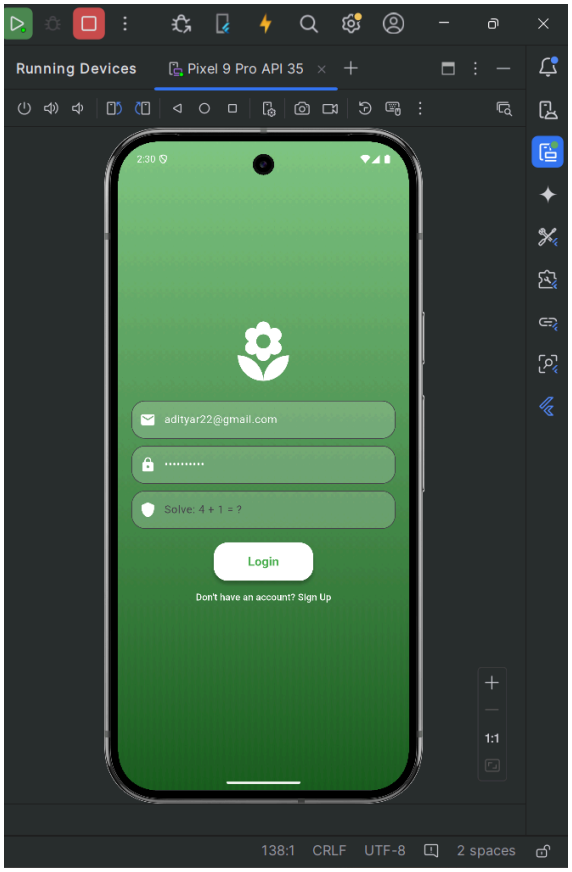

Running Devices
Pixel 9 Pro API 35
9:29
Full Name
Full Name is required
gfuyfjhcjhc
Enter a valid email
Password
Password is required
Sign Up
Already have an account? Login
1:1
73:4 CRLF UTF-8 2 spaces

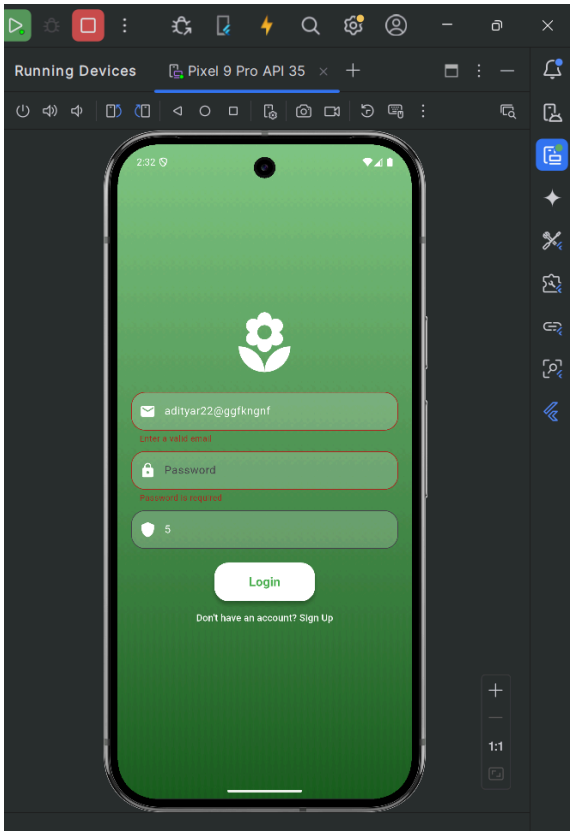
Login_screen.dart

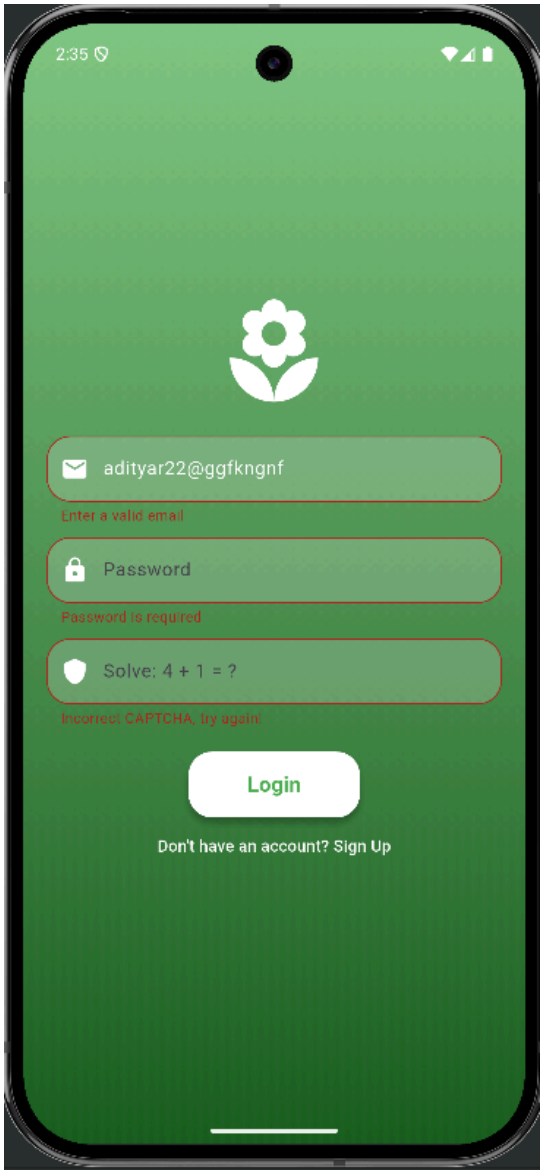
```
import 'package:flutter/material.dart';
import 'dart:math'; // Used for generating random numbers
import 'signup_screen.dart';
import 'home_screen.dart';
import '../widgets/custom_button.dart';

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController captchaController = TextEditingController();

  int _num1 = 0;
  int _num2 = 0;
  int _captchaAnswer = 0;

  @override
  void initState() {
    super.initState();
    _generateCaptcha();
  }

  void _generateCaptcha() {
    final Random random = Random();
    _num1 = random.nextInt(9) + 1; // Random number between 1-9
    _num2 = random.nextInt(9) + 1; // Random number between 1-9
    _captchaAnswer = _num1 + _num2; // Expected answer
    setState(() {}); // Refresh UI with new CAPTCHA
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: EdgeInsets.all(20),
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.green.shade300, Colors.green.shade900],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            child: Form(
              key: _formKey,
```

```
            child: Column(
              mainAxisSize: MainAxisSize.min,
              children: [
                Icon(Icons.local_florist, size: 100, color: Colors.white),
                SizedBox(height: 20),
                _buildTextField(emailController, "Email", Icons.email, validator: _validateEmail),
                SizedBox(height: 10),
                _buildTextField(passwordController, "Password", Icons.lock, obscureText: true, validator:
_validatePassword),
                SizedBox(height: 10),

                // CAPTCHA Field
                _buildTextField(
                  captchaController,
                  "Solve: $_num1 + $_num2 = ?",
                  Icons.shield,
                  validator: _validateCaptcha,
                ),

                SizedBox(height: 20),
                CustomButton(
                  text: "Login",
                  onPressed: () {
                    if (_formKey.currentState!.validate()) {
                      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) =>
HomeScreen()));
                    }
                  },
                ),
                TextButton(
                  onPressed: () {
                    Navigator.push(context, MaterialPageRoute(builder: (context) => SignupScreen()));
                  },
                  child: Text("Don't have an account? Sign Up", style: TextStyle(color: Colors.white)),
                ),
              ],
            ),
          ),
        ),
      ),
    ),
  );
 }

  Widget _buildTextField(TextEditingController controller, String hint, IconData icon, {bool obscureText =
false, String? Function(String?)? validator}) {
    return TextFormField(
      controller: controller,
      obscureText: obscureText,
      validator: validator,
      decoration: InputDecoration(
        hintText: hint,
        prefixIcon: Icon(icon, color: Colors.white),
        filled: true,
        fillColor: Colors.white.withOpacity(0.2),
        border: OutlineInputBorder(borderRadius: BorderRadius.circular(20)),
```

```
      ),
      style: TextStyle(color: Colors.white),
    );
  }

  String? _validateEmail(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "Email is required";
    }
    final emailRegex = RegExp(r'^[^@]+@[^@]+\.[^@]+$');
    if (!emailRegex.hasMatch(value)) {
      return "Enter a valid email";
    }
    return null;
  }

  String? _validatePassword(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "Password is required";
    }
    return null;
  }

  String? _validateCaptcha(String? value) {
    if (value == null || value.trim().isEmpty) {
      return "CAPTCHA is required";
    }
    if (int.tryParse(value) != _captchaAnswer) {
      return "Incorrect CAPTCHA, try again!";
    }
    return null;
  }
}
```


Running Devices
Pixel 9 Pro API 35
2:30
adityar22@gmail.com
Solve: 4 + 1 = ?
Login
Don't have an account? Sign Up
1:1
138:1
CRLF
UTF-8
2 spaces


Running Devices
Pixel 9 Pro API 35
2:32
adityar22@ggfkngnf
Enter a valid email
Password
Password is required
5
Login
Don't have an account? Sign Up
1:1


2:35
adityar22@ggfkngnf
Enter a valid email
Password
Password is required
Solve: 4 + 1 = ?
Incorrect CAPTCHA, try again
Login
Don't have an account? Sign Up

main.dart

```
import 'package:flutter/material.dart';
import 'screens/login_screen.dart';
import 'screens/home_screen.dart';
import 'screens/details_screen.dart';

void main() {
  runApp(PlantCareApp());
}

class PlantCareApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => LoginScreen(),
        '/home': (context) => HomeScreen(),
        '/details': (context) => DetailsScreen(),
      },
    );
  }
}
```

home_screen.dart

```
import 'package:flutter/material.dart';

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Plant Care Home",
          style: TextStyle(fontFamily: 'CustomFont', fontSize: 22),
        ),
        backgroundColor: Colors.green,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
```

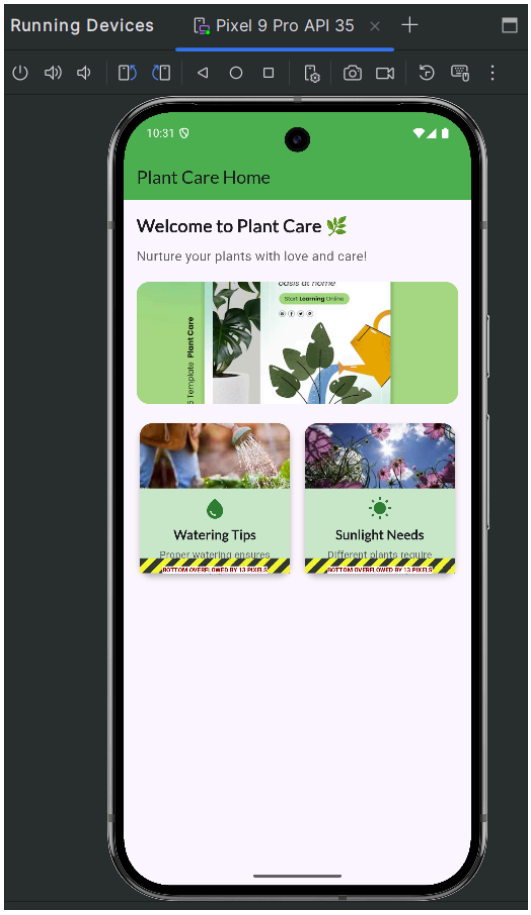
```
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            "Welcome to Plant Care 🌿",
            style: TextStyle(
              fontSize: 22,
              fontWeight: FontWeight.bold,
              fontFamily: 'CustomFont',
            ),
          ),
          SizedBox(height: 10),
          Text(
            "Nurture your plants with love and care!",
            style: TextStyle(fontSize: 16, color: Colors.grey.shade700),
          ),
          SizedBox(height: 20),

          Center(
            child: GestureDetector(
              onTap: () {
                Navigator.pushNamed(context, '/details');
              },
              child: ClipRRect(
                borderRadius: BorderRadius.circular(15),
                child: Image.asset("assets/images/plant_banner.avif",
                    width: double.infinity, height: 150, fit: BoxFit.cover),
              ),
            ),
          ),
          SizedBox(height: 20),

          Expanded(
            child: GridView.count(
              crossAxisCount: 2,
              crossAxisSpacing: 10,
              mainAxisSpacing: 10,
              children: [
                _buildPlantCard(context, "Watering Tips", "Proper watering ensures healthy growth",
                    Icons.water_drop, "assets/images/watering.jpeg"),
                _buildPlantCard(context, "Sunlight Needs", "Different plants require different
sunlight",
                    Icons.wb_sunny, "assets/images/sunlight.jpeg"),
              ],
            ),
          ),
        ],
      ),
    ),
  );
 }
```

