

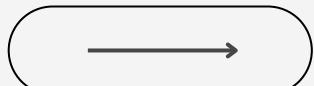


SMART FARMING

ENHANCING SUSTAINABLE AGRICULTURE

GUIDE NAME

Bhavya Parvathi P
Assistant Professor
ADS Department



PRESENTED BY

Madhav Manohar
Mohammed Anfas P V
Mohammed Afthab
Mohammed Hamdan

TABLE OF CONTENTS

01	INTRODUCTION	05	ARCHITECTURE	09	TIME PLAN
02	PROBLEM STATEMENT	06	PROPOSED SYSTEM	10	CONCLUSION
03	OBJECTIVE	07	DATASET	11	REFERENCES
04	RELATED WORKS	08	CURRENT PROGRESS		

INTRODUCTION

- Smart Farming: An innovative software platform that utilizes data-driven insights and AI to assist farmers in adopting sustainable practices.
- Significance: Addresses the pressing challenges in agriculture, such as resource inefficiency and environmental degradation.
- Key feature: Provides true and accurate information to farmers



PROBLEM STATEMENT

Current Challenges:

- Inefficient resource management leading to increased costs.
- Difficulty in pest identification and management.
- Information overload and the prevalence of misinformation about farming practices.
- Impact: These challenges contribute to reduced agricultural productivity and environmental harm.



OBJECTIVE

Primary Goals:

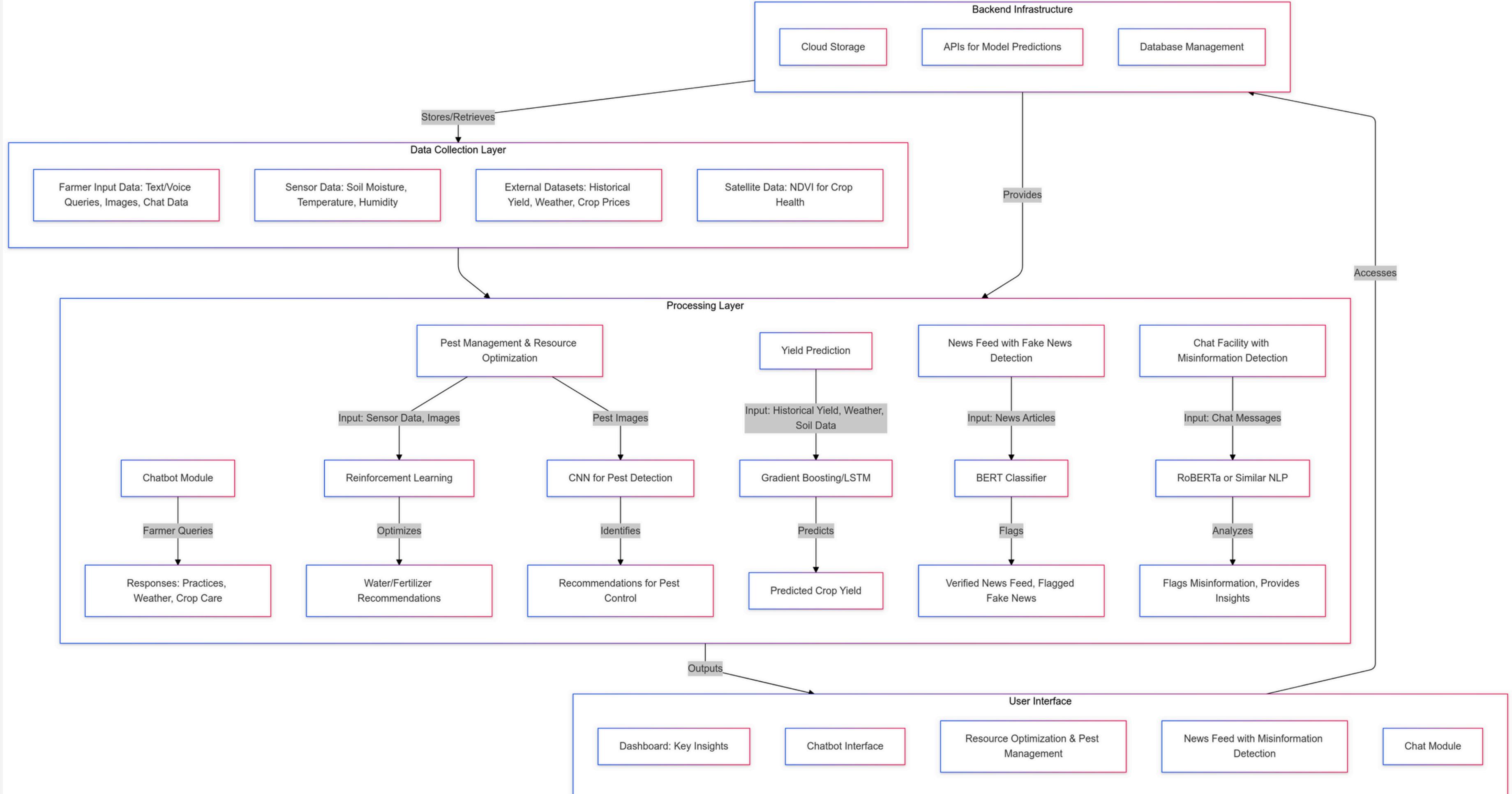
- To develop a user-friendly platform that provides personalized farming plans.
- To leverage AI for optimized resource usage and pest management.
- To promote sustainable agricultural practices and ecological restoration.
- To provide true and accurate information for farmers.



