IOT and Reinforcement Learning-Based Smart Irrigation and Crop Recommendation System

D. Vaishnavi[1]; P. Lakshmi Indira[2]; R. Sree Vishnu Priya[3]; V. Manasa[4]; V. Roja[5]

[1]Assistant Professor, CSE, [2,3,4,5]B. Tech 4th Year Student
[1,2,3,4,5] CSE, Vignan's Institute of Management and Technology for Women, Hyderabad, India.

Publication Date: 2026/04/21

Abstract: Efficient water use significantly contributes to sustainable agricultural development. Common methods of irrigation using flooding techniques or timers do not change based on environmental conditions and therefore either over-irrigate or under-irrigate the crop and do not allow for proper growth. The objective of this research paper is to propose a methodology for farming by using Internet of Things (IOT), machine learning and reinforcement learning. Sensors connected to an ESP32 microcontroller gather information like moisture level, temperature, humidity and rainfall. The collected data will be sent to a Flask application hosted in the cloud for analyzing and processing. An IOT framework using a Q-Learning algorithm takes input which identifies appropriate action and time for irrigation. The crop recommendation system also suggests what type of crops you should grow as per the type of soil and climatic conditions. All the data is saved in the cloud using a MongoDB atlas database, and it is showcased on a web-based dashboard for easy usage. Using such a mechanism will not only save time and water but will also ensure smart farming.

Keywords: *Reinforcement Learning, Adaptive Irrigation Management, ESP32 Microcontroller-Based Edge Computing IoT, Cloud-Based Flask Application with the Mongodb Atlas Database.*

How to Cite: D. Vaishnavi; P. Lakshmi Indira; R. Sree Vishnu Priya; V. Manasa; V. Roja (2026) IOT and Reinforcement Learning-Based Smart Irrigation and Crop Recommendation System. *International Journal of Innovative Science and Research Technology*, 11(4), 1351-1361. https://doi.org/10.38124/ijisrt/26apr1285

I. INTRODUCTION

Agriculture is essential for sustainability of food security as well as the economy of the world. However, it is a huge challenge today. With consistent shortage of water and non-efficient usage of resources, it has become impossible to meet the requirements of more than 7 billion people in the world. Flood irrigation and other conventional irrigation methods like operating timers do not take into account real-time changes in the environment thus causing the misusage of the water. Water wastage leads to degradation of quality of land and yields not only low but also less crop. With the emergence of digital technologies, the adoption of IoT in agriculture allows the automated monitoring of environmental parameters like soil moisture, temperature, humidity and rainfall. Even though the real-time data provided by the IoT solutions is available, a lot of systems in existence use fixed threshold mechanisms.

To tackle these challenges, the paper proposes the use of the intelligent smart farming system, which employs IoT along with ML and RL techniques for effective irrigation and crop recommendation. The system uses a Q-learning-based RL model to find optimal irrigation actions based on real-time sensor inputs, and a machine learning module recommends suitable crops based on soil and climatic data. Data acquisition is performed by an ESP32 microcontroller, processing is done using a cloud-based Flask backend, and a web dashboard is used for real-time monitoring and visualization. The main contributions include the development of an adaptive irrigation system, the integration of RL for decision-making, the establishment of a crop recommendation module, and the implementation of a cloud-based framework for data storage and analysis. The proposed system aims to improve water efficiency, reduce manual work and foster sustainable agricultural practices.

II. LITERATURE REVIEW

Several research works are conducted to enhance the agriculture irrigation efficiency and productivity using IoT and machine learning methods.

In 2022, an IoT-based smart irrigation system was developed by Yohannes Tace, Mohamed Tabaa and Sanaa Elfilali [1] for automatic pump control, which evaluates several machine learning algorithms. Water management via real-time monitoring: 98.3% of accuracy for the K-Nearest Neighbours (KNN) model results

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

In 2024, S. Roy et al. [2] suggested an intelligent system of irrigation based on decision tree and random forest methods, which has reached an accuracy of 98.7% in predicting soil characteristics and improving scheduling.

In 2022, K. Premkumar et al. [3] presented their edge-based computing system, which relies on a combination of various algorithms for the IoT-driven prediction of soil moisture and is characterized by very low latency.

A cloud-based smart irrigation system that integrates sensor information and weather forecast has been proposed by R. Reddy et al. [4] in 2024. This system was able to reduce the consumption of water to 40% without compromising the productivity of crops.

A simple IoT based smart irrigation system for remote control has been implemented in 2024 using NodeMCU and Blynk by B. Manjula [5]. Alerts in the system reduced the requirement of manual supervision.

A reinforcement learning based irrigation strategy has been proposed using Q-Learning technique in 2025 by Z. Zhang et al. [6] to respond to dynamic environments.

The use of Internet of Things and machine learning in the context of irrigation system has been comprehensively discussed in a review article by F. Nsoh et al. [7] **in 2024**. The main issues associated with interoperability and security of the smart farming systems have been emphasized in the review.

Fog computing-based smart irrigation system has been proposed using LoRaWAN in 2023 by **Rajasekhar et al. [8]**. Machine learning is used to predict rainfall and resulted in around 60% water savings over traditional irrigation methods.

III. PROPOSED SYSTEM

A. System Architecture:

The proposed Smart Irrigation System can be implemented using the four-layer IoT-based system architecture that makes use of environment sensors, cloud computing, reinforcement learning decision engine, crop recommendation through machine learning, and web-based visual interface. This will be possible by using ESP32 hardware along with the Flask back-end, MongoDB Atlas database, and React dashboard. The following are the different layers of the IoT smart irrigation system:

- Hardware/Field Layer – Layer 1
- Flask Back-End (AI Decision Engine) – Layer 2
- MongoDB Atlas (Cloud Database) – Layer 3
- React Dashboard (User Interface) – Layer 4

