Implementation of Third Module

Weekly Date

21/10/24 to 26/10/24

XGBoost Algorithm Integration and Full-Stack Application Development:

Objective:

The goal of this module is to enhance model accuracy by implementing the XGBoost algorithm and to deploy the model using a full-stack architecture. This includes frontend development (web/mobile), backend API creation, secure authentication, and dynamic report generation.

Machine Learning Model Used:

XGBoost Classifier

- **Purpose:** To predict optimized fertilizer usage based on soil type, weather data, and other agricultural inputs.
- **Tool Used:** XGBoost library in Python.
- **Training Data:** Pre-processed dataset with agricultural and weather parameters.
- **Accuracy Achieved:** 74.00% (improved over Decision Tree and Random Forest models).
- **Reason for Selection:**
    - High accuracy
    - In-built regularization to avoid overfitting
    - Faster training and prediction time
    - Suitable for structured tabular data

Technologies Used:

Component	Technology	Purpose
Frontend	React (Web) / Flutter (App)	For user interaction and input submission
Backend	Python (Flask or Django)	To handle requests and serve ML predictions via REST API
ML Algorithm	XGBoost (scikit-learn)	To make accurate predictions for fertilizer usage
Database	Firebase or SQLite	To store user data, prediction logs, and history
Weather API	OpenWeatherMap API	To fetch real-time weather data based on user's location
Authentication	Firebase Authentication / Auth0	To verify and manage user access
Security	JWT (JSON Web Token)	To secure data transmission between frontend and backend
Report Generator	ReportLab / PDFKit (Python)	To generate downloadable PDF reports of prediction results

Workflow Description:

1. **User Authentication:**
    - User signs in or registers via Firebase or Auth0.
    - A secure JWT token is issued to maintain session integrity.
2. **User Input:**
    - Users enter details such as crop type, soil type, and location through the frontend (React or Flutter).
3. **Weather Data Fetching:**
    - Based on the location, weather data (temperature, humidity, etc.) is fetched from the OpenWeatherMap API.

4. **Prediction Generation:**
    - The backend uses the trained XGBoost model to generate fertilizer usage recommendations.
    - Predictions are calculated in real-time based on both static and dynamic inputs.
5. **Data Storage:**
    - All inputs and predictions are stored in Firebase (cloud) or SQLite (local) for tracking and analytics.
6. **PDF Report Generation:**
    - A personalized report is created using ReportLab or PDFKit.
    - Includes prediction summary, user inputs, weather data, and timestamp.
    - Users can download the report from the frontend.

Key Features of the System:

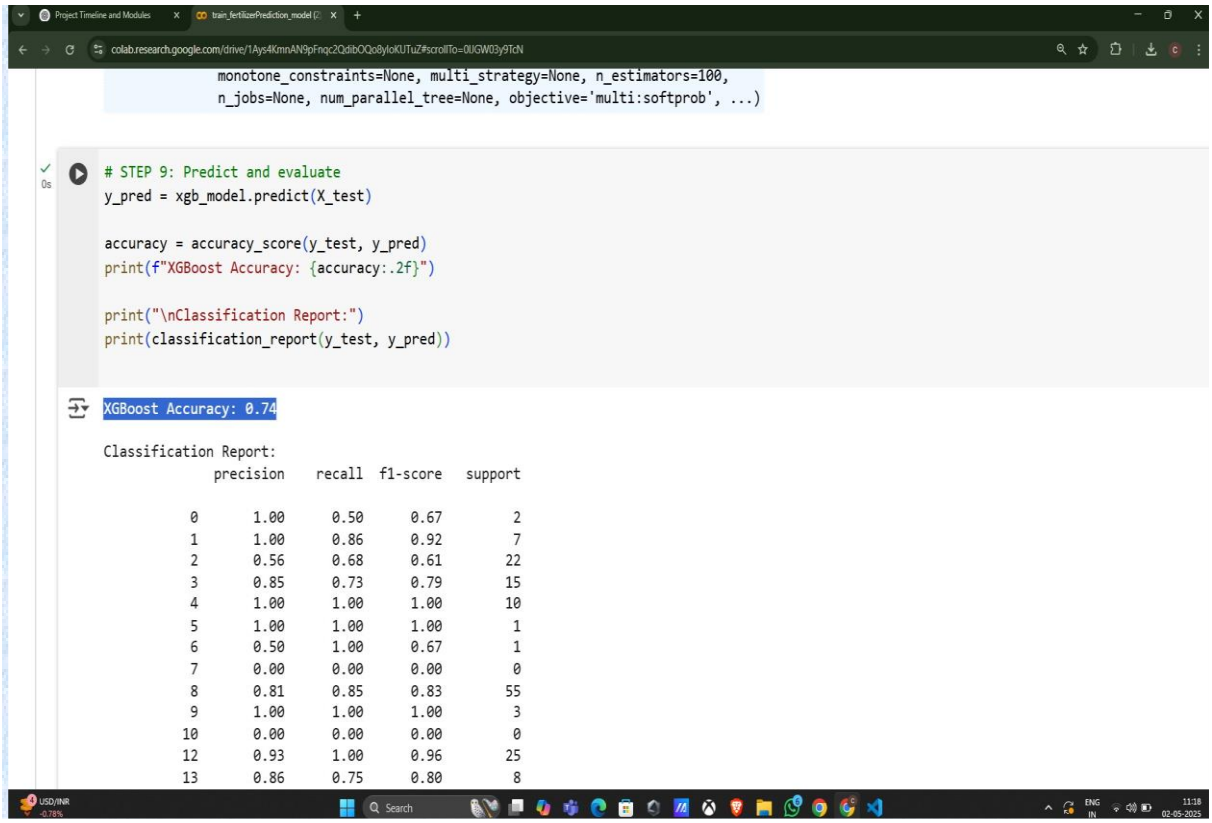
- Secure login system with Firebase/Auth0
- Accurate predictions using XGBoost (74% accuracy)
- Context-aware analysis with real-time weather integration
- User-specific PDF report generation
- Responsive frontend on web (React) or mobile (Flutter)
- JWT-based security for API and data access