RELATED WORKS

Title	Year	Abstract
<ul style="list-style-type: none">• AI-Driven Decision Support in Agriculture.	2023	<ul style="list-style-type: none">• This paper reviews AI-driven decision support systems that assist farmers in making informed choices about crop management and resource allocation for sustainability.
<ul style="list-style-type: none">• Chatbots for Agricultural Guidance.	2023	<ul style="list-style-type: none">• This study evaluates the usability and effectiveness of AI chatbots in delivering farming advice and facilitating communication among farmers.
<ul style="list-style-type: none">• Smart Pest Management.	2022	<ul style="list-style-type: none">• The paper discusses integrating IoT sensors and AI for real time pest detection and management, improving crop health.
<ul style="list-style-type: none">• Fake News Detection in Agricultural Contexts.	2022	<ul style="list-style-type: none">• The study presents a machine learning framework designed to detect fake news related to agricultural practices and policies.

ARCHITECTURE



PROPOSED SYSTEM

Features of the Smart Farming Platform:

- **Chatbot for Personalized Plans:** Users input data to receive tailored farming strategies.
- **Resource Optimization Tools:** Suggestions for water usage, fertilizer application, etc. Sensor Data: Soil moisture, temperature, humidity sensors in fields.
- **Pest Management Recommendations:** Alerts and advice based on AI analysis.
- **News Feed with Misinformation Detection:** Delivers verified information and alerts users to fake news.
- **User Communication:** Chat feature to connect farmers for knowledge sharing with Misinformation Detection.

PROPOSED SYSTEM

- **Overall Aim:** Enhance agricultural productivity while promoting ecological sustainability
- **Yield Prediction:** Estimates potential crop yield based on historical data and inputs.