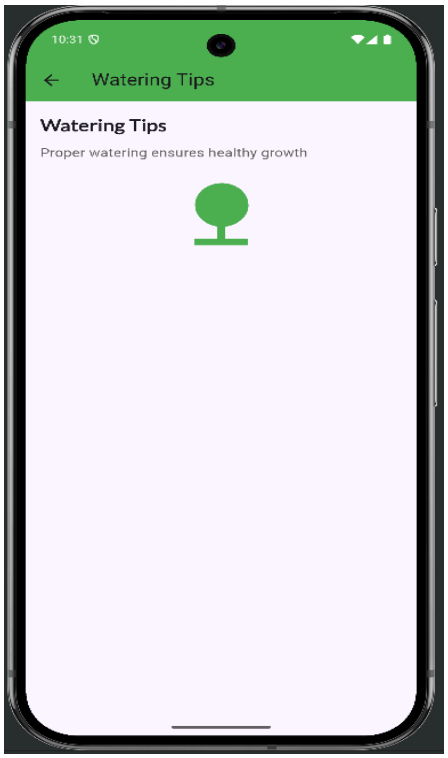
```
  Widget _buildPlantCard(BuildContext context, String title, String description, IconData icon,
String imagePath) {
    return GestureDetector(
      onTap: () {
        Navigator.pushNamed(context, '/details', arguments: {'title': title, 'description': description});
      },
      child: Card(
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
        elevation: 5,
        color: Colors.green.shade100,
        child: Column(
          children: [
            ClipRRect(
              borderRadius: BorderRadius.vertical(top: Radius.circular(15)),
              child: Image.asset(imagePath, height: 80, width: double.infinity, fit: BoxFit.cover),
            ),
            Padding(
              padding: EdgeInsets.all(10),
              child: Column(
                children: [
                  Icon(icon, size: 30, color: Colors.green.shade800),
                  SizedBox(height: 5),
                  Text(title, textAlign: TextAlign.center,
                      style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold, fontFamily:
'CustomFont')),
                  SizedBox(height: 5),
                  Text(description, textAlign: TextAlign.center,
                      style: TextStyle(fontSize: 12, color: Colors.grey.shade700)),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

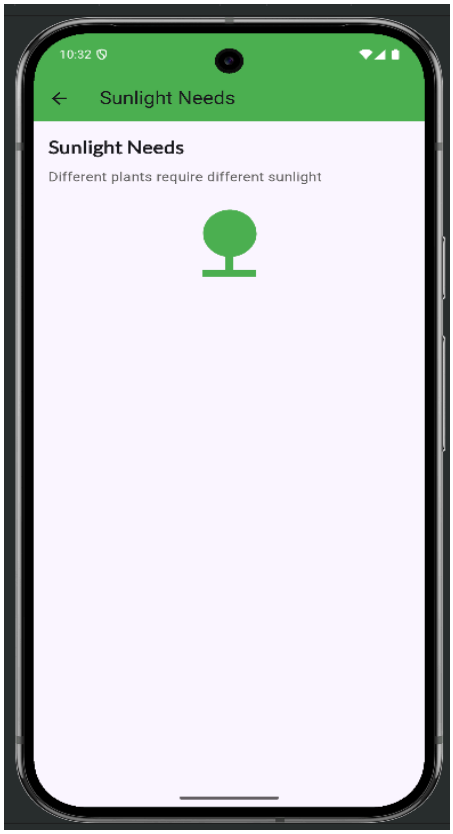
details_screen.dart

```
import 'package:flutter/material.dart';

class DetailsScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final Map<String, String>? args = ModalRoute.of(context)?.settings.arguments as
Map<String, String>?;

    return Scaffold(
```

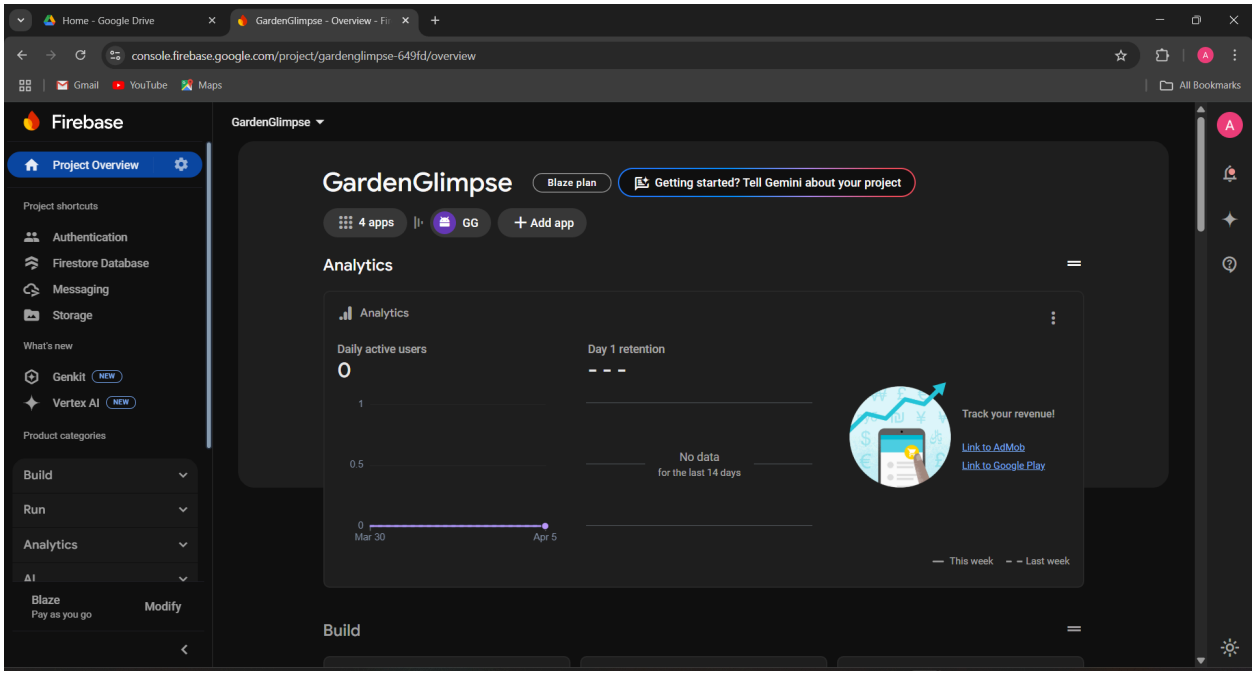
```
      appBar: AppBar(
        title: Text(args?['title'] ?? 'Plant Details'),
        backgroundColor: Colors.green,
      ),
      body: GestureDetector(
        onHorizontalDragEnd: (details) {
          Navigator.pop(context);
        },
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                args?['title'] ?? 'Plant Details',
                style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold, fontFamily: 'CustomFont'),
              ),
              SizedBox(height: 10),
              Text(
                args?['description'] ?? 'No details available',
                style: TextStyle(fontSize: 16, color: Colors.grey.shade700),
              ),
              SizedBox(height: 20),
              Center(
                child: Icon(Icons.nature, size: 100, color: Colors.green),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

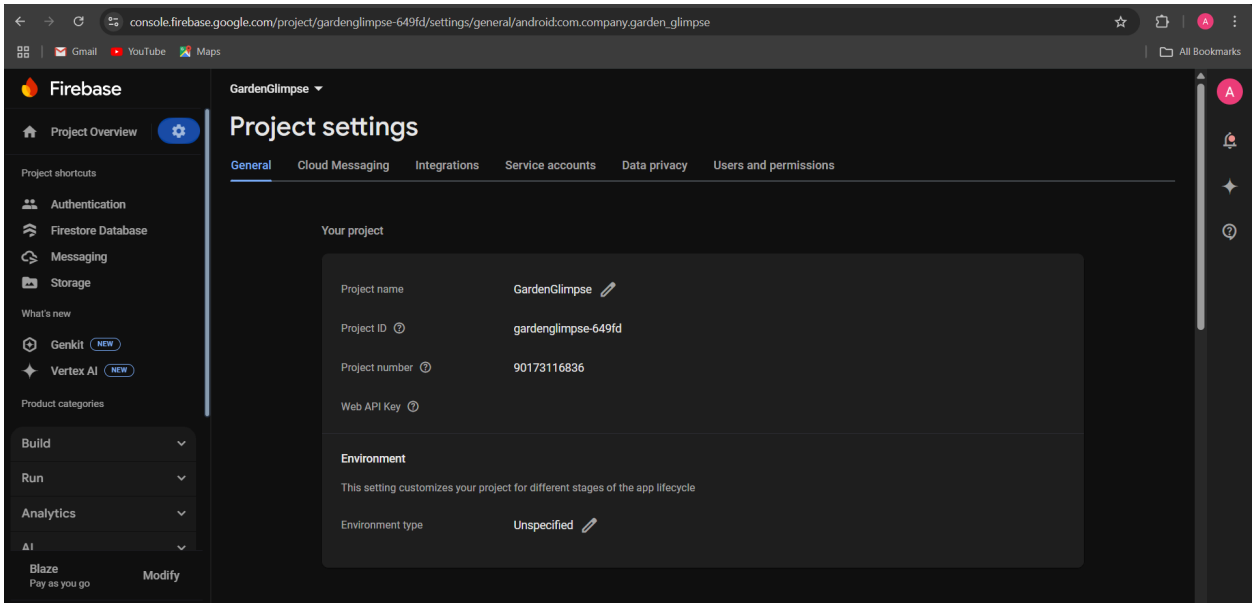

Running Devices
Pixel 9 Pro API 35
10:31
Plant Care Home
Welcome to Plant Care
Nurture your plants with love and care!
Watering Tips
Sunlight Needs