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

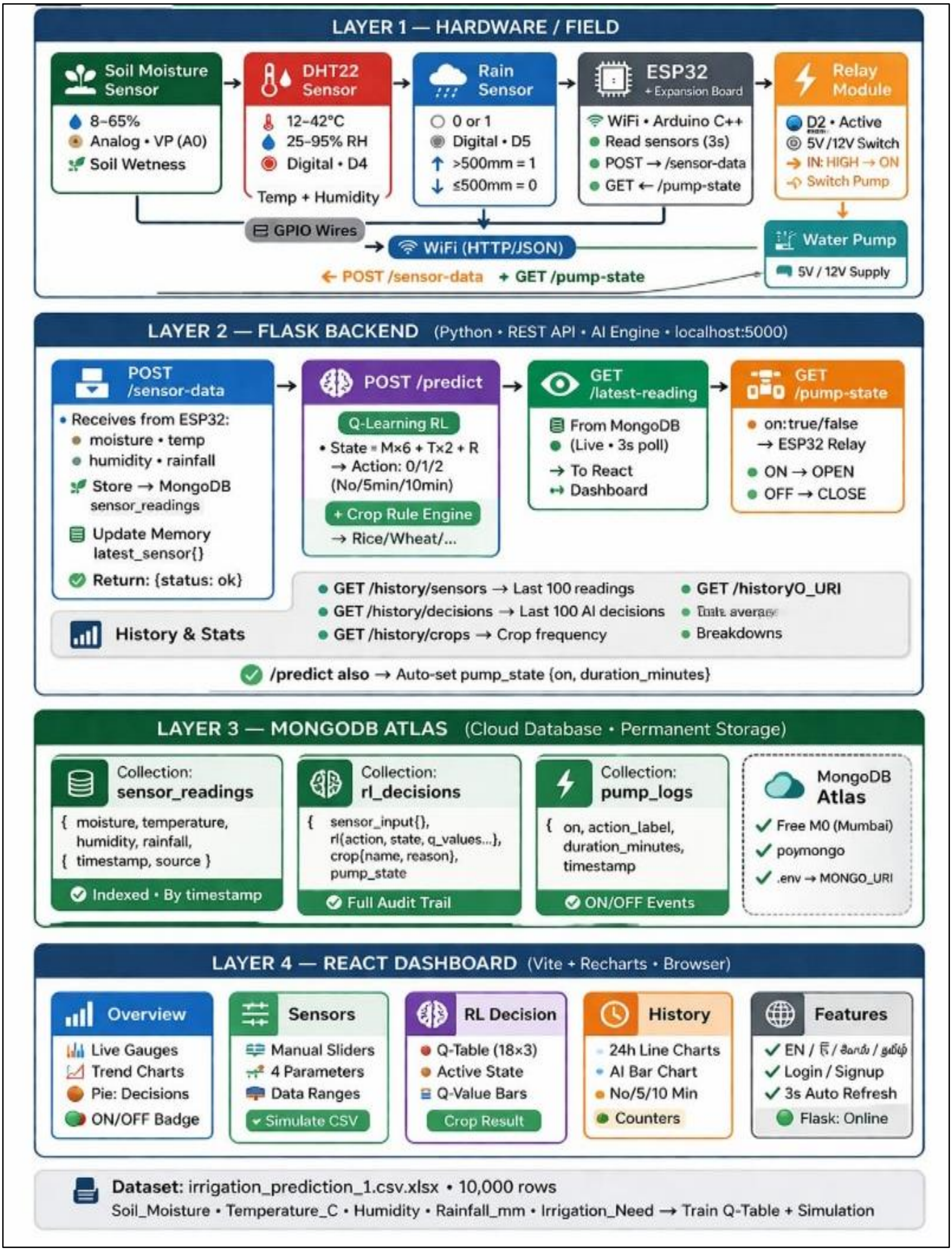

Fig 1 System Architecture

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

B. Layer 1 – Hardware / Field Layer

The hardware layer will include environmental sensing capabilities in real-time, basic control functions, and the connectivity capability to the cloud computing infrastructure. The hardware layer includes the presence of the soil moisture sensor to determine soil conditions related to moisture and facilitate irrigation requirements calculation. In addition, DHT22 sensors measure the ambient air temperature and relative humidity to estimate evapotranspiration levels and hence the amount of water that the crops need. Further, a rain sensor identifies rain incidents and disables the irrigation process. This set of hardware sensors will be linked to the ESP32 microcontroller to take advantage of the inbuilt wifi capabilities and excellent processing power. The ESP32 gathers sensor data and formats it in JSON form before transmitting the information securely using HTTPS protocol to the backend servers. In addition, the ESP32 receives commands regarding pump operation from the backend server and acts based on them by controlling the relay module attached to a 5V/12V DC water pump.

C. Layer 2 – Flask Backend (RL and Recommendation Engine)

The backend layer makes use of Flask programming and is deployed in the cloud server to enable remote access and scalability. It offers REST API interfaces to accept sensor inputs, calculate irrigation actions, and generate crop recommendations. After accepting the environmental inputs, the backend will store the information and process the data based on the Q-learning algorithm of reinforcement learning. This allows the algorithm to determine the appropriate course of action by assessing the prevailing environmental state in order to optimize water usage. In addition to calculating the irrigation actions, there is also a crop recommendation function that determines the appropriate crops to plant based on environmental factors and the pre-set requirement datasets of various crops.

D. Layer 3 – MongoDB Atlas (Cloud Database Layer)

The information will be stored in a persistent scalable manner in the cloud database using the MongoDB Atlas Database. This involves storing sensor values, irrigation suggestions, status of operation of the pumps, and crop recommendation results. The time stamps are used in recording the data so that it can be easily traced in future. The database will allow secure storage, querying of information, and integration with both the front-end and the back-end. Evaluation of previous models can also be done through the database.

E. Layer 4 – React Dashboard (User Interface Layer)

The frontend is implemented with the React framework and forms the user interface of the system. The dashboard will provide live visualizations of the data coming from sensors, pumps, and reinforcement learning model along with crop recommendation. The data will be fetched using REST API calls and displayed on the dashboard using graphical elements to provide better interpretation. Environmental data trends and irrigation activities will be shown in the form of graphs to help monitor system behavior.

IV. IMPLEMENTATION

A. Hardware Implementation

The hardware implementation involves using an ESP32 controller linked with a soil moisture sensor, a DHT22 sensor, and a rain sensor. These sensors are integrated with GPIO pins in order to obtain environmental data from the field. A relay is connected to the ESP32 and used to turn ON/OFF a 5V/12V water pump according to the pump control signal coming from the backend. The ESP32 employs WIFI connectivity to transmit sensor data in JSON format to the cloud server.

B. Backend Implementation

Flask framework in Python is employed for the implementation of the backend which is hosted by the cloud server. It offers several REST API end points for receiving data from sensors, generating irrigation decisions, and controlling the water pump. An algorithm of Q-Learning is employed to compute optimal irrigation duration under certain environmental conditions. All data received and decisions made are stored in a cloud-based database.

C. Database Implementation

The MongoDB atlas database has been used in this context for implementation. The various sensor values, reinforcement learning, and pumping logs are stored in separate collections. Time stamps have been added to all entries to allow for historical analysis. Data protection, scalability, and accessibility are ensured in real time through this database.