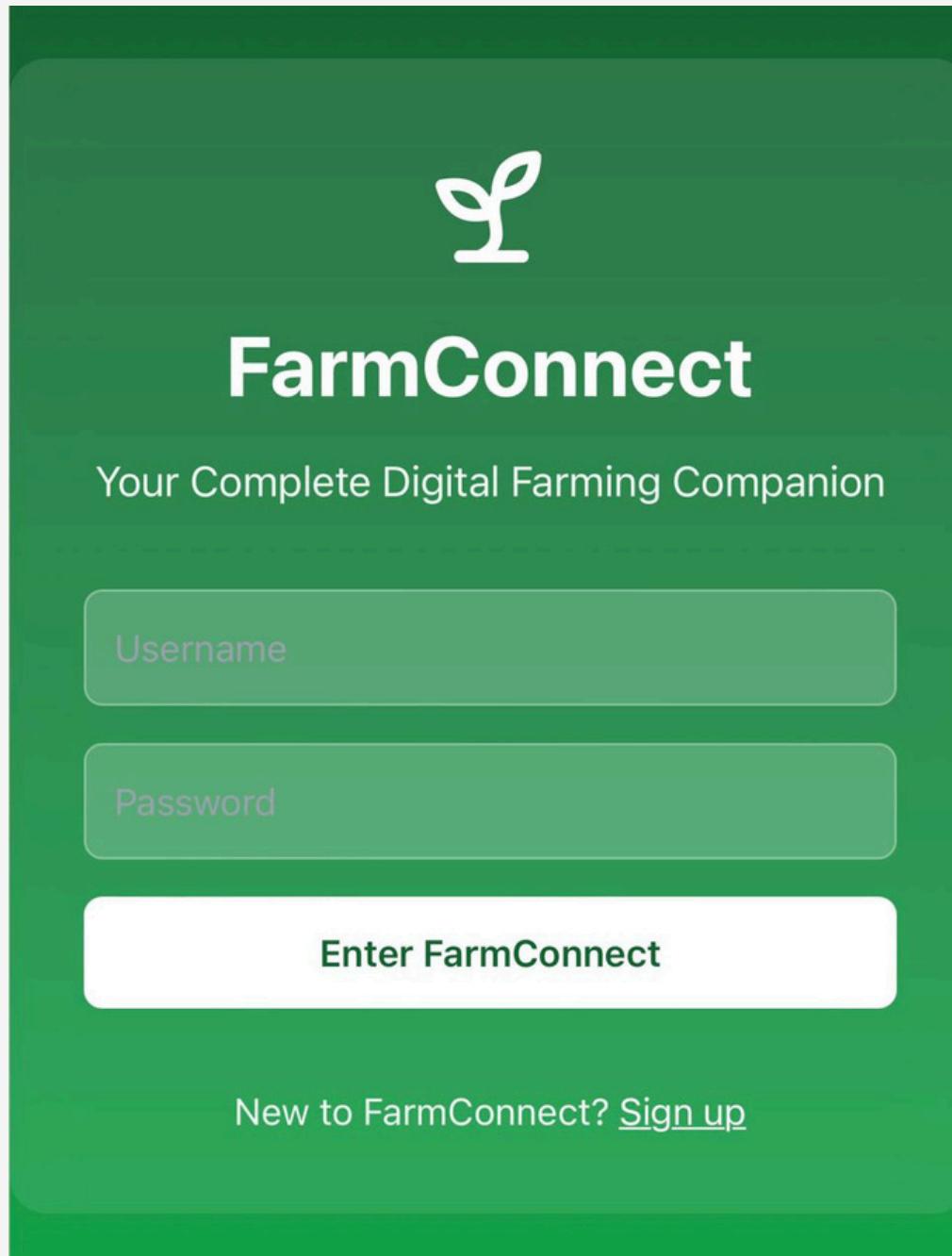


FRONT END DEVELOPMENT

1. Programming Language and Technologies Used.

- React.js: A JavaScript library for building user interfaces.
- JavaScript (JSX): Code is written in JavaScript with React's syntax extension (JSX).
- Tailwind CSS: For styling the app with predefined classes.
- Lucide-React: Icon library used for buttons and visuals.
- Custom Components: Reusable components like Alert, Card, etc., for better UI.

CURRENT PROGRESS



- This is the login page for the FarmConnect application.
- Username and Password Fields: Allows users to securely log in.
- Enter Button: A call-to-action button to access app's features
- Sign-Up Option: A link for new users to create an account.

```
my-app > src > JS FarmerApp.js > ...
1 import React, { useState } from 'react';
2 import WelcomePage from './components/WelcomePage';
3 import ChatbotPage from './components/ChatbotPage';
4 import NewsPage from './components/NewsPage';
5 import PestDetectionPage from './components/PestDetectionPage';
6 import WeatherPage from './components/WeatherPage';
7 import CommunityPage from './components/CommunityPage';
8 import DesktopIcon from './components/DesktopIcon';