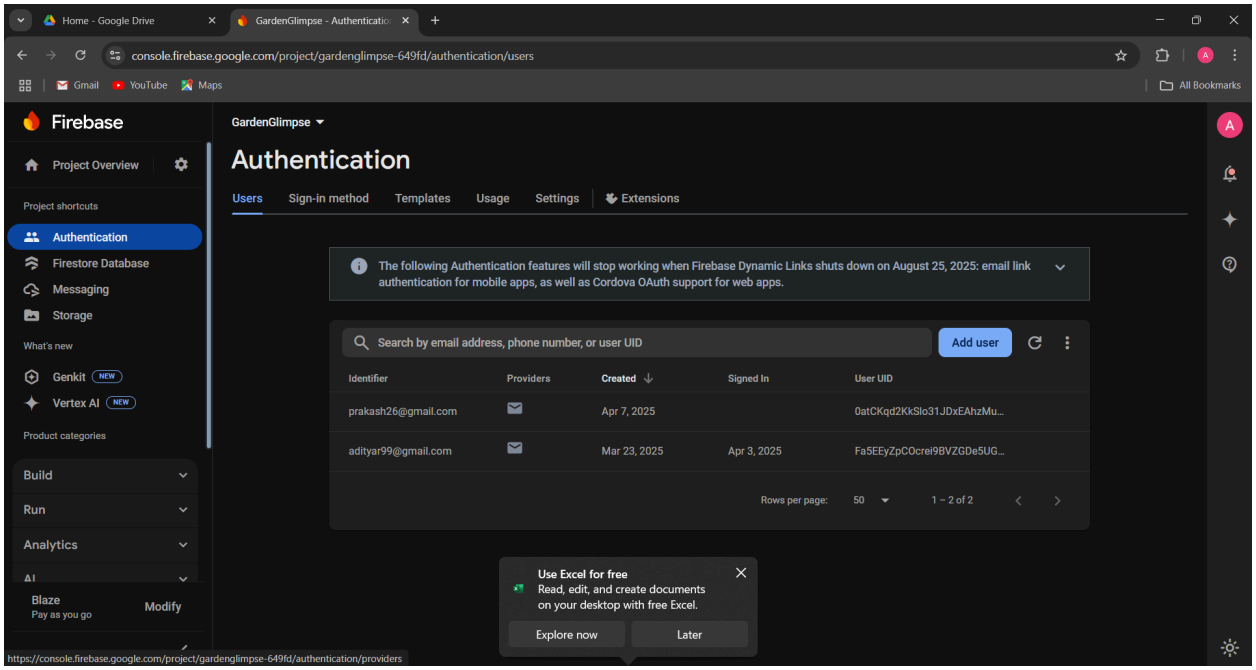

10:31
Watering Tips
Watering Tips
Proper watering ensures healthy growth

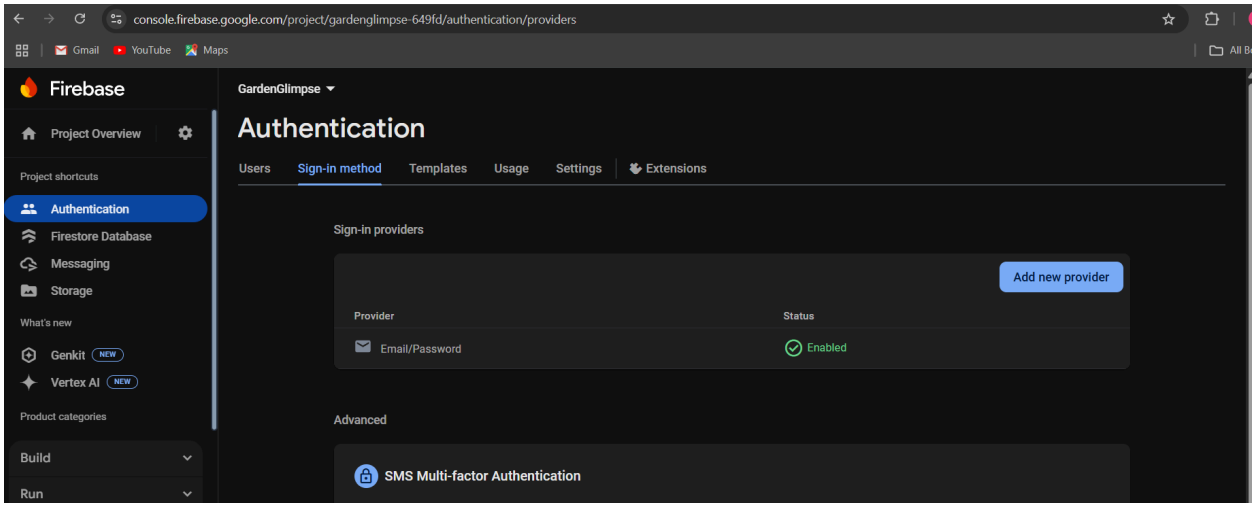

10:32
Sunlight Needs
Sunlight Needs
Different plants require different sunlight

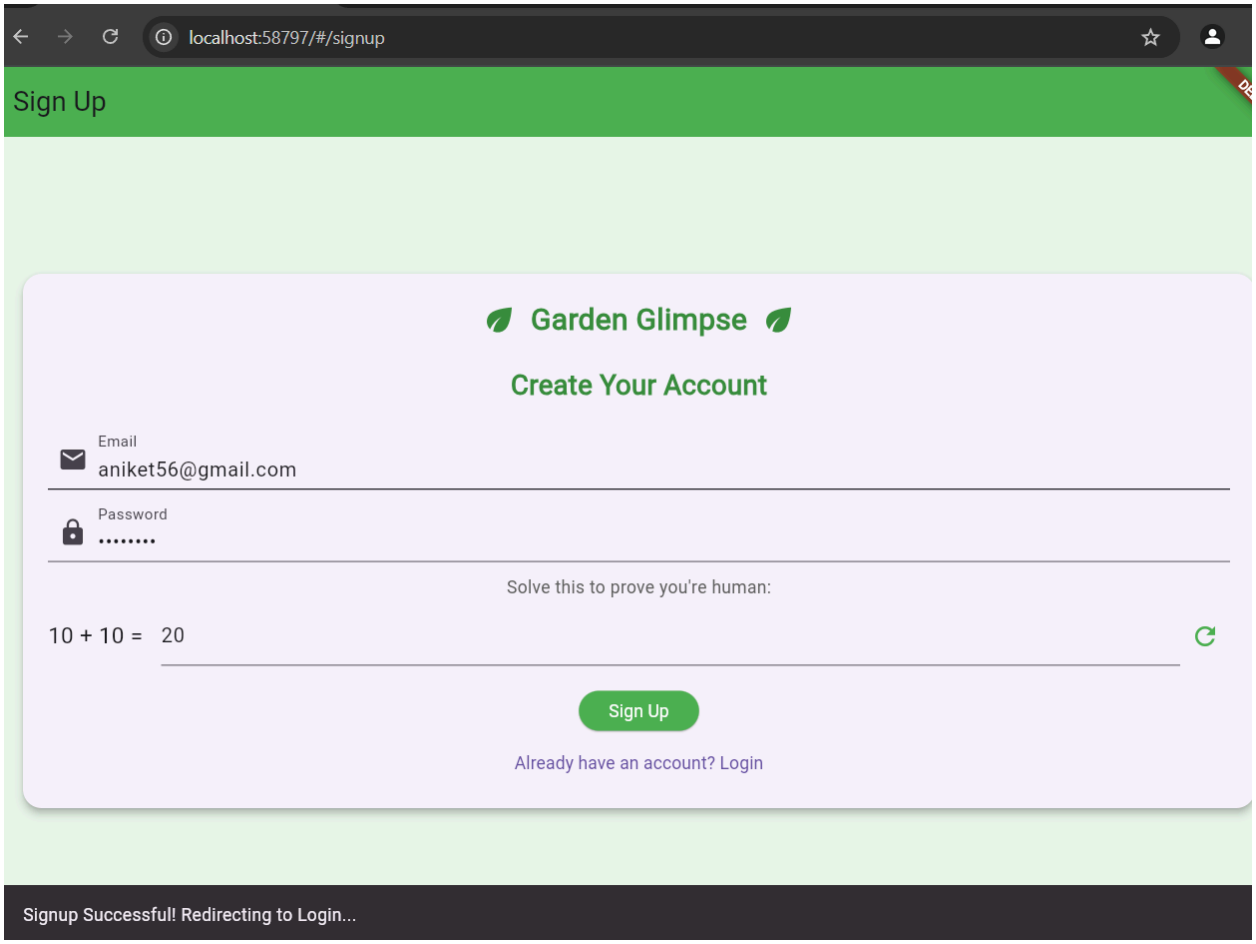

10:32
Plant Details
Plant Details
No details available

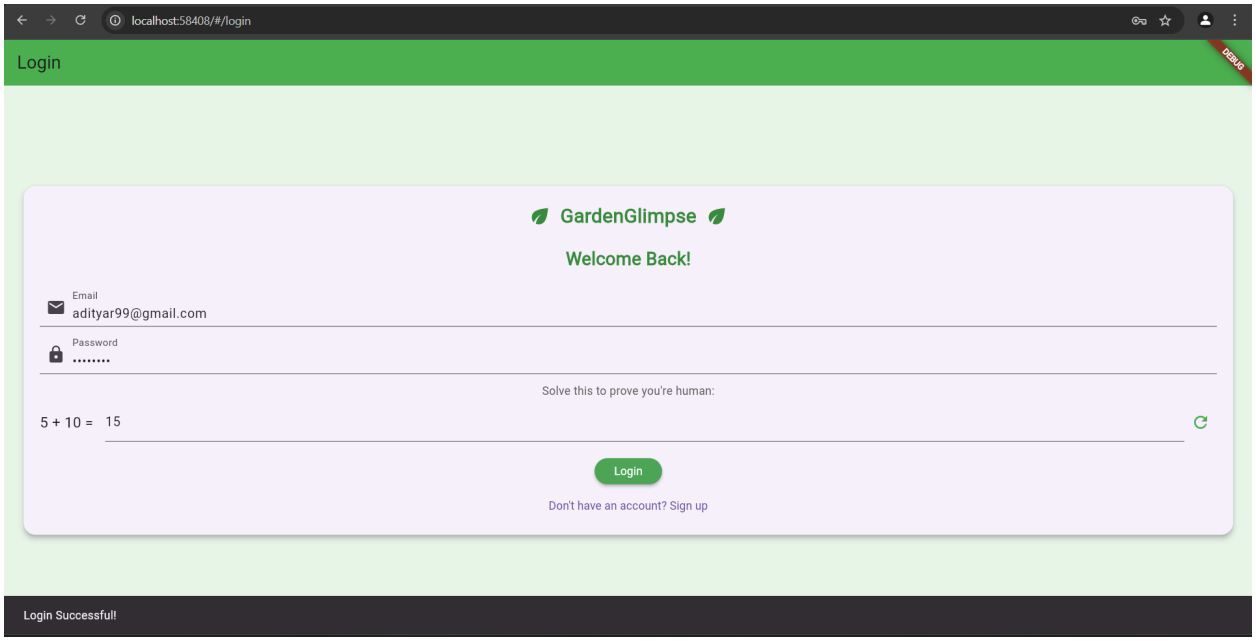

Home - Google Drive
GardenGlimpse - Overview - Fir
console.firebase.google.com/project/gardenglimpse-649fd/overview
Gmail
YouTube
Maps
All Bookmarks
Firebase
GardenGlimpse
Project Overview
Project shortcuts
Authentication
Firestore Database
Messaging
Storage
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
Blaze
Pay as you go
Modify
GardenGlimpse
Blaze plan
Getting started? Tell Gemini about your project
4 apps
GG
Add app
Analytics
Analytics
Daily active users
0
Day 1 retention
- - -
1
0.5
0
Mar 30
Apr 5
No data
for the last 14 days
Track your revenue!
Link to AdMob
Link to Google Play
This week
Last week
Build


console.firebase.google.com/project/gardenglimpse-649fd/settings/general/android:com.company.garden_glimpse
Gmail
YouTube
Maps
All Bookmarks
Firebase
GardenGlimpse
Project Overview
Project shortcuts
Authentication
Firestore Database
Messaging
Storage
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
Blaze
Pay as you go
Modify
Project settings
General
Cloud Messaging
Integrations
Service accounts
Data privacy
Users and permissions
Your project
Project name
GardenGlimpse
Project ID
gardenglimpse-649fd
Project number
90173116836
Web API Key
Environment
This setting customizes your project for different stages of the app lifecycle
Environment type
Unspecified