D. Frontend Implementation

The React technology framework has been used for implementing the frontend application. The various live sensor values, pumping system, and reinforcement learning are shown on the dashboard through visual representation. The frontend gets information from the backend through RESTful APIs in real time. Data analysis can be performed using the visualization technique.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

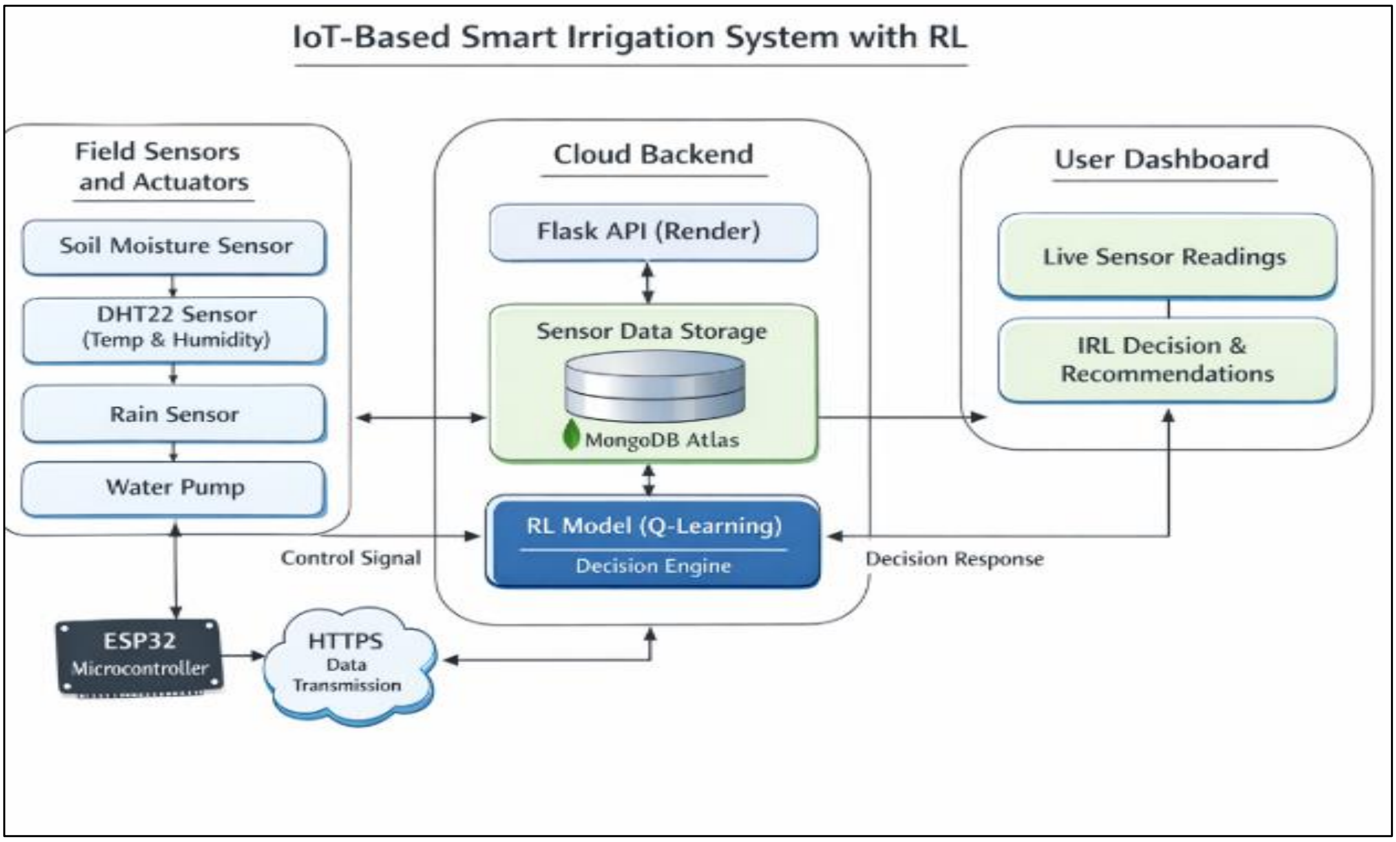

Fig 2 Implementation Details

V. ALGORITHM

A. Irrigation Algorithm using Reinforcement Learning

The proposed system uses a reinforcement learning algorithm that uses Q-learning for optimizing the process of irrigation. The steps used in the process are mentioned below:

➢ *Step 1: Initialization of Q-Table*

The first step involves initializing the Q table by giving the value zero for all state-action pairs. Define learning rate (α), discount factor (γ) and define the rewards.

➢ *Step 2: Current Sensor Data*

Collect the current data about the environmental factors such as soil moisture, temperature, humidity, and rainfall.

➢ *Step 3: Current State Identification*

Define the current states depending upon the values of sensors (like soil moisture level: dry/wet, temperature: high/low, rain: yes/no).

➢ *Step 4: Selecting an Action*

Depending upon the Q table, an action is selected as below:

- A0: Not Performing Irrigation Action
- A1: Irrigating for 5 minutes
- A2: Irrigating for 10 minutes

➢ *Step 5: Executing Action*

Depending upon the action selected in Step 4, the process of irrigation can be done as below:

- Pump ON (for irrigation)
- Pump OFF (if irrigation is not required)

➢ *Step 6: Calculate Reward*

After taking the required action, calculate the reward depending upon the below-mentioned criteria:

- Increase in soil moisture => positive reward
- Watering in excess or during rainfall => negative reward

➢ *Step 7: Update Q-value*

Update the Q-value according to the formula:

$$Q(s,a)=Q(s,a)+\alpha[r+\gamma maxQ(s\text{^}',a\text{^}')-Q(s,a)]$$

Where,

s = state

a = action

r = calculated reward

s' = state'

➢ *Step 8: Store Decisions*

Store the decisions taken by sensors in the database.