Outcomes of Module 3:

- Improved prediction reliability with XGBoost.
- Fully functional backend API integrated with a secure frontend.
- Cloud-compatible, scalable solution using Firebase or portable SQLite**.**
- End-to-end automation of data input, processing, output, and reporting.
- Foundation ready for deployment and further enhancements.

Conclusion:

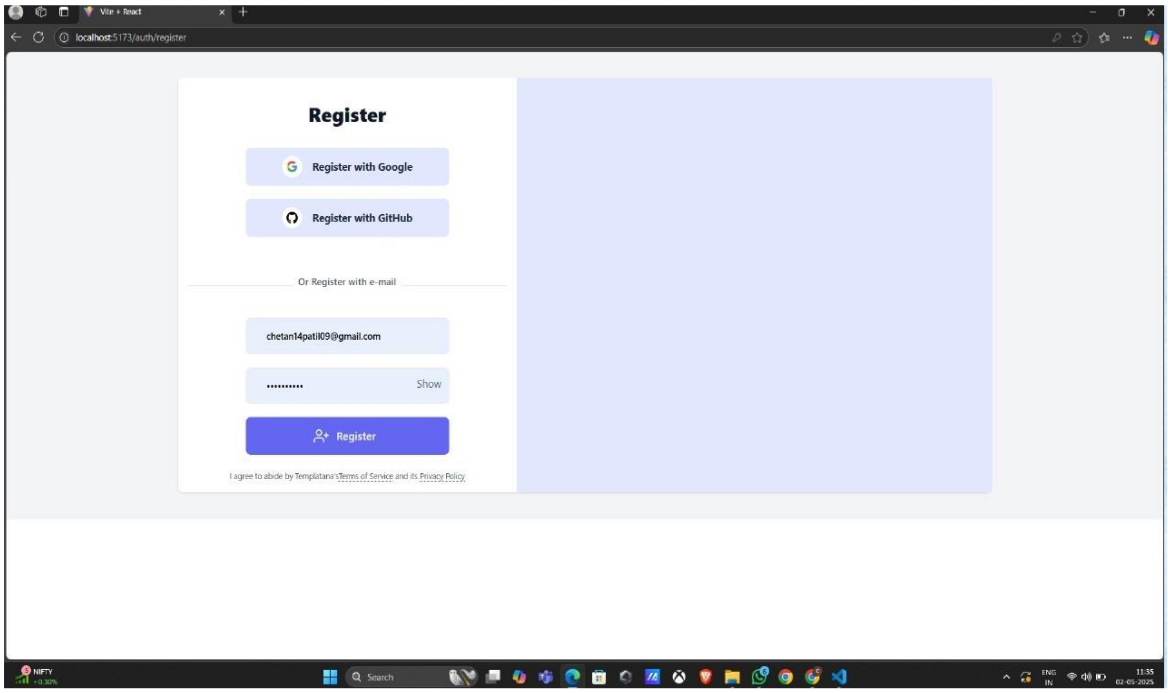
Module 3 marks a significant step in making the project deployable, secure, and user-friendly. By integrating XGBoost with a full-stack application, we achieved both technical accuracy and practical usability. This paves the way for final deployment and real-time field use.