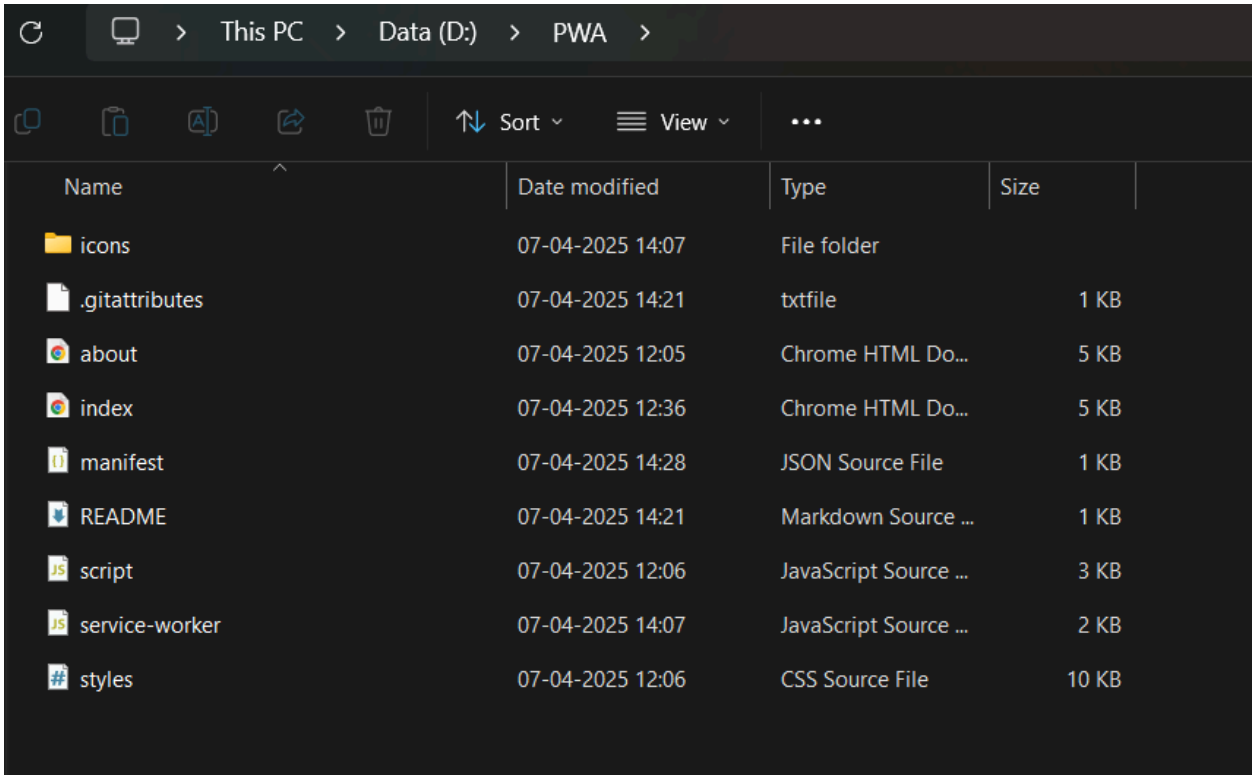

Home - Google Drive
GardenGlimpse - Authentication
console.firebase.google.com/project/gardenglimpse-649fd/authentication/users
Gmail
YouTube
Maps
All Bookmarks
Firebase
GardenGlimpse
Project Overview
Project shortcuts
Authentication
Firestore Database
Messaging
Storage
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
Blaze
Pay as you go
Modify
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
prakash26@gmail.com
Apr 7, 2025
0atCKqd2KkSlo31JDxEAhzMu...
adityar99@gmail.com
Mar 23, 2025
Apr 3, 2025
Fa5EEyZpCOcrei9BVZGDe5UG...
Rows per page:
50
1 – 2 of 2
Use Excel for free
Read, edit, and create documents on your desktop with free Excel.
Explore now
Later
https://console.firebase.google.com/project/gardenglimpse-649fd/authentication/providers


console.firebase.google.com/project/gardenglimpse-649fd/authentication/providers
Gmail
YouTube
Maps
Firebase
GardenGlimpse
Project Overview
Project shortcuts
Authentication
Firestore Database
Messaging
Storage
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
Sign-in providers
Add new provider
Provider
Status
Email/Password
Enabled
Advanced
SMS Multi-factor Authentication

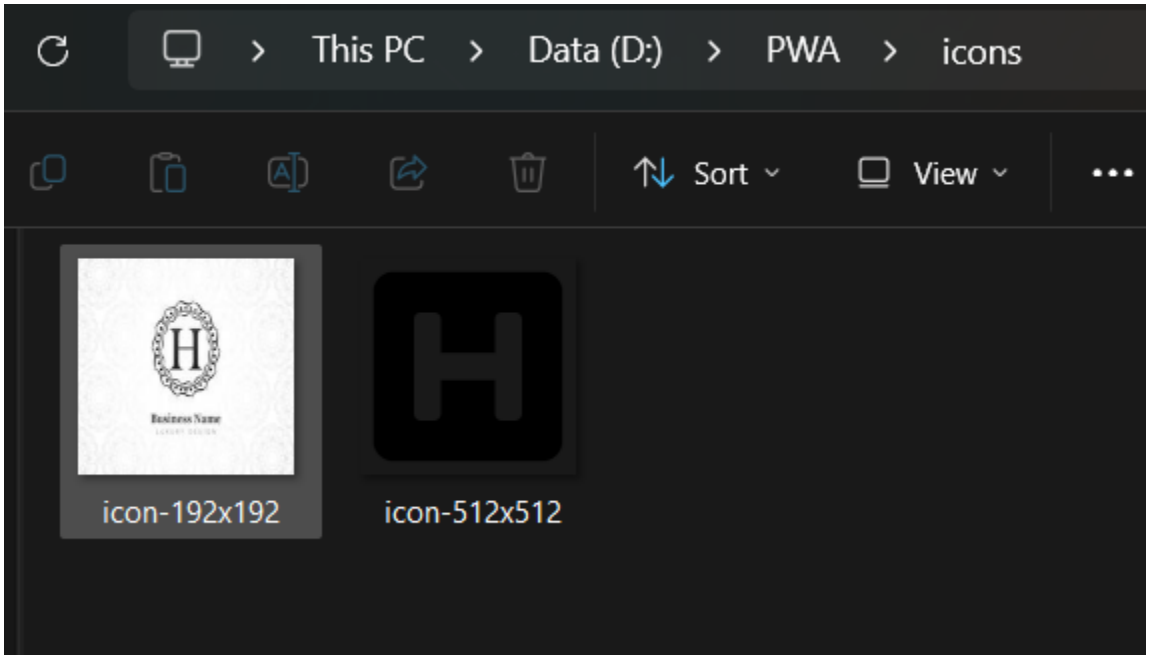

localhost:58797/#/signup
Sign Up
Garden Glimpse
Create Your Account
Email
aniket56@gmail.com
Password
Solve this to prove you're human:
10 + 10 = 20
Sign Up
Already have an account? Login
Signup Successful! Redirecting to Login...


localhost:58408/#/login
Login
GardenGlimpse
Welcome Back!
Email
adityar99@gmail.com
Password
Solve this to prove you're human:
5 + 10 = 15
Login
Don't have an account? Sign up
Login Successful!

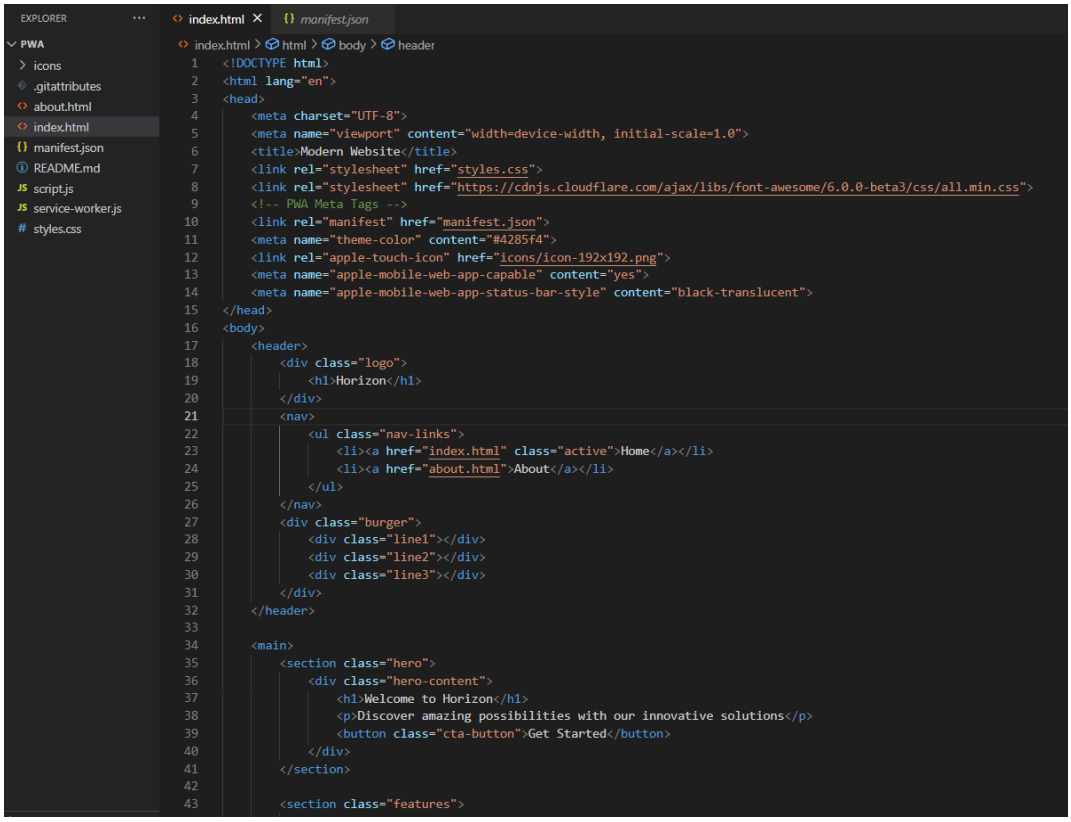
Progressive web app folder



Icons for the app



Index.html and manifest.json

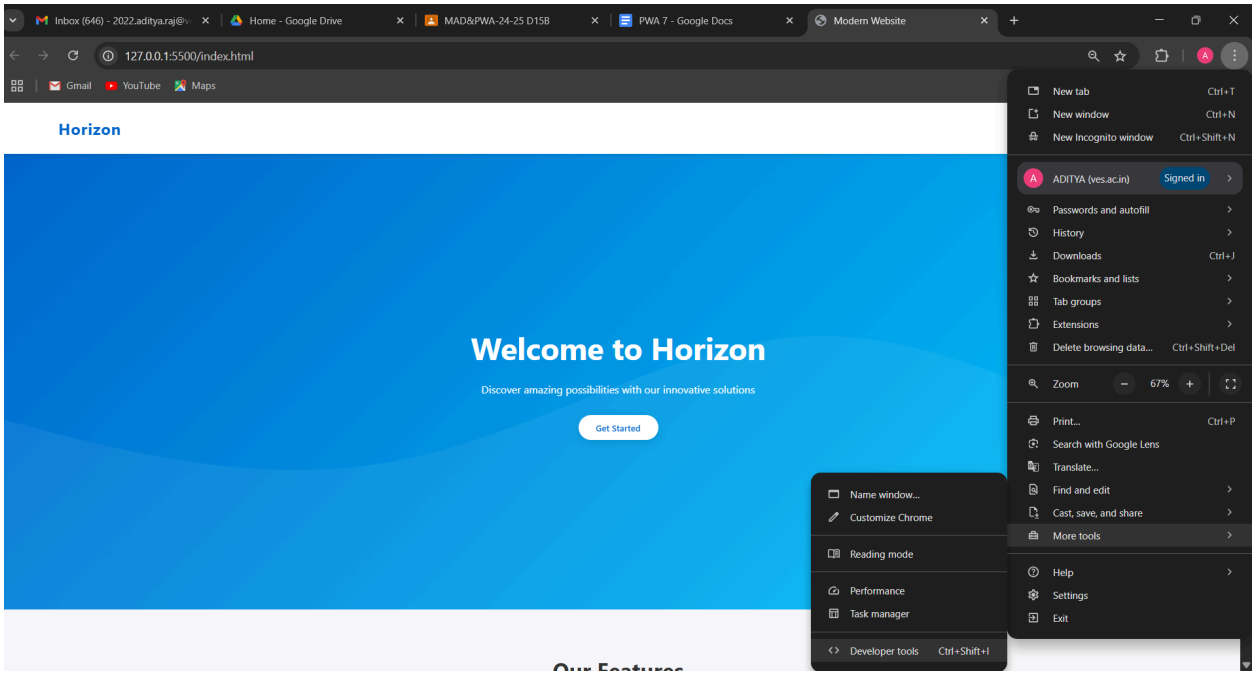


```
EXPLORER
PWA
  icons
  .gitattributes
  about.html
  index.html
  manifest.json
  README.md
  script.js
  service-worker.js
  styles.css
```