9 import { Sprout, Newspaper, Camera, Cloud, Users } from 'lucide-react'; // Import additional icons
10 import './App.css'; // Ensure you import your CSS file
11
12 const FarmerApp = () => {
13   const [isLoggedIn, setIsLoggedIn] = useState(false);
14   const [currentPage, setCurrentPage] = useState('desktop');
15
16   if (!isLoggedIn) {
17     return <WelcomePage onEnter={() => setIsLoggedIn(true)} />;
18   }
19
20   const renderPage = () => {
21     switch (currentPage) {
22       case 'chatbot':
23         return <ChatbotPage onBack={() => setCurrentPage('desktop')} />;
24       case 'news':
25         return <NewsPage onBack={() => setCurrentPage('desktop')} />;
26       case 'pest-detection':
27         return <PestDetectionPage onBack={() => setCurrentPage('desktop')} />;
28       case 'weather':
29         return <WeatherPage onBack={() => setCurrentPage('desktop')} />;
30       case 'community':
31         return <CommunityPage onBack={() => setCurrentPage('desktop')} />;
32       case 'desktop-icon':
33         return <DesktopIcon onBack={() => setCurrentPage('desktop')} />;
34     }
35   }
36
37   return <div>{renderPage()}</div>
38 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

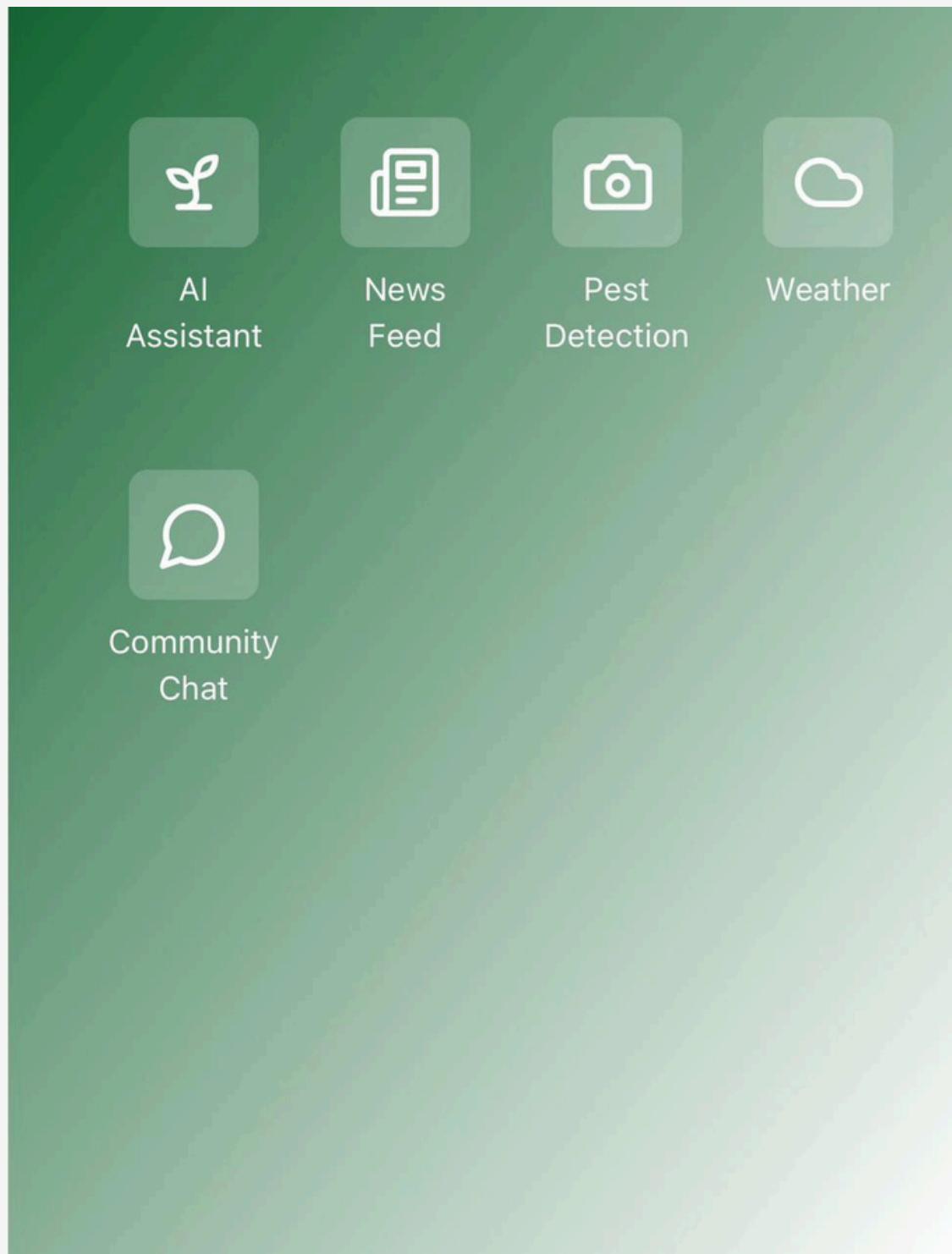
Login page front end code