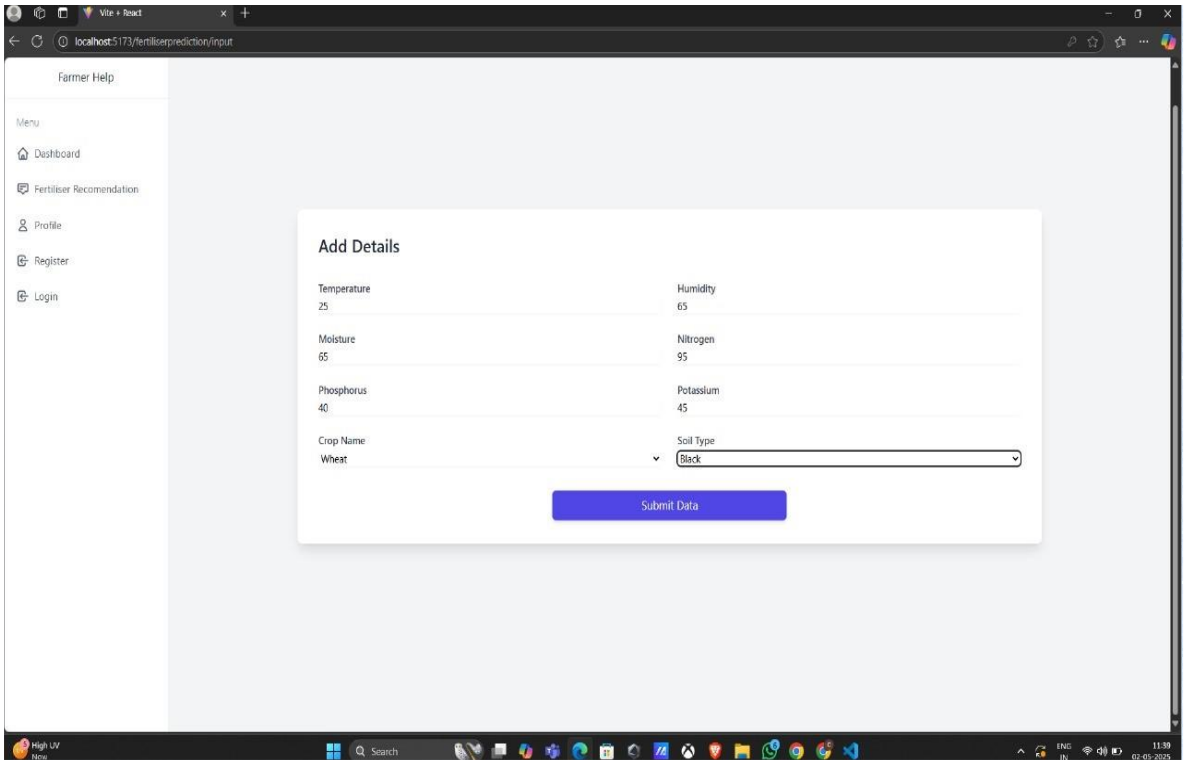


```
                monotone_constraints=None, multi_strategy=None, n_estimators=100,
                n_jobs=None, num_parallel_tree=None, objective='multi:softprob', ...)
```

```
# STEP 9: Predict and evaluate
y_pred = xgb_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"XGBoost Accuracy: {accuracy:.2f}")

print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
XGBoost Accuracy: 0.74

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       1.00      0.86      0.92         7
           2       0.56      0.68      0.61        22
           3       0.85      0.73      0.79        15
           4       1.00      1.00      1.00        10
           5       1.00      1.00      1.00         1
           6       0.50      1.00      0.67         1
           7       0.00      0.00      0.00         0
           8       0.81      0.85      0.83        55
           9       1.00      1.00      1.00         3
          10       0.00      0.00      0.00         0
          12       0.93      1.00      0.96        25
          13       0.86      0.75      0.80         8
```


Vite + React
localhost:5173/auth/register
Register
Register with Google
Register with GitHub
Or Register with e-mail
chetan14patil09@gmail.com
Show
Register
I agree to abide by Templatana's Terms of Service and its Privacy Policy.


Vite + React
localhost:5173/fertiliserprediction/input
Farmer Help
Menu
Dashboard
Fertiliser Recomendation
Profile
Register
Login
Add Details
Temperature
25
Humidity
65
Moisture
65
Nitrogen
95
Phosphorus
40
Potassium
45
Crop Name
Wheat
Soil Type
Black
Submit Data