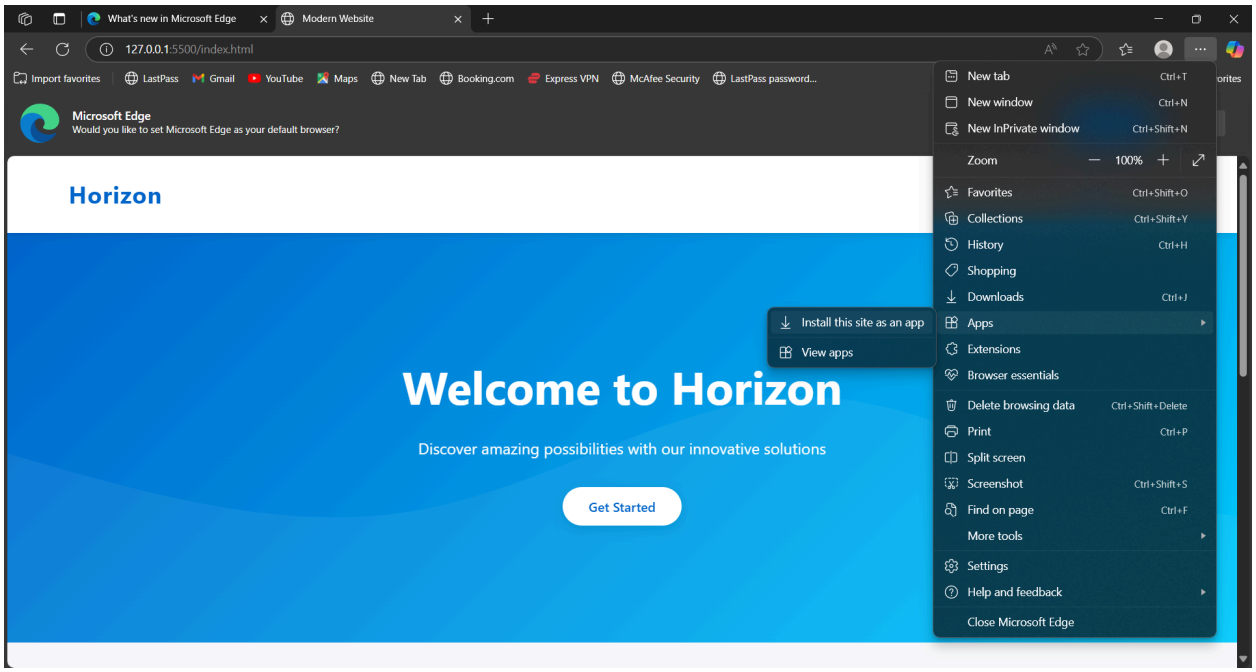
index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modern Website</title>
    <link rel="stylesheet" href="styles.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
    <!-- PWA Meta Tags -->
    <link rel="manifest" href="manifest.json">
    <meta name="theme-color" content="#4285f4">
    <link rel="apple-touch-icon" href="icons/icon-192x192.png">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">
</head>
<body>
    <header>
        <div class="logo">
            <h1>Horizon</h1>
        </div>
        <nav>
            <ul class="nav-links">
                <li><a href="index.html" class="active">Home</a></li>
                <li><a href="about.html">About</a></li>
            </ul>
        </nav>
        <div class="burger">
            <div class="line1"></div>
            <div class="line2"></div>
            <div class="line3"></div>
        </div>
    </header>

    <main>
        <section class="hero">
            <div class="hero-content">
                <h1>Welcome to Horizon</h1>
                <p>Discover amazing possibilities with our innovative solutions</p>
                <button class="cta-button">Get Started</button>
            </div>
        </section>

        <section class="features">
```

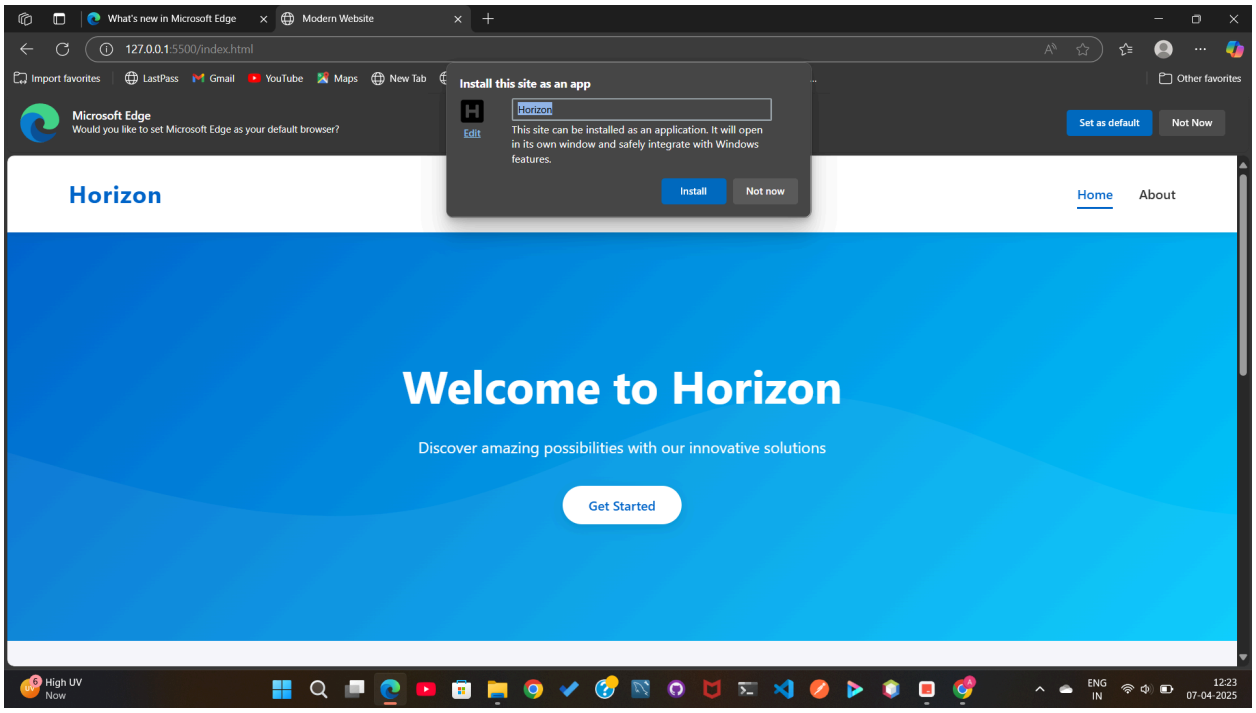
Developer tools option



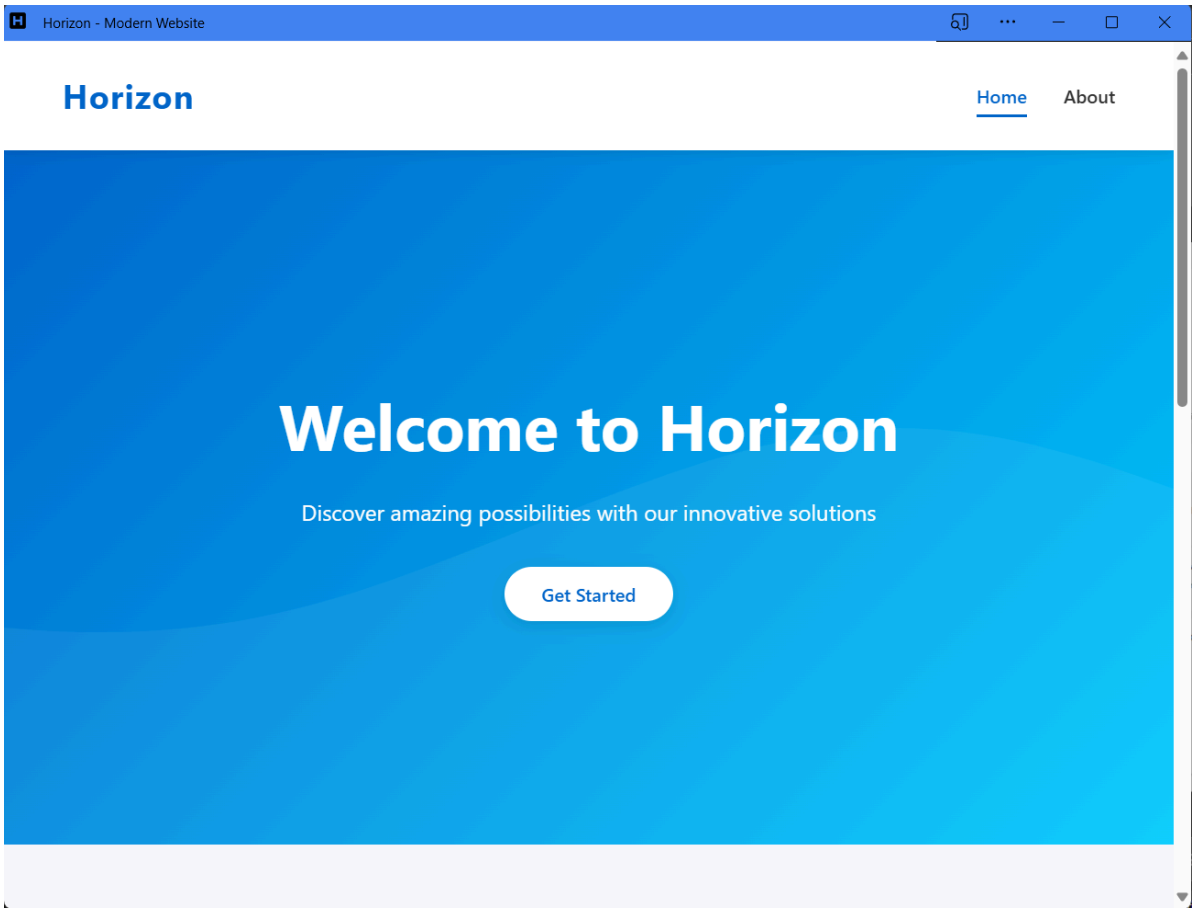
Install this site as an app



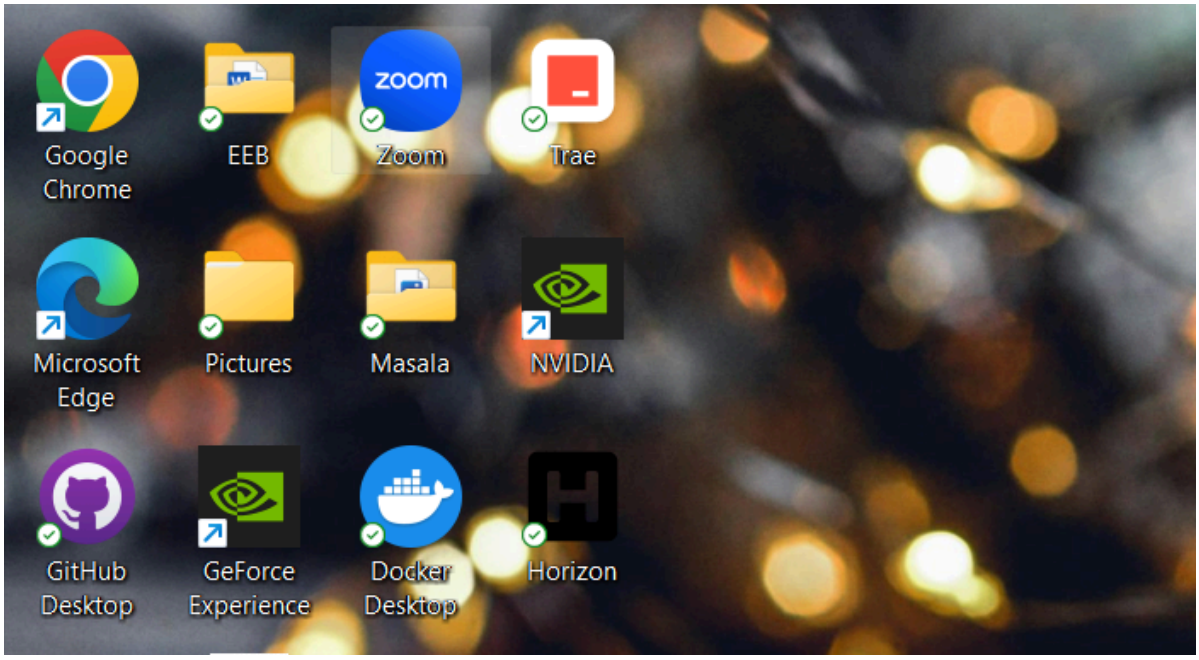
Name and shortcut



The installed progressive web app



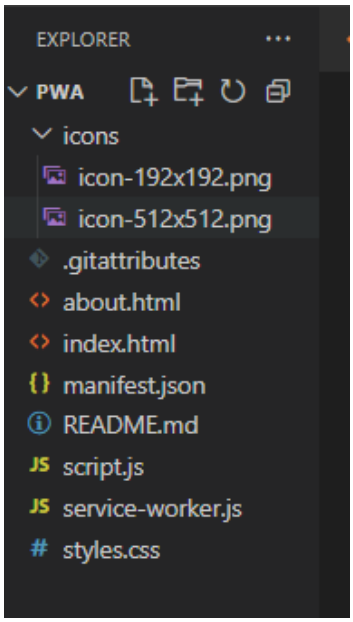
Desktop shortcut for the app



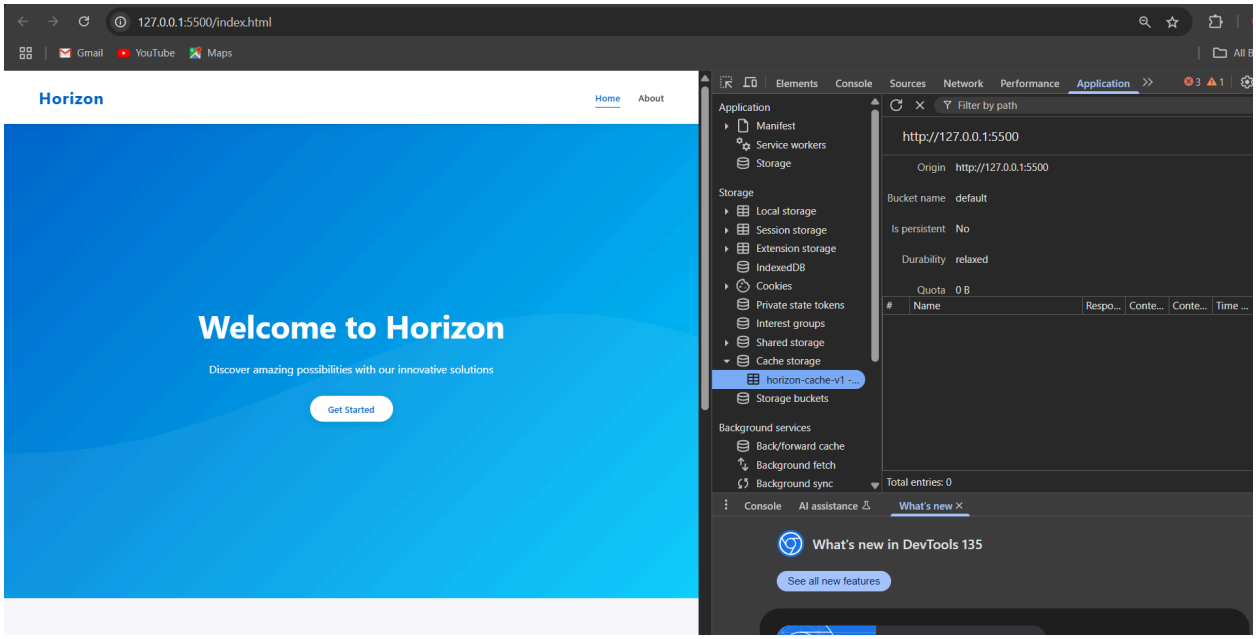
Service-worker.js