```
my-app > app.py > login
1 @app.route('/')
2 def home():
3     return "Hello, Flask!"
4
5 @app.route('/login', methods=['POST'])
6 def login():
7     data = request.json
8     print("Received request data:", data) # Debug log
9
10    if not data:
11        return jsonify({"message": "Invalid request format"}), 400
12
13    username = data.get('username')
14    password = data.get('password')
15
16    print(f"Received username: {username}")
17    print(f"Received password: {password}")
18
19    if username == 'admin' and password == 'password':
20        return jsonify({"message": "Login successful"}), 200
21    else:
22        return jsonify({"message": "Invalid credentials"}), 401
23
24 if __name__ == '__main__':
25     app.run()
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Login page back end code

CURRENT PROGRESS



- This is the main dashboard of the Farm Connect application.
- users can navigate to various features.

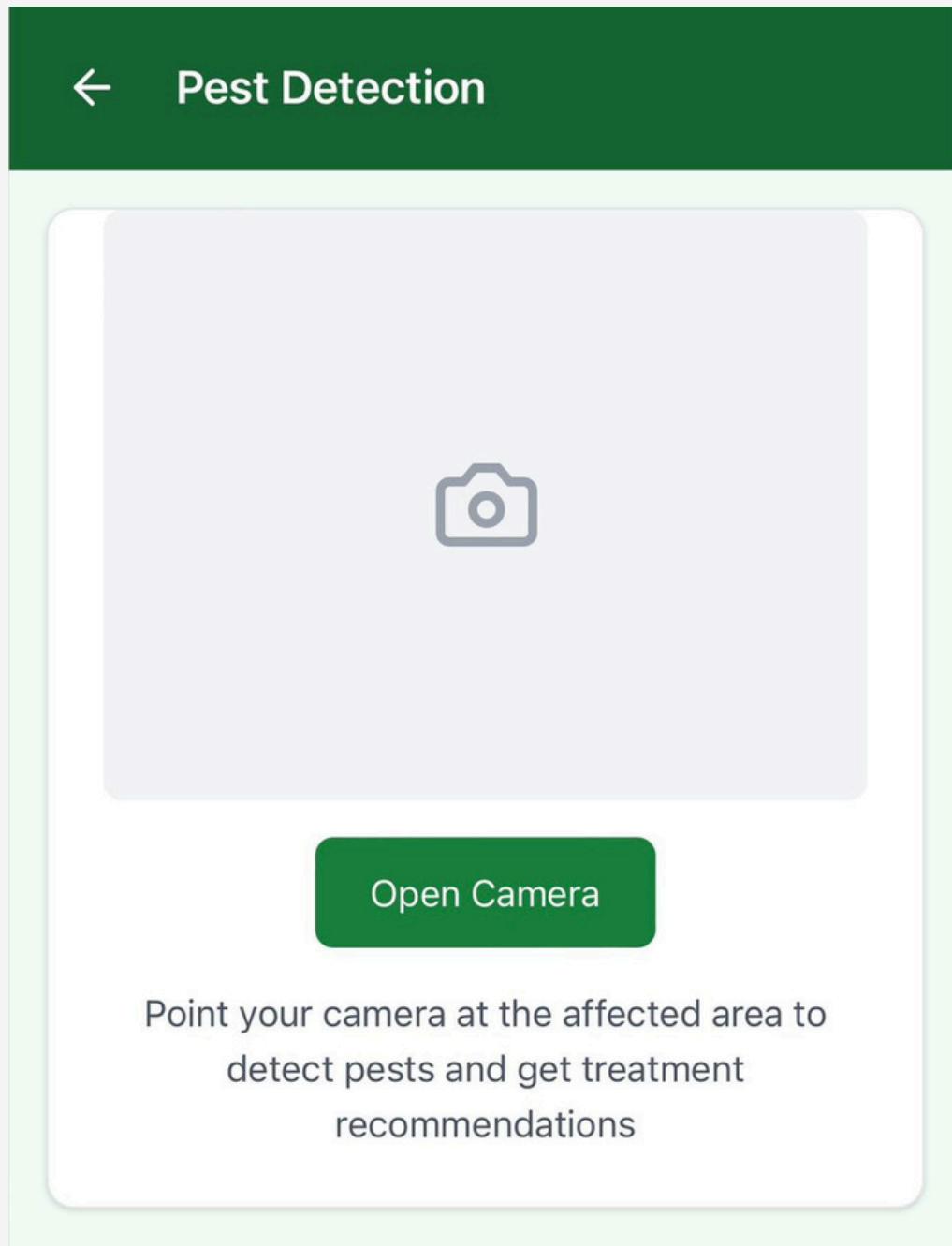
```
my-app > src > JS FarmerApp.js > ...
1 import React, { useState } from 'react';
2 import WelcomePage from './components/WelcomePage';
3 import ChatbotPage from './components/ChatbotPage';
4 import NewsPage from './components/NewsPage';
5 import PestDetectionPage from './components/PestDetectionPage';
6 import WeatherPage from './components/WeatherPage';
7 import CommunityPage from './components/CommunityPage';
8 import DesktopIcon from './components/DesktopIcon';
9 import { Sprout, Newspaper, Camera, Cloud, Users } from 'lucide-react'; // Import additional icons
10 import './App.css'; // Ensure you import your CSS file
11
12 const FarmerApp = () => {
13   const [isLoggedIn, setIsLoggedIn] = useState(false);
14   const [currentPage, setCurrentPage] = useState('desktop');
15
16   if (!isLoggedIn) {
17     return <WelcomePage onEnter={() => setIsLoggedIn(true)} />;
18   }
19