➢ *Step 9: Repeat Cycle*

Repetitively perform steps from 1 to 8.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

VI. RESULTS

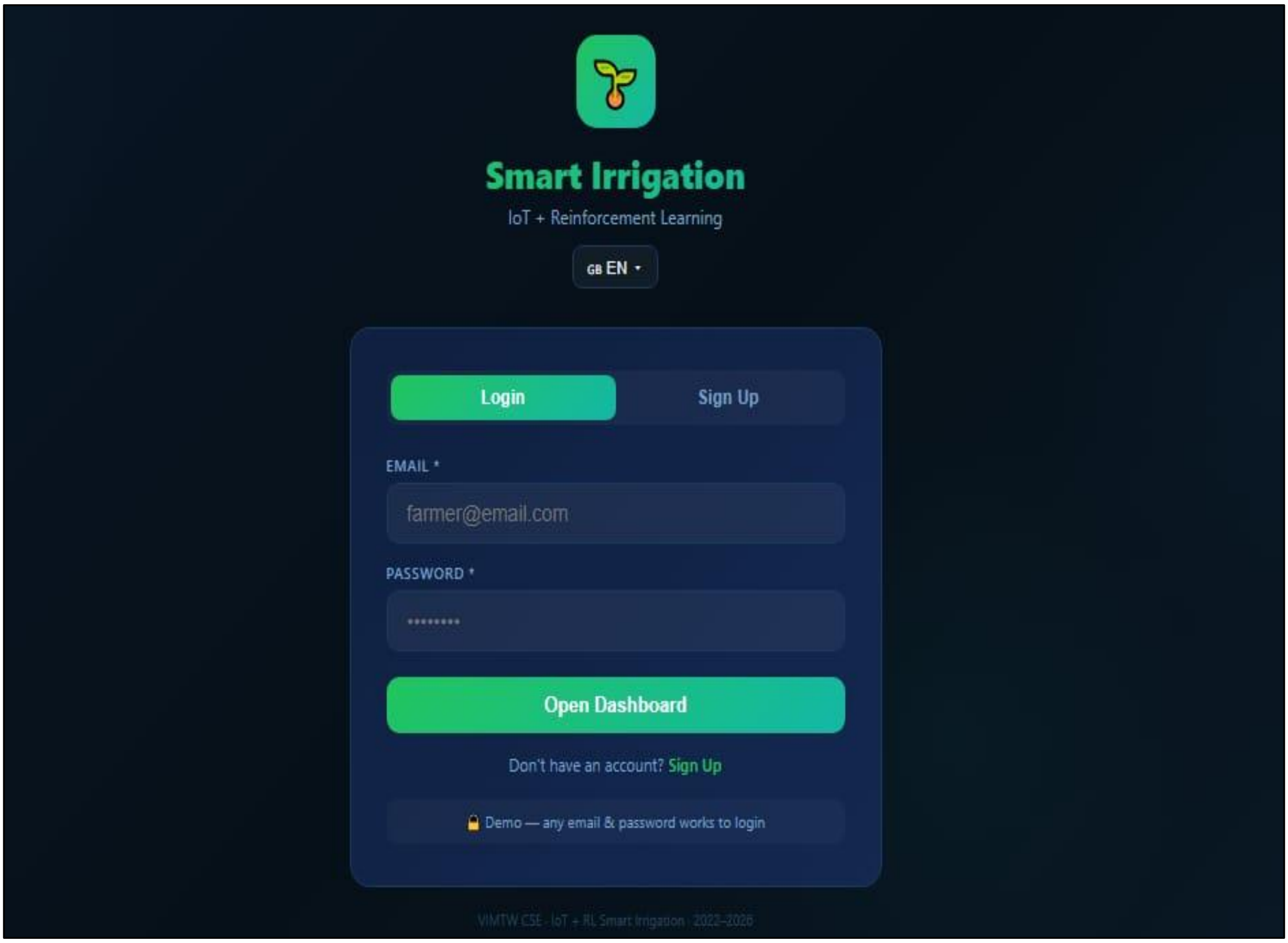

Fig 3 Login page

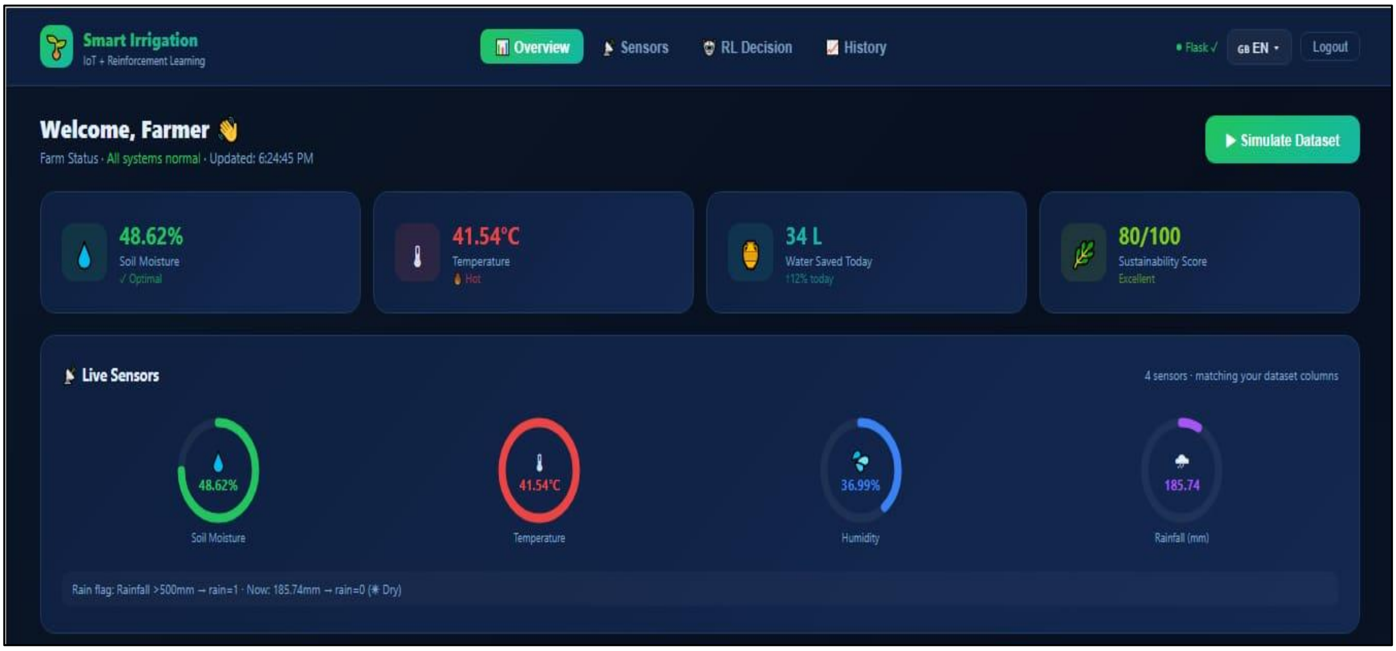

Fig 4 User Dashboard

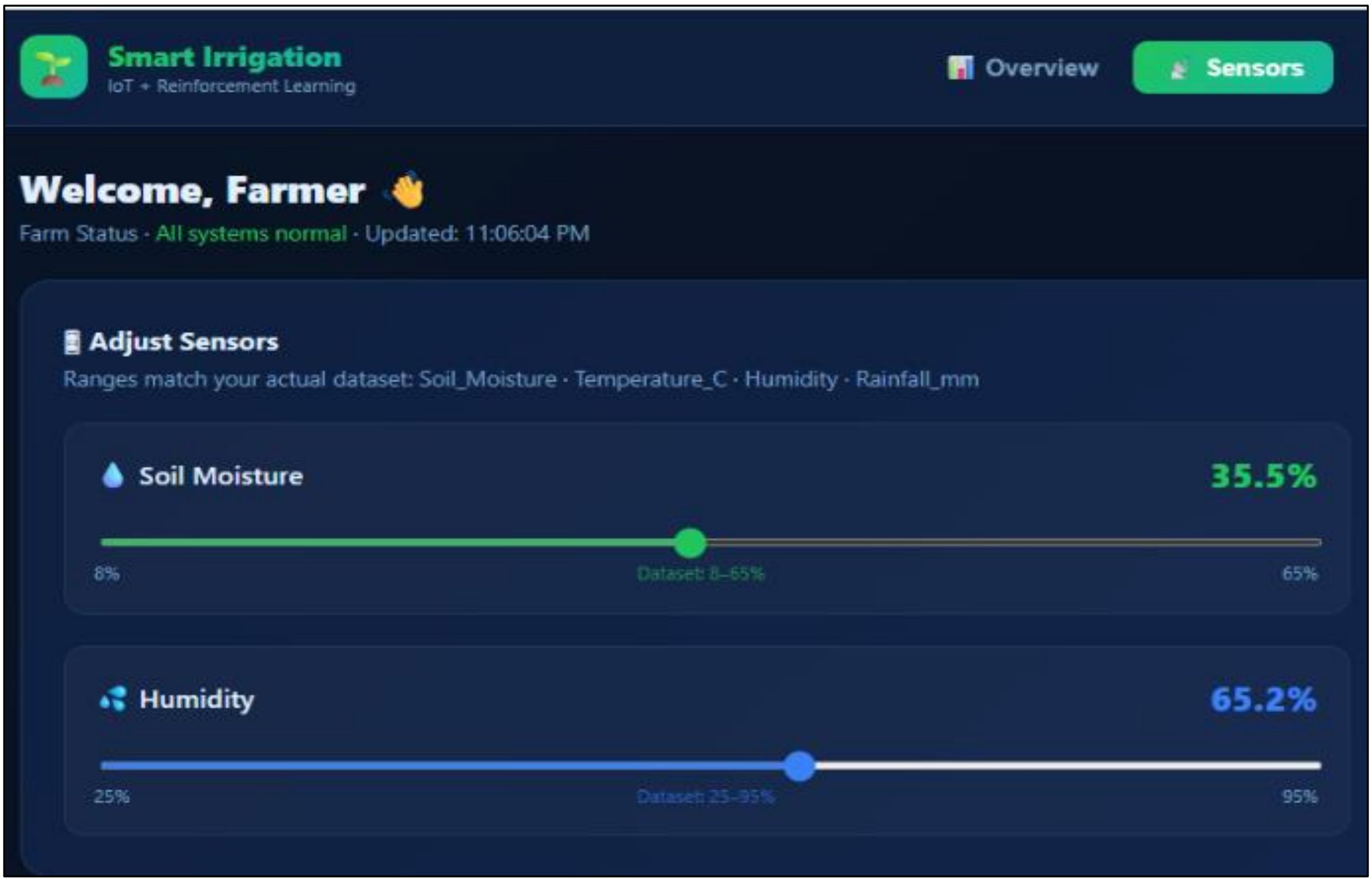

Fig 5 Soil & Humidity Sensor Readings

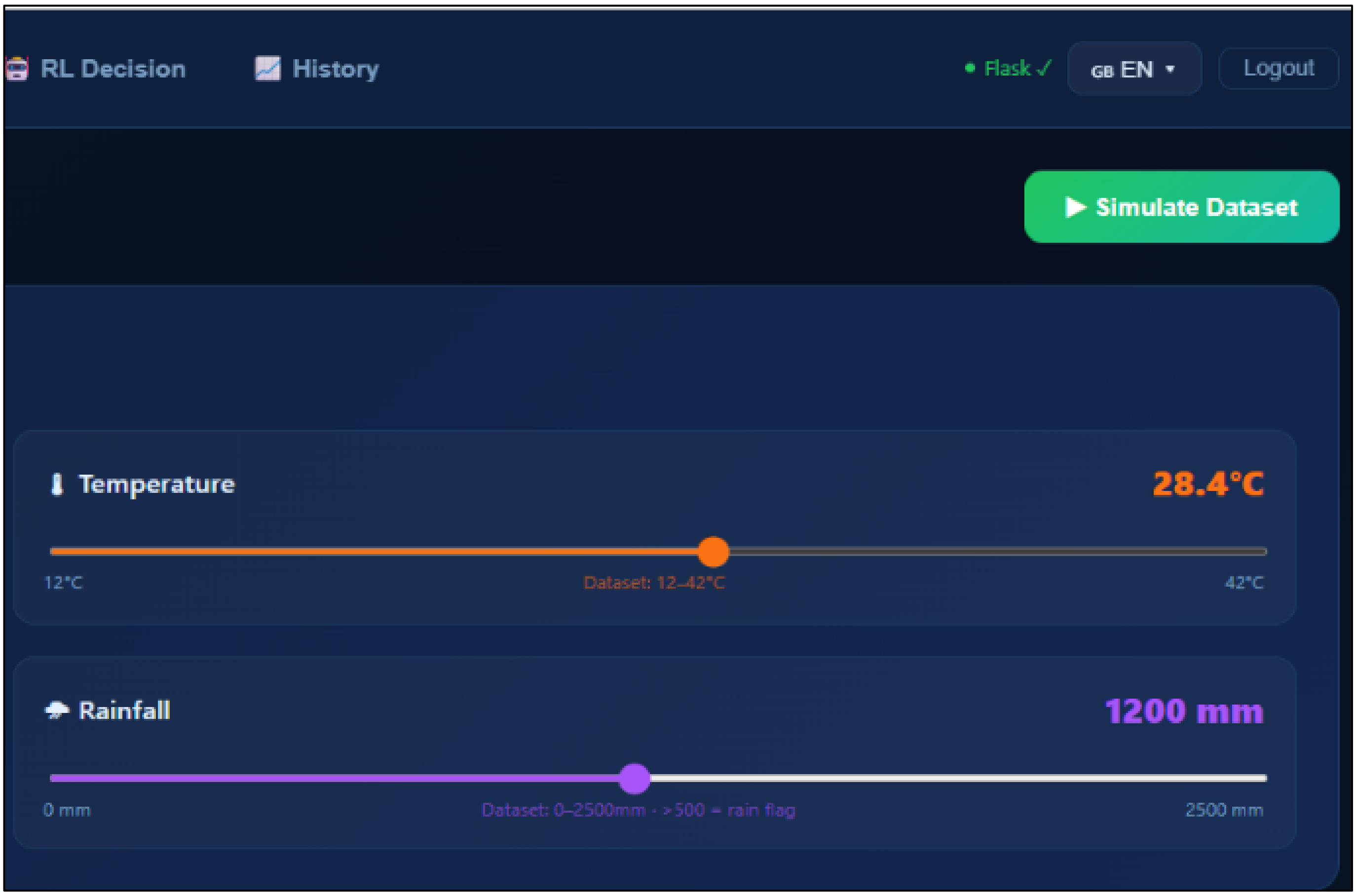

Fig 6 Temperature & Rainfall Sensor Readings

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

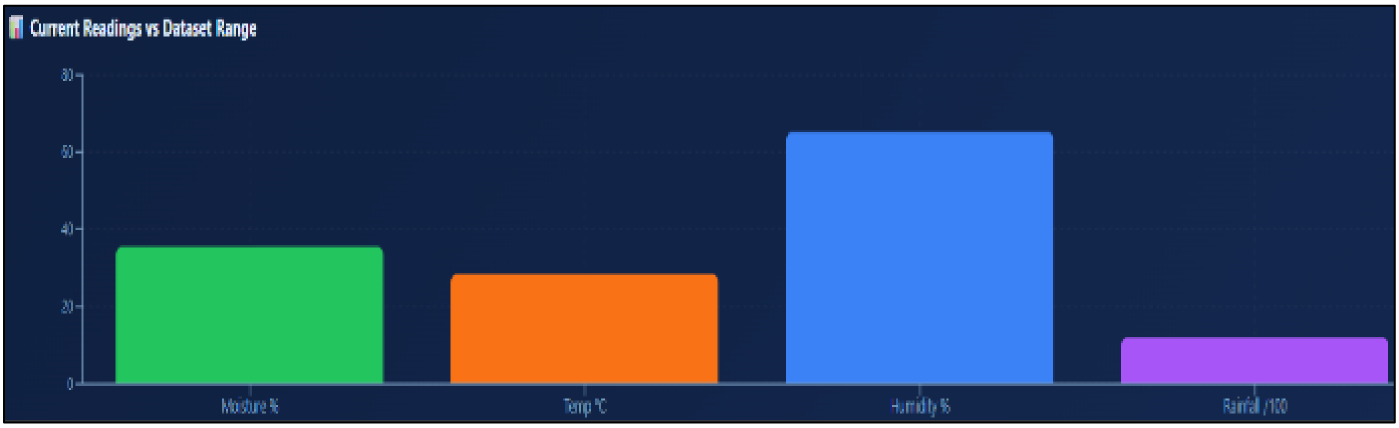

Fig 7 Current Readings Vs Dataset Range

Fig 8 RL Irrigation Decision

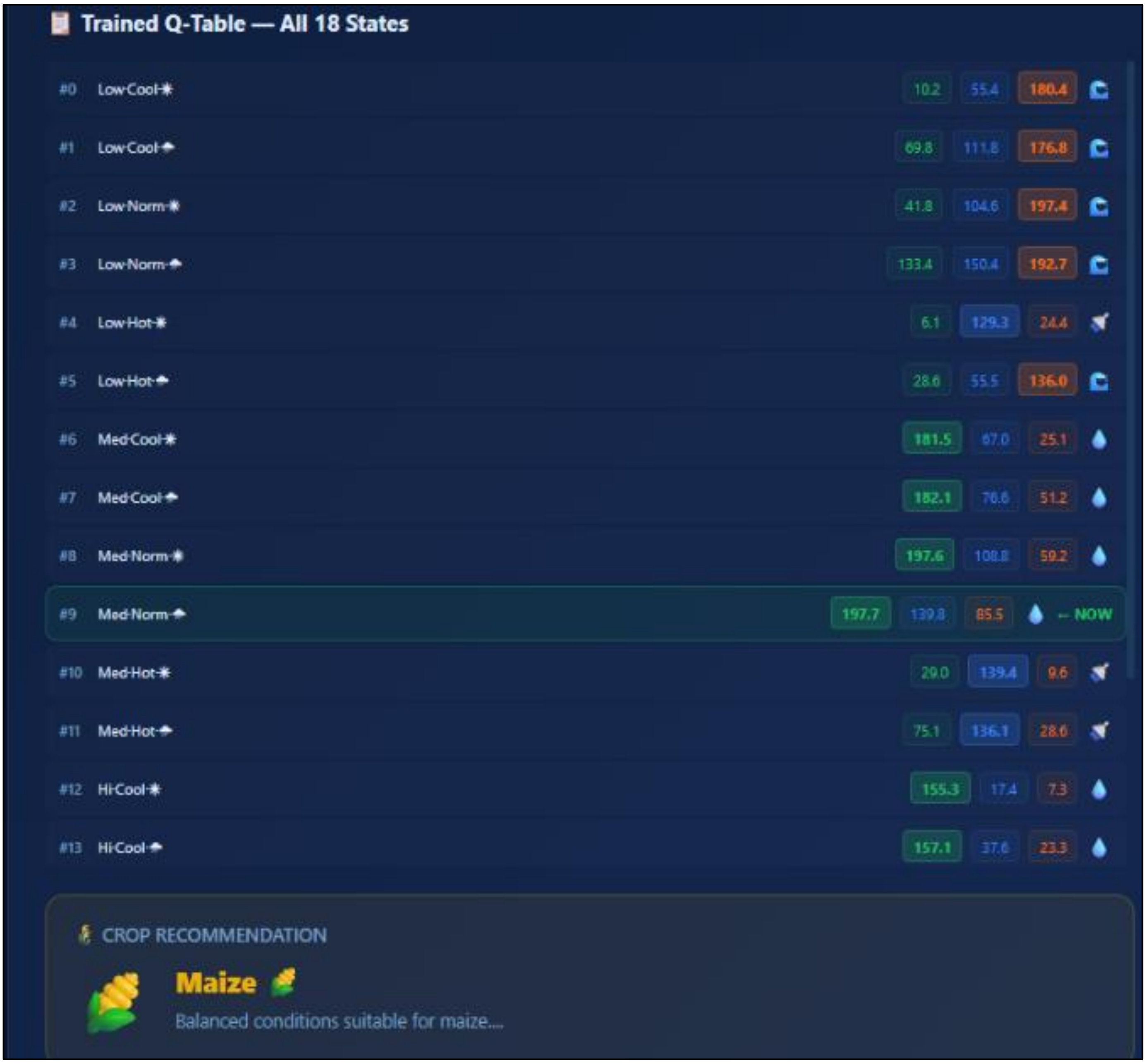

Fig 9 Q-Table & Crop Recommendation

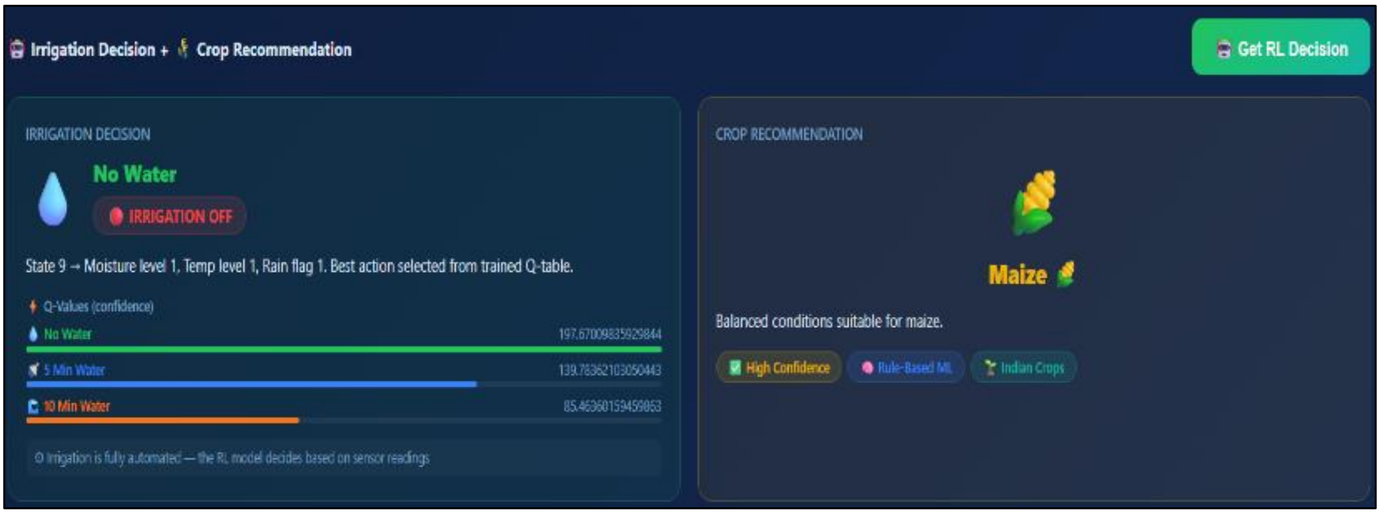

Fig 10 Irrigation Decision & Crop Recommendation

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr1285

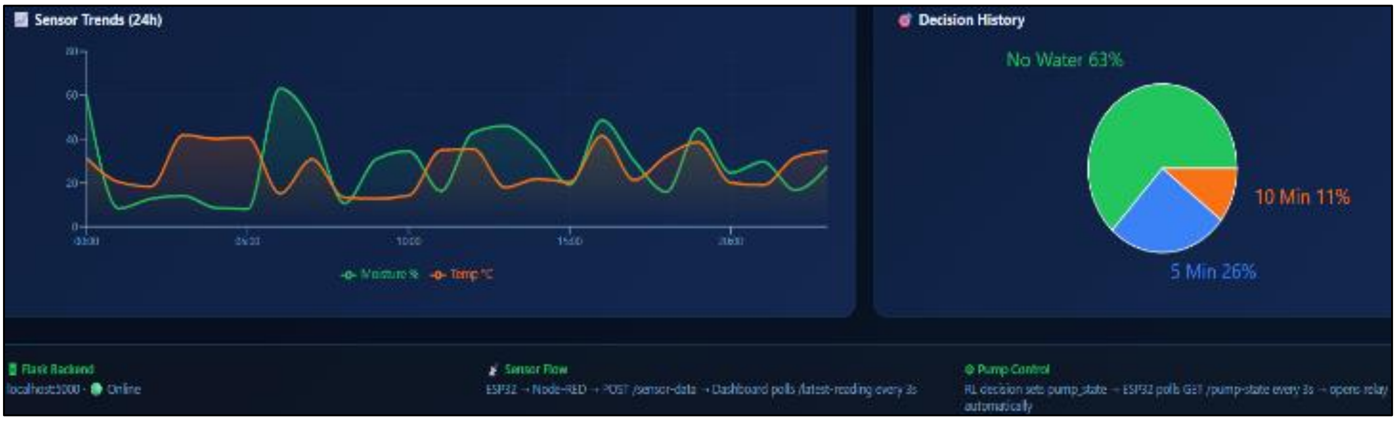

Fig 11 Sensor Trends

Fig 12 Historical Data

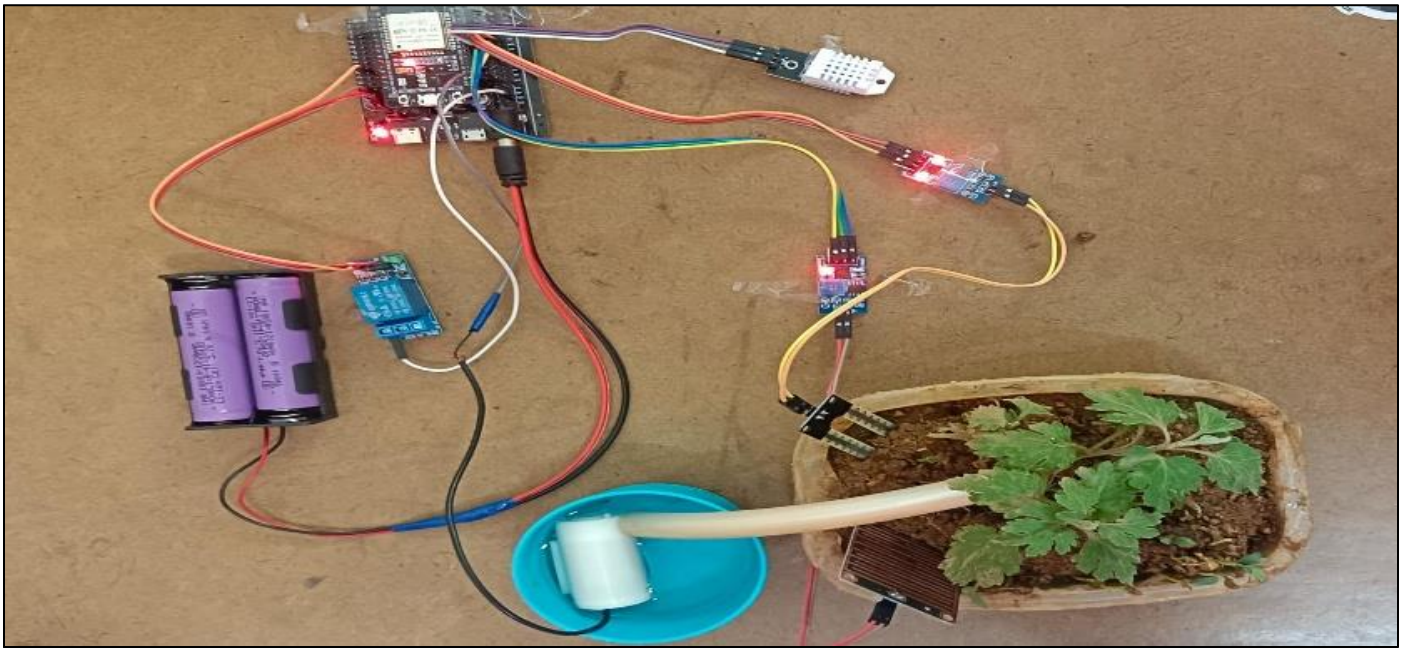

Fig 13 Sensor Setup

VII. CONCLUSION