```
const CACHE_NAME = 'horizon-cache-v1';
const urlsToCache = [
  '/',
  '/index.html',
  '/about.html',
  '/styles.css',
  '/script.js',
  '/icons/icon-192x192.png',
  '/icons/icon-512x512.png'
];

// Install event - cache assets
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});

// Activate event - clean up old caches
self.addEventListener('activate', event => {
  const cacheWhitelist = [CACHE_NAME];
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.map(cacheName => {
          if (cacheWhitelist.indexOf(cacheName) === -1) {
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});

// Fetch event - serve from cache or network
self.addEventListener('fetch', event => {
  event.respondWith(
```

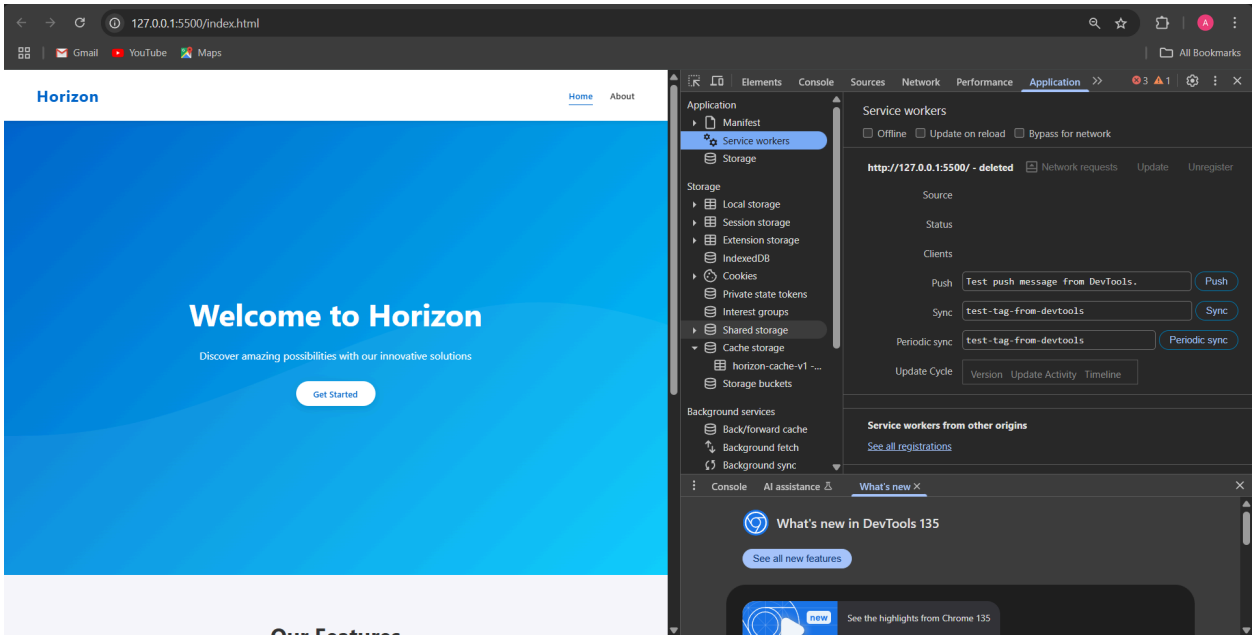
```
    caches.match(event.request)
      .then(response => {
        // Cache hit - return response
        if (response) {
          return response;
        }
        return fetch(event.request).then(
          response => {
            // Check if we received a valid response
            if (!response || response.status !== 200 || response.type !== 'basic') {
              return response;
            }

            // Clone the response
            const responseToCache = response.clone();

            caches.open(CACHE_NAME)
              .then(cache => {
                cache.put(event.request, responseToCache);
              });