20   const renderPage = () => {
21     switch (currentPage) {
22       case 'chatbot':
23         return <ChatbotPage onBack={() => setCurrentPage('desktop')} />;
24       case 'news':
25     }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Front end code for dashboard page

CURRENT PROGRESS



- This page helps users identify pests affecting their crops.
- provides treatment recommendations.
- Users can tap the "Open Camera" button to activate their device's camera.
- Point the camera at the affected crop area to detect pests.
- The system will analyze the image and provide recommendations for pest control.

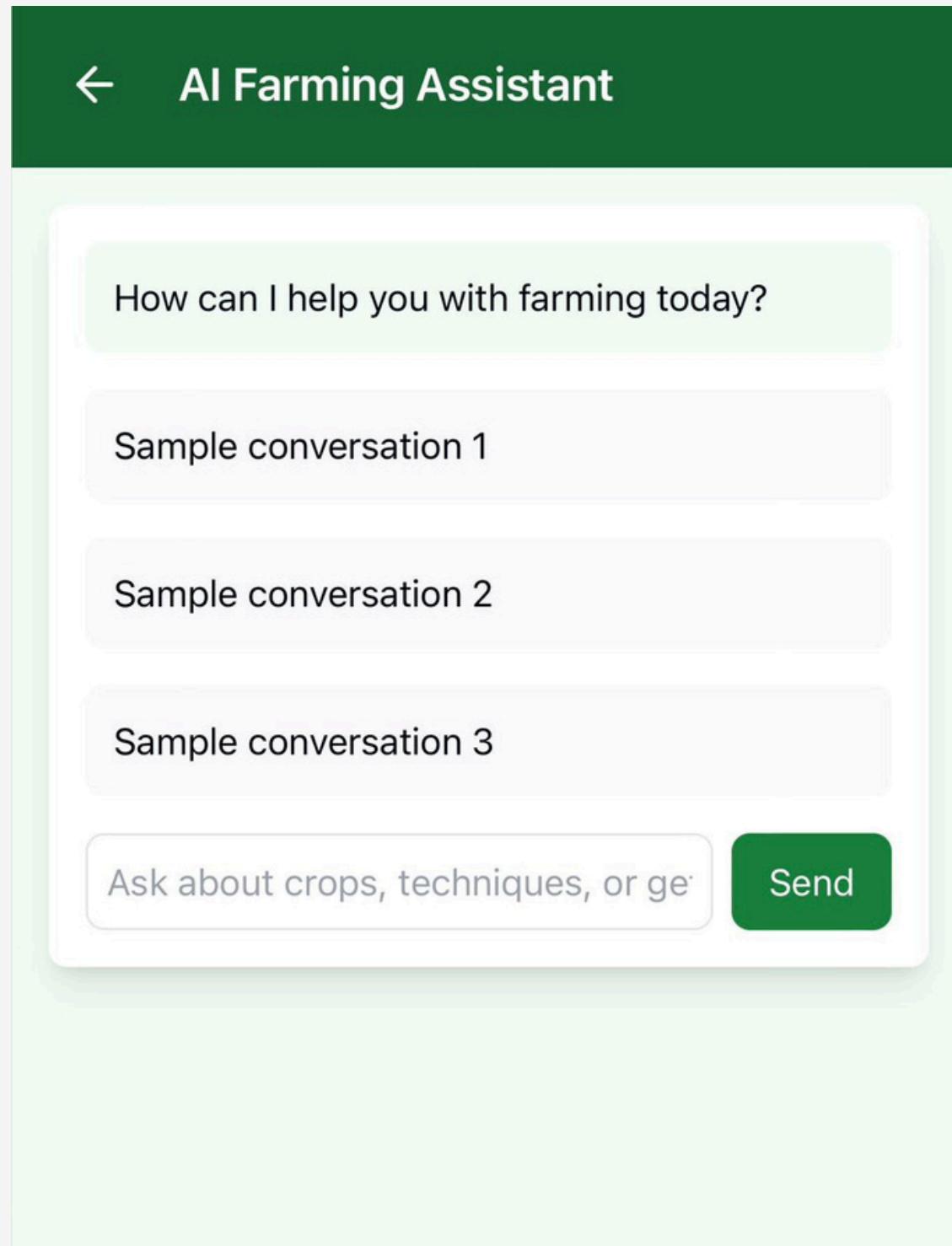
```
my-app > src > components > JS PestDetectionPage.js > ...
1 import React from 'react';
2 import { Card,CardContent } from './ui/card'; // Ensure this path is correct
3 import { Camera } from 'lucide-react';
4
5 const PestDetectionPage = ({ onBack }) => {
6   return (
7     <div className="min-h-screen bg-gradient-to-br from-green-800 to-brown-600 p-8">
8       <button onClick={onBack} className="bg-green-700 text-white px-6 py-3 rounded-lg mb-4">
9         Back
10        </button>
11        <div className="flex justify-center items-center">
12          <Card>
13            <CardContent className="p-4 flex flex-col items-center">
14              <div className="w-full h-64 bg-gray-100 rounded-lg flex items-center justify-center mb-4">
15                <Camera size={48} className="text-gray-400" />
16              </div>
17              <button className="bg-green-700 text-white px-6 py-3 rounded-lg">
18                Open Camera
19              </button>
20              <p className="mt-4 text-gray-600 text-center">
21                Point your camera at the affected area to detect pests and get treatment recommendations
22              </p>
23            </CardContent>
24          </Card>
25        </div>

