FRONT END

import 'package:flutter/material.dart';

void main() {

runApp(FarmersApp());

}

class FarmersApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Farmers CrowdSource App',

theme: ThemeData(

primarySwatch: Colors.green,

),

home: HomePage(),

);

}

}

class HomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Farmers CrowdSource App'),

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

ElevatedButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (context) => InputPage()),

);

},

child: Text('Report Disease or Pest'),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (context) => ViewReportsPage()),

);

},

child: Text('View Reports'),

),

],

),

),

);

}

}

class InputPage extends StatefulWidget {

@override

\_InputPageState createState() => \_InputPageState();

}

class \_InputPageState extends State<InputPage> {

final TextEditingController \_nameController = TextEditingController();

final TextEditingController \_descriptionController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Report Disease/Pest'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

children: <Widget>[

TextField(

controller: \_nameController,

decoration: InputDecoration(labelText: 'Disease or Pest Name'),

),

TextField(

controller: \_descriptionController,

decoration: InputDecoration(labelText: 'Description'),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {

// Placeholder for submitting data to the backend

String name = \_nameController.text;

String description = \_descriptionController.text;

print('Submitted: $name - $description');

// Clear the text fields after submission

\_nameController.clear();

\_descriptionController.clear();

},

child: Text('Submit'),

),

],

),

),

);

}

}

class ViewReportsPage extends StatelessWidget {

// Example data from crowd-sourced database

final List<Map<String, String>> reports = [

{

'name': 'Tomato Leaf Spot',

'description': 'Brown spots on leaves, causing wilting.'

},

{

'name': 'Corn Rust',

'description': 'Red-brown pustules on leaves. Spreads fast.'

},

];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Reported Diseases/Pests'),

),

body: ListView.builder(

itemCount: reports.length,

itemBuilder: (context, index) {

return ListTile(

title: Text(reports[index]['name']!),

subtitle: Text(reports[index]['description']!),

);

},

),

);

}

}

BACK END

mkdir farmers-app-backend

cd farmers-app-backend

npm init -y

npm install express mongoose body-parser cors

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

const PORT = 3000;

// Middleware

app.use(cors());

app.use(bodyParser.json());

// Connect to MongoDB

mongoose.connect('mongodb://localhost:27017/farmers\_app', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'connection error:'));

db.once('open', () => {

console.log('Connected to MongoDB');

});

// Schema for Reports

const reportSchema = new mongoose.Schema({

name: String,

description: String,

date: { type: Date, default: Date.now },

});

const Report = mongoose.model('Report', reportSchema);

// Routes

// Create a new report

app.post('/api/reports', (req, res) => {

const newReport = new Report({

name: req.body.name,

description: req.body.description,

});

newReport.save((err, report) => {

if (err) return res.status(500).send(err);

return res.status(200).send(report);

});

});

// Get all reports

app.get('/api/reports', (req, res) => {

Report.find({}, (err, reports) => {

if (err) return res.status(500).send(err);

return res.status(200).send(reports);

});

});

// Start the server

app.listen(PORT, () => {

console.log(`Server is running on port ${PORT}`);

});

TEST UR API

mongod

node index.js

{

"name": "Tomato Leaf Spot",

"description": "Brown spots on leaves causing wilting."

}

CONNECTING FRONTEND

import 'dart:convert';

import 'package:http/http.dart' as http;

// Function to submit data to the backend

Future<void> submitReport(String name, String description) async {

final url = Uri.parse('http://localhost:3000/api/reports');

final response = await http.post(

url,

headers: {'Content-Type': 'application/json'},

body: jsonEncode({'name': name, 'description': description}),

);

if (response.statusCode == 200) {

print('Report submitted successfully');

} else {

print('Failed to submit report');

}

}