            return response;
          }
        );
      })
  );
});
```



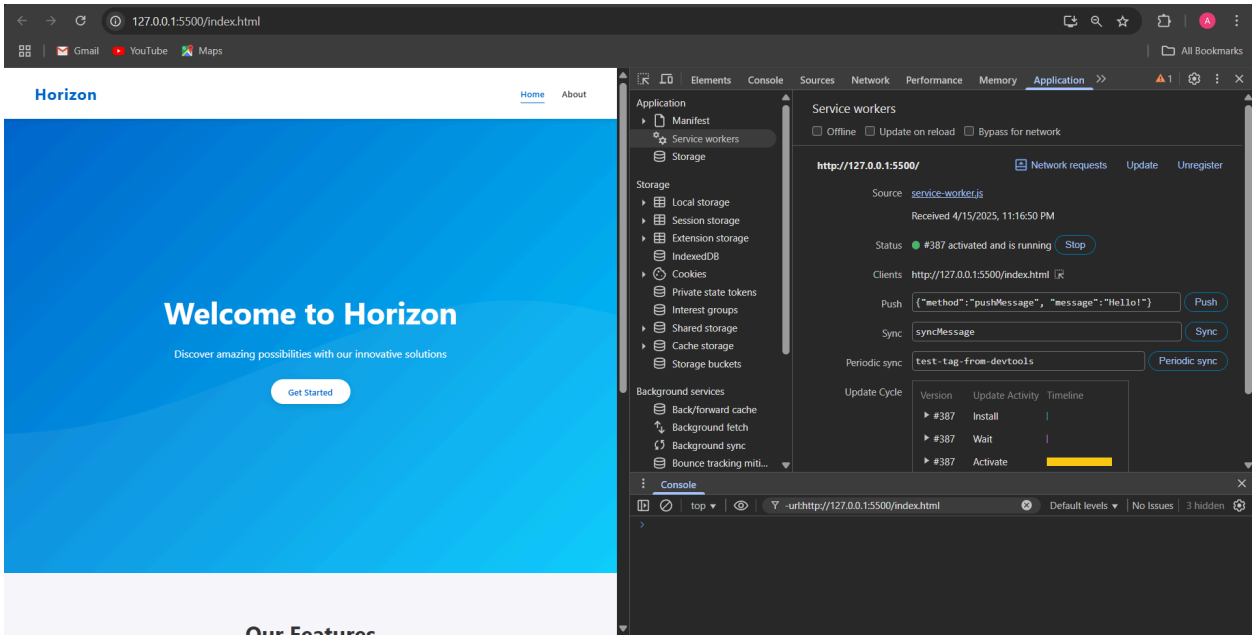
Cache storage

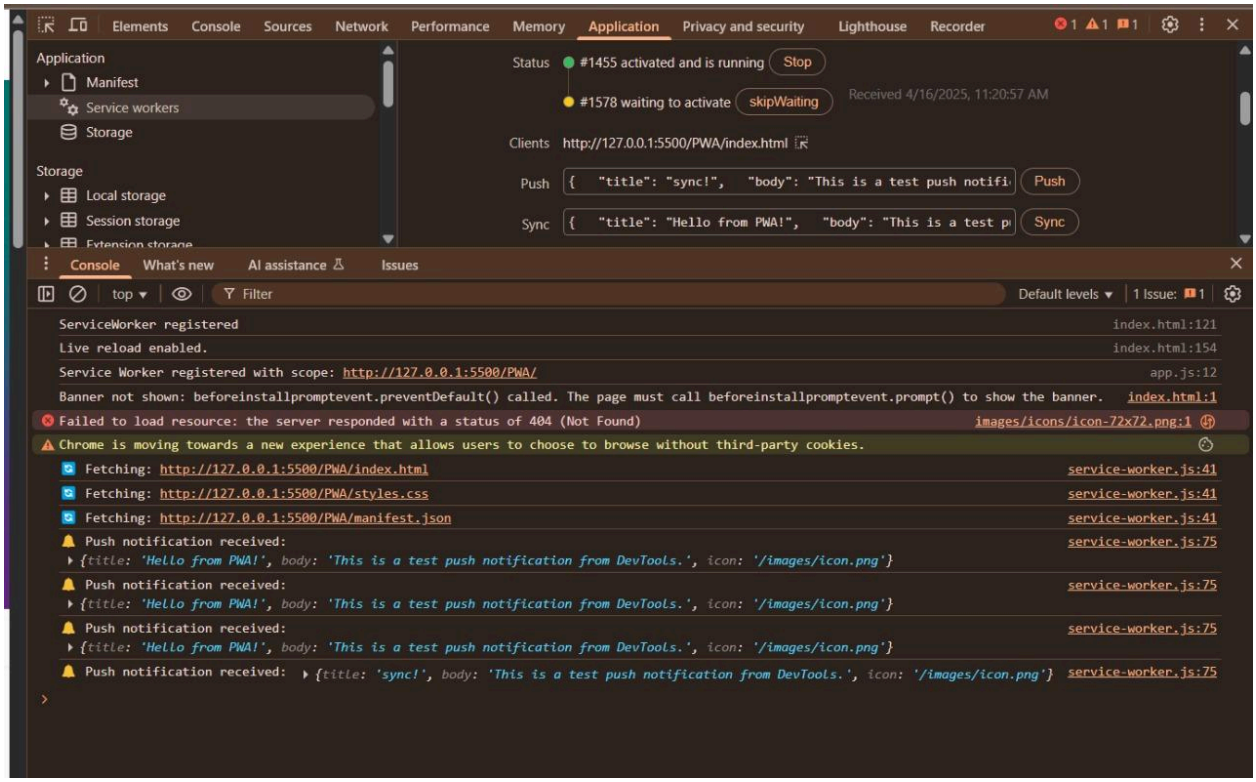


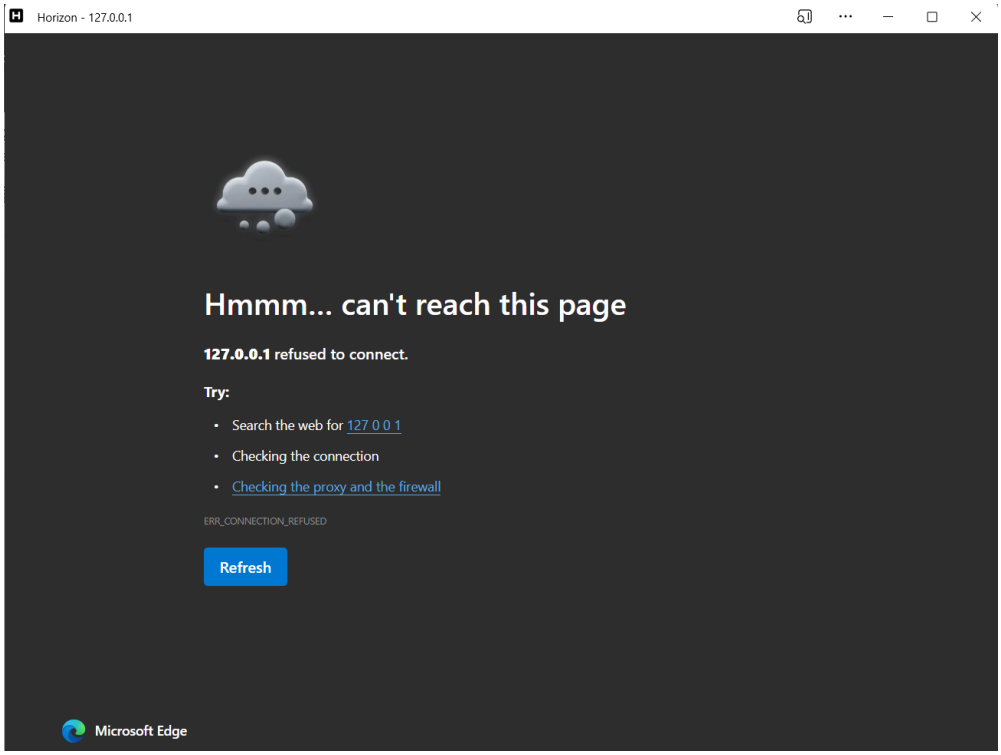
Service workers

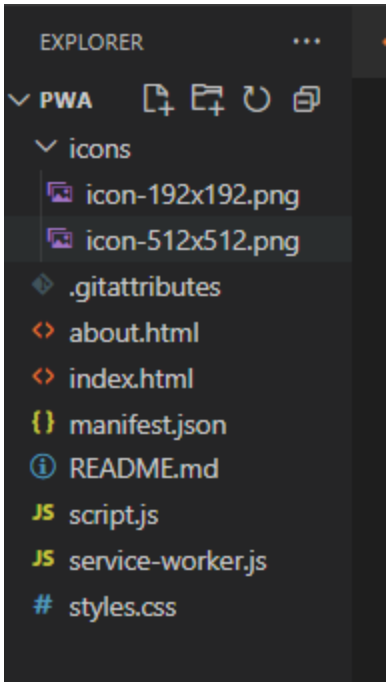


Aim : To implement service worker events like fetch, sync and push

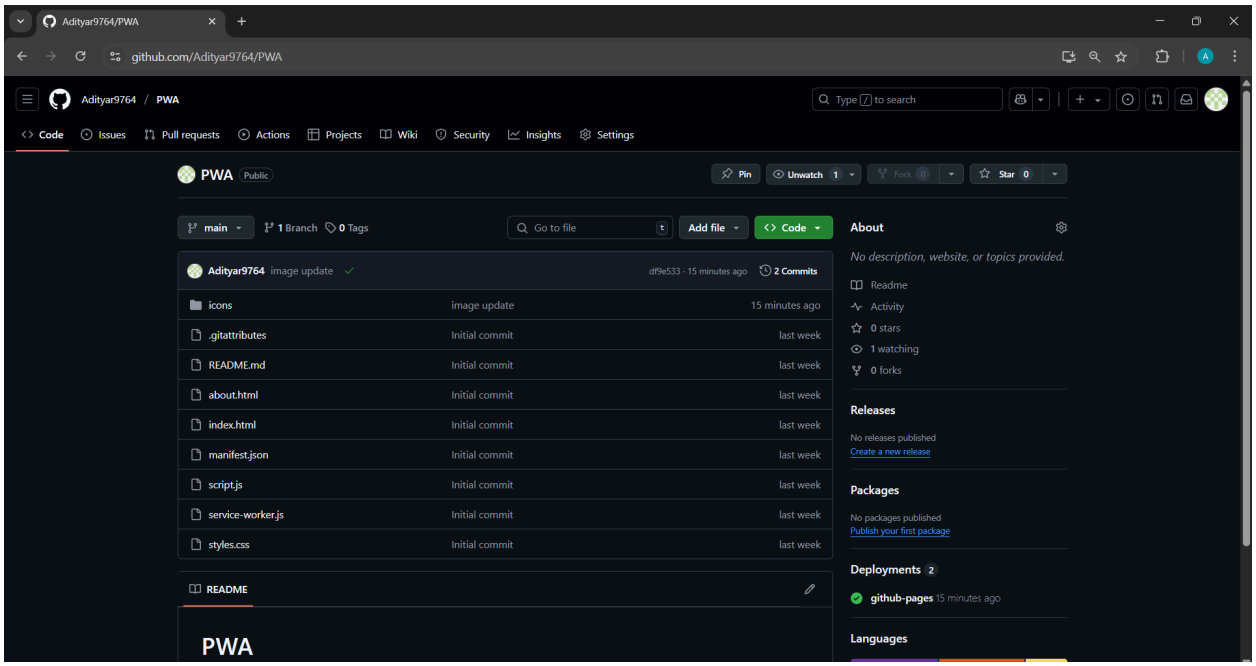



Elements
Console
Sources
Network
Performance
Memory
Application
Privacy and security
Lighthouse
Recorder
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
Status
#1455 activated and is running
Stop
#1578 waiting to activate
skipWaiting
Received 4/16/2025, 11:20:57 AM
Clients
http://127.0.0.1:5500/PWA/index.html
Push
{ "title": "sync!", "body": "This is a test push notifi
Push
Sync
{ "title": "Hello from PWA!", "body": "This is a test p
Sync
Console
What's new
AI assistance
Issues
top
Filter
Default levels
1 Issue:
ServiceWorker registered
index.html:121
Live reload enabled.
index.html:154
Service Worker registered with scope: http://127.0.0.1:5500/PWA/
app.js:12
Banner not shown: beforeinstallpromptevent.preventDefault() called. The page must call beforeinstallpromptevent.prompt() to show the banner.
index.html:1
Failed to load resource: the server responded with a status of 404 (Not Found)
images/icons/icon-72x72.png:1
Chrome is moving towards a new experience that allows users to choose to browse without third-party cookies.
Fetching: http://127.0.0.1:5500/PWA/index.html