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS

Front end code for pest detection page.

CURRENT PROGRESS



- Suggests the best crops to grow based on soil type, weather, and season.
- Offers guidance on planting, fertilization, irrigation, and pest management..
- Provides solutions to common farming issues using AI-powered insights..

The screenshot shows a code editor interface with a dark theme. The top navigation bar includes tabs for "Release Notes: 1.96.2", "requirements.txt", and the active file "ChatbotPage.js". Below the navigation bar, the file path "my-app > src > components > ChatbotPage.js" is displayed. The code itself is a React component named "ChatbotPage". It imports "React" and "PageHeader" from "src/components". The component returns a JSX structure with a header, a main content area containing a sample conversation, and a text input field at the bottom.

```
1 import React from 'react';
2 import PageHeader from './PageHeader';
3
4 const ChatbotPage = ({ onBack }) => {
5   return (
6     <div className="min-h-screen bg-green-50">
7       <PageHeader title="AI Farming Assistant" onBack={onBack} />
8       <div className="max-w-2xl mx-auto p-4 space-y-4">
9         <div className="bg-white p-4 rounded-lg shadow-lg space-y-4">
10          <div className="bg-green-50 p-3 rounded-lg">
11            | How can I help you with farming today?
12          </div>
13          <div className="space-y-4">
14            {[1, 2, 3].map((msg) => (
15              <div key={msg} className="p-3 bg-gray-50 rounded-lg">
16                | Sample conversation {msg}
17              </div>
18            )))
19          </div>
20          <div className="flex gap-2">
21            <input
22              type="text"
23              placeholder="Ask about crops, techniques, or get advice..."
24              className="flex-1 p-2 border rounded-lg"
25            />
26        </div>
27      </div>
28    </div>
29  );
30}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Front end code for chatbot page.

CURRENT PROGRESS

← Farming News

⚠ Misinformation Detection Active
News content is being verified for accuracy

Latest Farming Update 1

Detailed news about sustainable farming practices, market updates, and agricultural innovations...

Latest Farming Update 2

Detailed news about sustainable farming practices, market updates, and agricultural innovations...

Latest Farming Update 3

Detailed news about sustainable farming

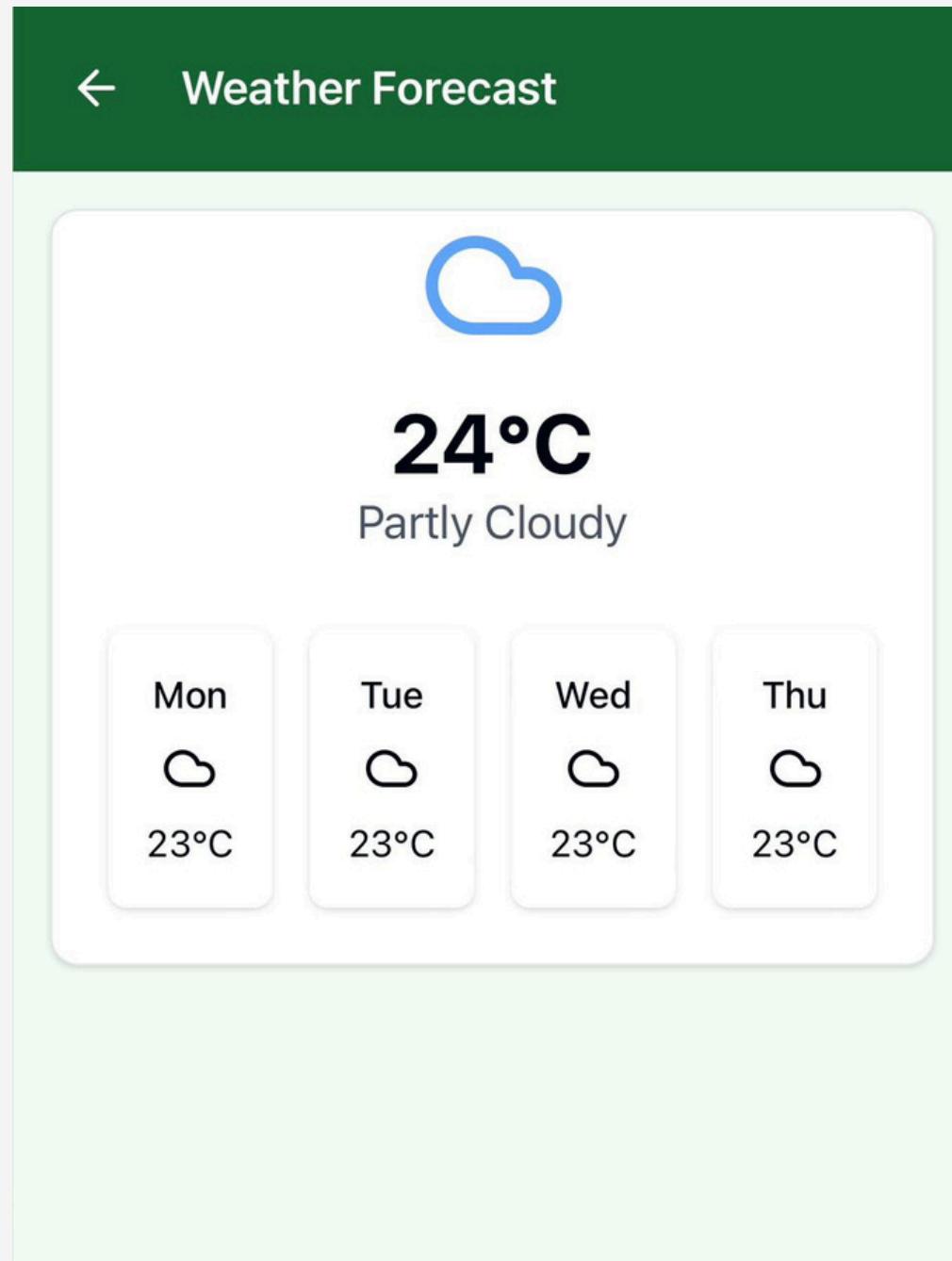
- Delivers the latest articles on farming techniques, market trends, government policies, and global agricultural events.
- Provides region-specific news for farmers to stay informed about local farming practices and regulations.
- Includes interviews and insights from agricultural experts and scientists.

```
my-app > src > components > JS NewsPage.js > ...
  1  import React from 'react';
  2  import PageHeader from './PageHeader';
  3  import { Alert, AlertTitle, AlertDescription } from './ui/alert';
  4  import { AlertTriangle } from 'lucide-react';
  5  import { Card,CardContent } from './ui/card';
  6
  7  const NewsPage = ({ onBack }) => {
  8    return (
  9      <div className="min-h-screen bg-green-50">
 10        <PageHeader title="Farming News" onBack={onBack} />
 11        <div className="max-w-2xl mx-auto p-4 space-y-4">
 12          <Alert variant="warning" className="bg-amber-50">
 13            <AlertTriangle className="h-4 w-4" />
 14            <AlertTitle>Misinformation Detection Active</AlertTitle>
 15            <AlertDescription>
 16              News content is being verified for accuracy
 17            </AlertDescription>
 18          </Alert>
 19          {[1, 2, 3, 4, 5].map(item) => (
 20            <Card key={item}>
 21              <CardContent className="p-4">
 22                <h3 className="font-medium text-lg mb-2">Latest Farming Update {item}</h3>
 23                <p className="text-gray-600">
 24                  Detailed news about sustainable farming practices, market updates, and agricultural i
 25            </Card>
 26          )}
 27        </div>
 28      </div>
 29    );
 30  }
 31
 32  export default NewsPage;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Front end code for news detection page