The Smart Irrigation System is another demonstration of the use of IoT, ML, and RL in agriculture. While conventional systems require full involvement of humans, the Smart Irrigation System autonomously gathers data through sensors (soil moisture level, temperature, humidity, and rain) via ESP32 and sends them to the AI backend, which runs on Render. Based on its findings, the Q-learning algorithm determines whether irrigation is necessary and which action to perform - not irrigate at all, water for 5 minutes or water for 10 minutes. This will help optimize the performance of two objectives – minimizing water usage and avoiding stressing the crops in question. There is also a crop recommendation engine in the application that suggests crops that are most suitable under existing nutrients and environmental conditions. Sensor measurements, irrigation actions and the log of the pump operations are stored in the MongoDB Atlas database. All data collected by this system is accessible in the React frontend, which presents the results of data gathering including the sensor measurements, AI decision and justification, historical analysis, crop suggestions and manual irrigation functionality.

FUTURE SCOPE

However, there is a lot of room left for improving the performance and capabilities of the proposed smart irrigation system, including the implementation of predictive analytics and machine learning algorithms. In addition, the incorporation of other soil variables, such as the pH value, mineral content, and electrical conductivity, could make the system perform even better at crop recommendations and soil quality evaluation. The scalability of the smart irrigation system could be achieved through its use of LoRa or 5G wireless technology. Moreover, by incorporating the data obtained from weather forecasting APIs or satellite monitoring services, proactive measures could be taken in relation to the irrigation process.

REFERENCES

[1]. Risheh, Ali, Amirmohammad Jalili, and Ehsan Nazerfard. "Smart Irrigation IoT solution using transfer learning for neural networks." In *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 342-349. IEEE, 2020. Doi: https://doi.org/10.1109/ICCKE50421.2020.9303612

[2]. Kashyap, Pankaj Kumar, Sushil Kumar, Ankita Jaiswal, Mukesh Prasad, and Amir H. Gandomi. "Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network." *IEEE Sensors Journal* 21, no. 16 (2021): 17479-17491. https://doi.org/10.1109/JSEN.2021.3069266