service-worker.js:41
Fetching: http://127.0.0.1:5500/PWA/styles.css
service-worker.js:41
Fetching: http://127.0.0.1:5500/PWA/manifest.json
service-worker.js:41
Push notification received:
service-worker.js:75
{title: 'Hello from PWA!', body: 'This is a test push notification from DevTools.', icon: '/images/icon.png'}
Push notification received:
service-worker.js:75
{title: 'Hello from PWA!', body: 'This is a test push notification from DevTools.', icon: '/images/icon.png'}
Push notification received:
service-worker.js:75
{title: 'Hello from PWA!', body: 'This is a test push notification from DevTools.', icon: '/images/icon.png'}
Push notification received:
{title: 'sync!', body: 'This is a test push notification from DevTools.', icon: '/images/icon.png'}
service-worker.js:75

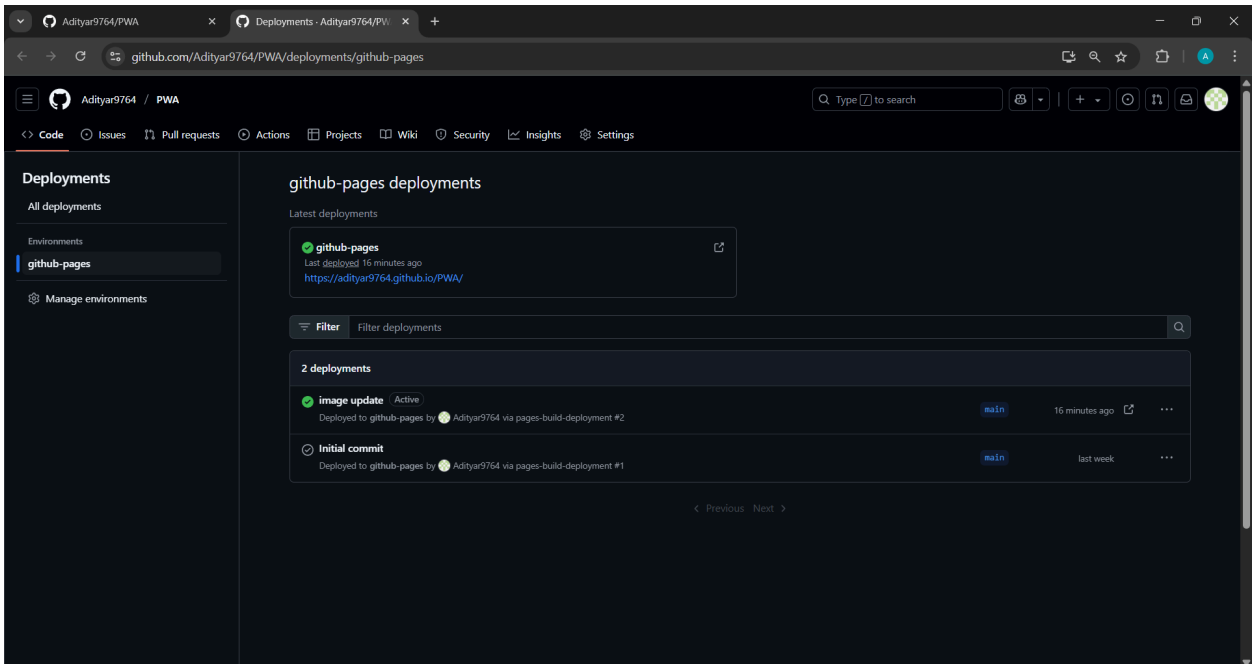

Horizon - 127.0.0.1
Hmmm... can't reach this page
127.0.0.1 refused to connect.
Try:
Search the web for 127 0 0 1
Checking the connection
Checking the proxy and the firewall
ERR_CONNECTION_REFUSED
Refresh
Microsoft Edge



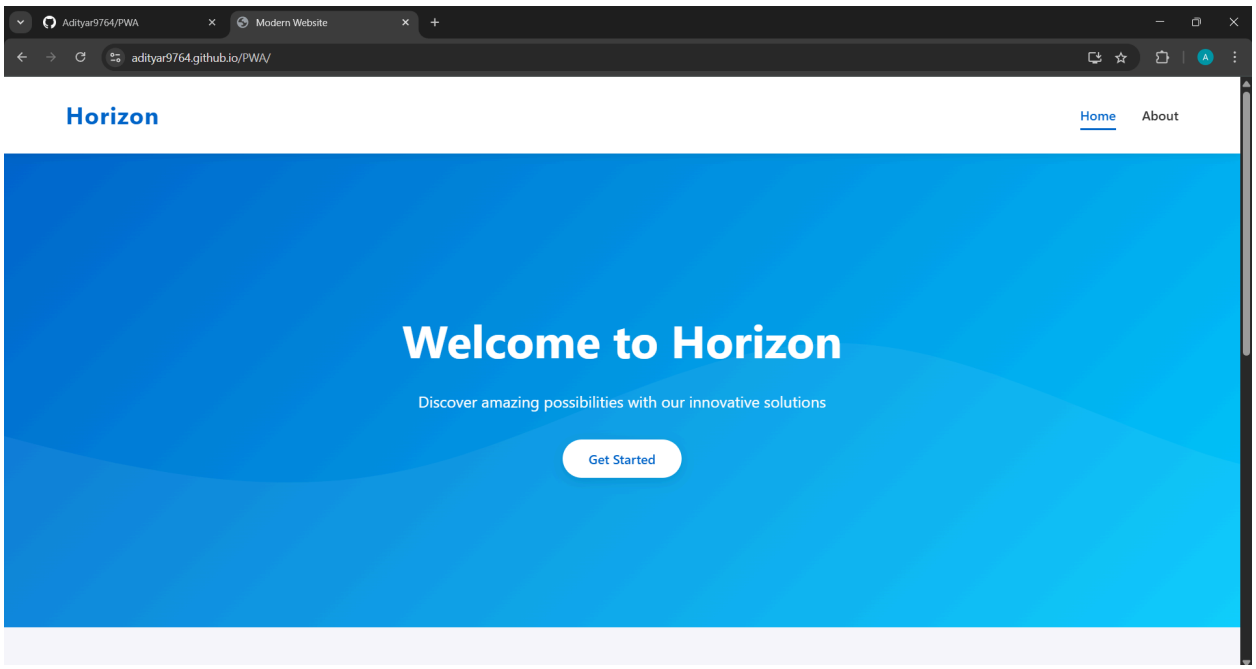
Commit and push files to Github



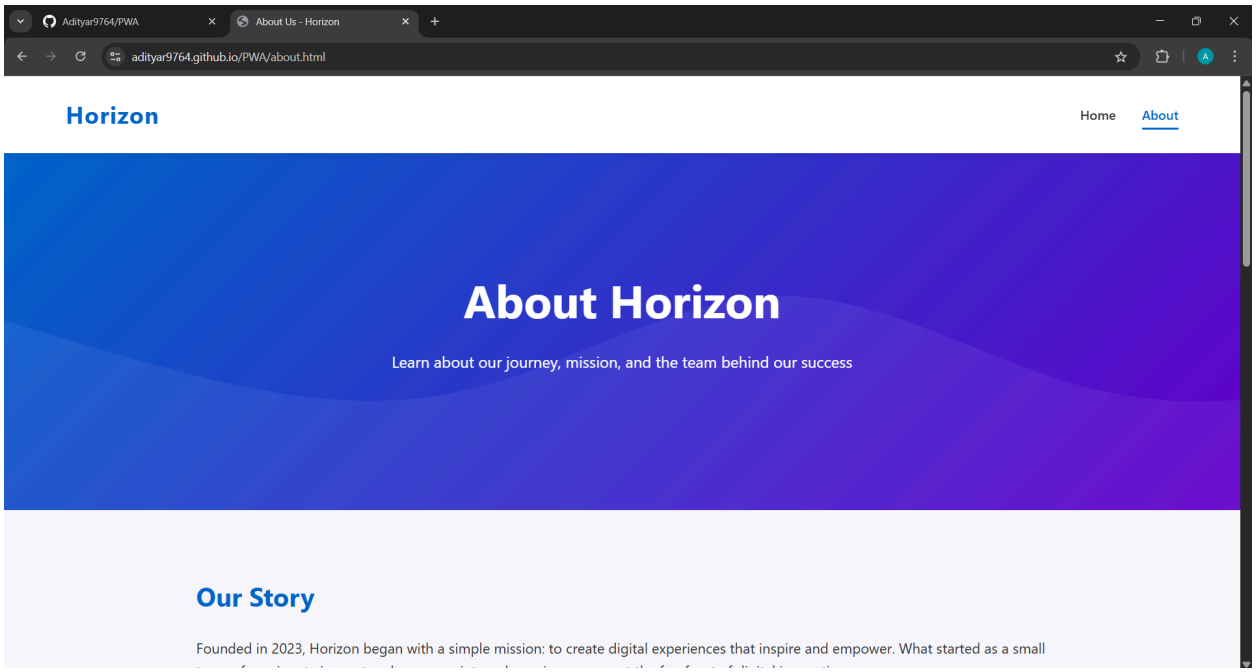
Go to deployments and configure



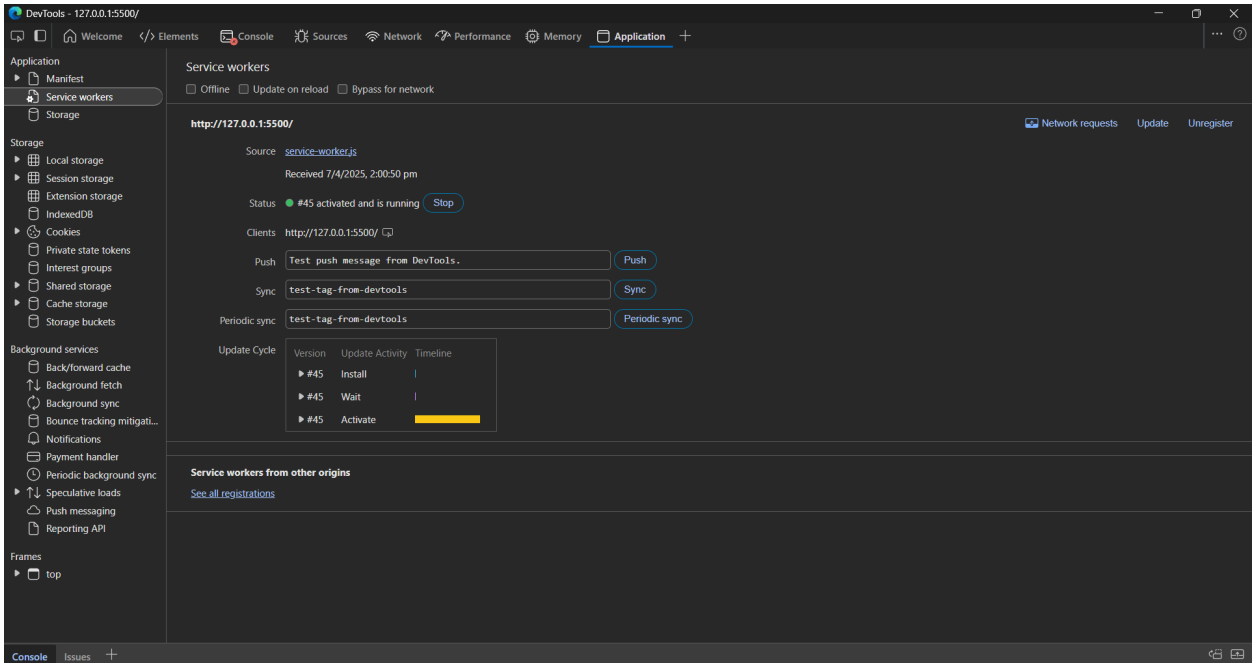
Visiting our deployed site



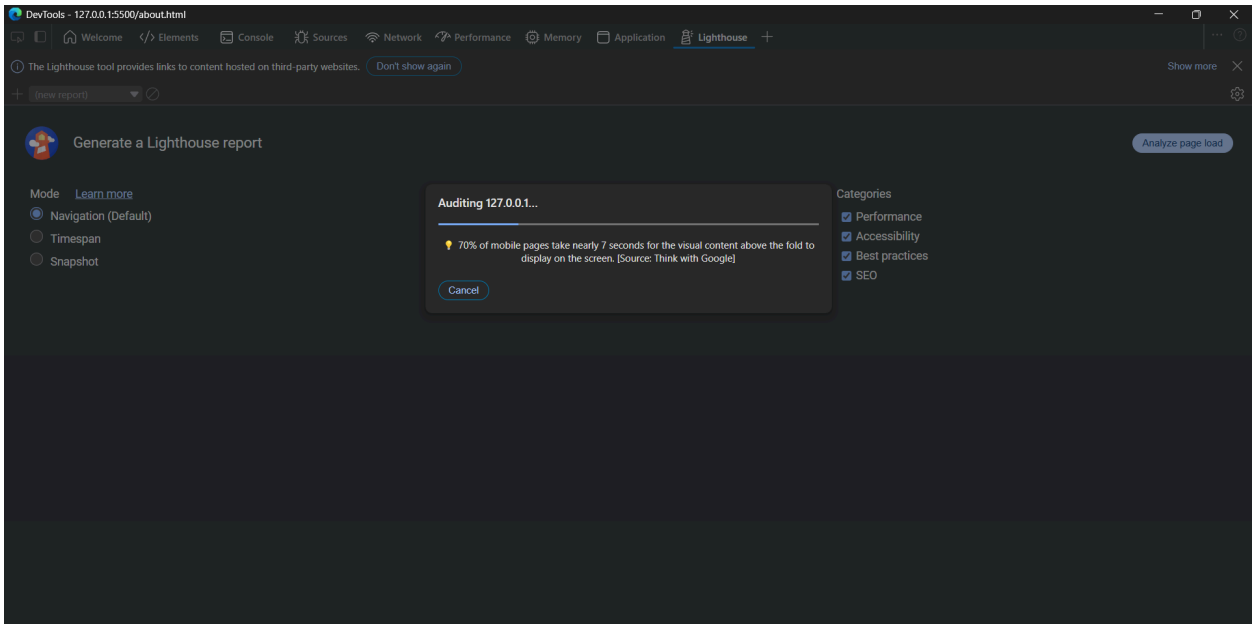
Checking if all routes are working



DevTools → application → service workers



Lighthouse report generation



Final lighthouse report