CURRENT PROGRESS



- Displays real-time weather conditions, such as temperature, humidity, and wind speed.
- Provides detailed weather predictions for the upcoming days, including rainfall chances and temperature variations.
- Notifies farmers of extreme weather events like storms, heavy rainfall, or drought conditions..
- Offers weather data specific to the farmer's region or selected location.

```
my-app > src > components > JS WeatherPage.js > ...
1 import React from 'react';
2 import PageHeader from './PageHeader';
3 import { cloud, sun, cloudRain } from 'lucide-react';
4
5 const (property) React.JSX.IntrinsicElements.div:
6   ret React.DetailedHTMLProps<React.HTMLAttributes<HTMLDivElement>, HTMLDivElement>
7     <div className="min-h-screen bg-blue-50">
8       <PageHeader title="Weather Updates" onBack={onBack} />
9       <div className="max-w-2xl mx-auto p-4 space-y-4">
10         <div className="flex items-center gap-4 bg-white p-4 rounded-lg shadow-md">
11           <Sun size={48} className="text-yellow-500" />
12           <div>
13             <h2 className="text-lg font-semibold">Current Weather</h2>
14             <p className="text-gray-600">Sunny, 28°C</p>
15           </div>
16         </div>
17         <div className="flex items-center gap-4 bg-white p-4 rounded-lg shadow-md">
18           <Cloud size={48} className="text-gray-400" />
19           <div>
20             <h2 className="text-lg font-semibold">Tomorrow</h2>
21             <p className="text-gray-600">Cloudy, 25°C</p>
22           </div>
23         </div>
24         <div className="flex items-center gap-4 bg-white p-4 rounded-lg shadow-md">
25           <Cloud size={48} className="text-gray-400" />
26           <div>
27             <h2 className="text-lg font-semibold">Next Week</h2>
28             <p className="text-gray-600">Partly Cloudy, 27°C</p>
29           </div>
30         </div>
31       </div>
32     </div>
33   </div>
34 </b>
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS

Front end code weather forcast page

CURRENT PROGRESS

← Community Chat

 Safe Chat Active
Messages are being monitored for misinformation

Farmer 1
Discussion about sustainable farming practices...

Farmer 2
Discussion about sustainable farming practices...

Farmer 3
Discussion about sustainable farming practices...

- Participate in topic-specific discussions related to farming techniques, crop management, pest control, and more..
- Ask questions and get answers from experienced farmers or agricultural experts.
- Share tips, success stories, and best practices with the community.
- Communicate in preferred regional languages for easy understanding.

```
my-app > src > components > JS CommunityPage.js > ...
1 import React, { useState } from 'react';
2 import PageHeader from './PageHeader';
3
4 const CommunityPage = ({ onBack }) => {
5   const [messages, setMessages] = useState([
6     { user: 'Farmer Joe', text: 'How do you manage pests in your field?' },
7     { user: 'Sarah', text: 'I use organic pesticides; they work great for me!' },
8   ]);
9
10  const [inputMessage, setInputMessage] = useState('');
11
12  const handleSend = () => {
13    if (inputMessage.trim()) {
14      setMessages([...messages, { user: 'You', text: inputMessage }]);
15      setInputMessage('');
16    }
17  };
18
19  return (
20    <div className="min-h-screen bg-gray-50">
21      <PageHeader title="Community Chat" onBack={onBack} />
22      <div className="max-w-2xl mx-auto p-4 space-y-4">
23        <div className="bg-white p-4 rounded-lg shadow-md h-96 overflow-y-auto space-y-2">
24          {messages.map((message, index) => (
25            <div key={index} className="flex justify-between items-end">
26              <div>{message.user}: {message.text}</div>
27              <div>...</div>
28            </div>
29          ))}
30        </div>
31        <div>
32          <input type="text" value={inputMessage} onChange={(e) => setInputMessage(e.target.value)} />
33          <button onClick={handleSend}>Send</button>
34        </div>
35      </div>
36    </div>
37  );
38}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Front end code for community chat page

DATASET

1. Chatbot for Farmers:

- AGROVOC Thesaurus and Agricultural Question Dataset.
- OpenWeather API for weather data.

2. Resource Optimization and Pest Management:

- USDA NASS and Sentinel-2 provide extensive agricultural and satellite data, used by many agricultural researchers.
- PlantVillage is an established dataset for plant disease and pest identification.

DATASET

3. News Feed with Fake News Detection:

- LIAR and FakeNewsNet are datasets for fake news research.
- AgFunder articles are publicly available for training on credible agricultural news.

4. Chatting Facility with Misinformation Detection:

- MisinfoMe and PHEME datasets are used for misinformation detection, with labeled conversation data.
- MS-MARCO by Microsoft is used for misinformation research.

TIME PLAN

- 3 months: Research and data collection
- 3 months: Development
- 2 months: Testing and feedback
- 1 month: Deployment and presentation prep
- Maintenance Plan: Regular updates; model retraining every 6 months.

CONCLUSION

- The Smart Farming platform aims to revolutionize agriculture by providing data-driven insights and tools that enhance productivity and sustainability.
- Offers tools for resource optimization, pest management, and yield predictions.
- Integrates government schemes for farmers with an intuitive, user-friendly interface.
- Empowers farmers to adopt sustainable practices through data-driven recommendations.

REFERENCES

- [1] Gonzalez, J., et al. (2023). "AI-Powered Decision Support Systems for Sustainable Agriculture: A Review."
- [2] López, J., et al. (2023). "Enhancing Farmer Engagement through AI Chatbots: A Study on Usability and Effectiveness."
- [3] Zhang, T., et al. (2023). "Smart Pest Management: Leveraging IoT and AI Technologies."
- [4] Patel, R., et al. (2022). "Combatting Misinformation in Agriculture: A Machine Learning Approach."
- [5] Verma, A., & Singh, R. (2023). "Evaluating Government Schemes for Agricultural Development: A Systematic Review."

THANK YOU!