[3]. Marios Angelopoulos, Constantinos, Gabriel Filios, Sotiris Nikoletseas, and Theofanis P. Raptis. "Keeping data at the edge of smart irrigation networks: A case study in strawberry greenhouses." *arXiv e-prints* (2021): arXiv-2109.

[4]. Jiménez, Andrés-F., Pedro-F. Cárdenas, and Fabián Jiménez. "Intelligent IoT-multiagent precision irrigation approach for improving water use efficiency in irrigation systems at farm and district scales." *Computers and Electronics in Agriculture* 192 (2022): 106635. https://doi.org/10.1016/j.compag.2021.106635.

[5]. Chen, Mengting, Yuanlai Cui, Xiaonan Wang, Hengwang Xie, Fangping Liu, Tongyuan Luo, Shizong Zheng, and Yufeng Luo. "A reinforcement learning approach to irrigation decision-making for rice using weather forecasts." *Agricultural Water Management* 250 (2021): 106838. https://doi.org/10.1016/j.agwat.2021.106838

[6]. Alibabaei, Khadijeh, Pedro D. Gaspar, Eduardo Assunção, Saeid Alirezazadeh, Tânia M. Lima, Vasco NGJ Soares, and João MLP Caldeira. "Comparison of on-policy deep reinforcement learning A2C with off-policy DQN in irrigation optimization: A case study at a site in Portugal." *Computers* 11, no. 7 (2022): 104. https://doi.org/10.3390/computers11070104

[7]. Rajasekhar, Dasari, Jinpeng Miao, Shivakant Mishra, Sanjeet Kumar Nayak, and Ramanarayan Yadav. "Intelligent irrigation technique for lora enabled fog assisted smart agriculture." In *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pp. 1-6. IEEE, 2023. https://doi.org/10.1109/WF-IoT58464.2023.10539579

[8]. Kwok, Jessica, and Yu Sun. "A smart IoT-based irrigation system with automated plant recognition using deep learning." In *Proceedings of the 10th international conference on computer modeling and simulation*, pp. 87-91. 2018. https://doi.org/10.1145/3177457.3177506

[9]. Moller, P., and D. Masseroni. "Evaluating performances of the first automatic system for paddy irrigation in Europe." (2018). http://tailieuso.tlu.edu.vn/handle/DHTL/4638

[10]. Kamienski, Carlos, Juha-Pekka Soininen, Markus Taumberger, Ramide Dantas, Attilio Toscano, Tullio Salmon Cinotti, Rodrigo Filev Maia, and André Torre Neto. "Smart water management platform: IoT-based precision irrigation for agriculture." *Sensors* 19, no. 2 (2019): 276. https://doi.org/10.3390/s19020276