
ClimatePulse: Local Voices, Smarter Forecasts for Farmers

Team Members:

Name	Role	College Name
Ritik Kumar	No-Code Developer	VIT Bhopal University
Kumari Tannu	Design and Research Coordinator	Techno India University
Gyan Chandra	Project Lead	IIITDM Kancheepuram

Hackathon: Vultr Cloud Innovate Hackathon

Date: 25-09-2024

ClimatePulse - Round 2 Project Submission PDF

---

1. Project Plan (1-2 pages)

Project Objectives

- **Goal**: To create an intuitive platform for farmers to share local weather observations, improving agricultural decision-making by providing more localized and accurate weather forecasts.
- **Problem Solved**: The platform addresses unpredictable weather, lack of localized data, limited community interaction, and resource management challenges in farming.
- **Impact**: Empower farmers with reliable weather data, improve crop management, and promote sustainable farming practices.

Timeline and Milestones

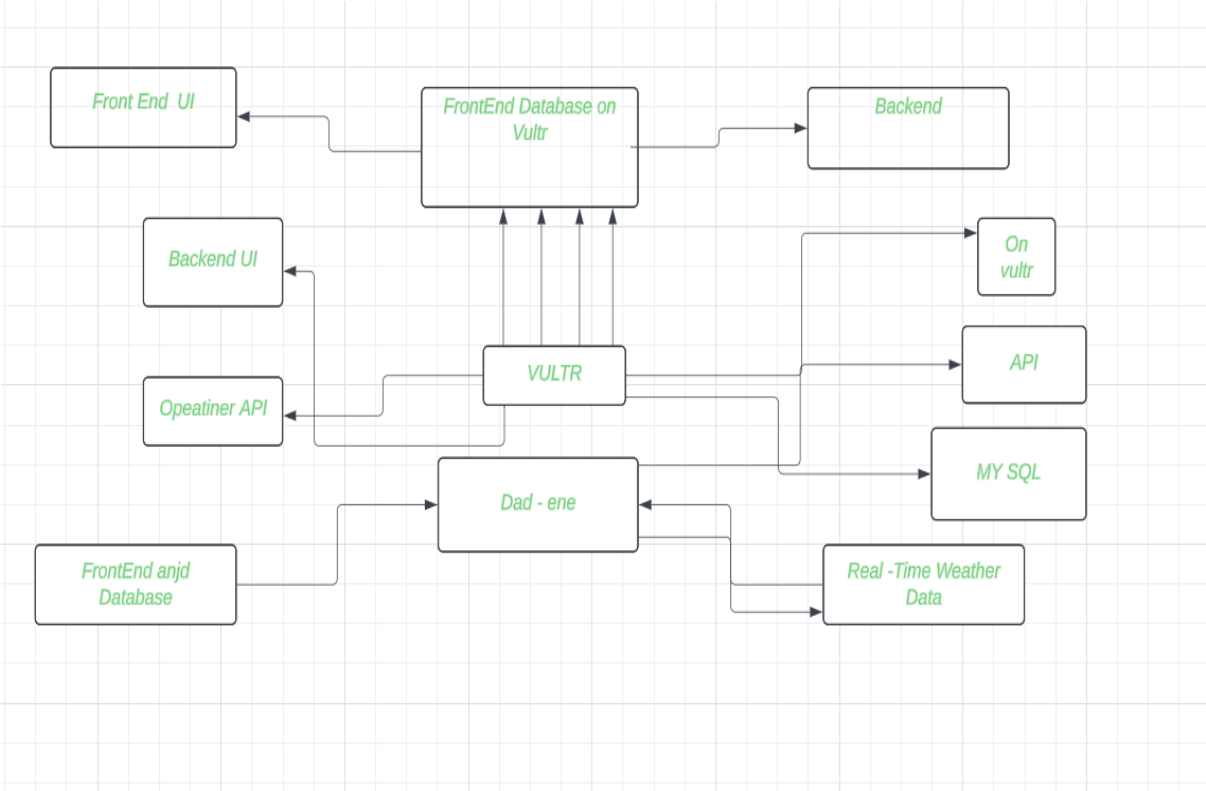
1. **Week 1-2**: Research and Requirements Gathering
    - Understand the problem in depth, finalize requirements, and plan the platform's core features.
2. **Week 3-4**: Development Phase (Using Bubble, Vultr, and MySQL)
    - Set up platform on Vultr, connect MySQL database, and integrate weather reporting features.
3. **Week 5**: Testing and Deployment
    - Test platform’s functionality, fix bugs, and deploy the first working version for initial feedback.
4. **Week 6**: Final Submission Preparation
    - Complete documentation, prepare codebase, and submit the project.

Deliverables

- Fully functioning ClimatePulse platform with weather reporting and data aggregation.
- Database integration with MySQL hosted on Vultr.
- User-facing dashboard displaying localized weather insights.
- Complete technical documentation.

---

2. Architecture Diagrams (1-2 pages)



High-Level Architecture Diagram

- **Frontend**: Bubble (No-code platform for UI and workflows).
- **Backend**: MySQL database hosted on Vultr.
- **API Integration**: OpenWeather API for weather data.
- **Data Flow**: Farmers input weather data, which is saved to the database. Data is processed for generating forecasts.

Component Diagram

- **Bubble Frontend**: Handles UI/UX, user authentication, and dashboard.
- **Vultr Hosting**: Cloud infrastructure to support the backend and database.
- **MySQL Database**: Stores weather data, user info, and aggregated data.
- **OpenWeather API**: Provides real-time weather data for forecasts.

Network Topology (if applicable)

- Data flows from the frontend (Bubble) to the backend (Vultr MySQL).
- API calls to external weather services (OpenWeather).

---

3. Technical Documentation (2-3 pages)

System Architecture

- **Bubble**: A no-code platform used for building the UI and handling user workflows.
- **MySQL Database**: Stores weather data, user information, and aggregated reports.
- **Vultr Cloud**: Cloud infrastructure that ensures the application's scalability and performance.
- **OpenWeather API**: Used for fetching weather data based on users' geographic location.

Key Components and Modules

- **User Authentication**: Login system connected to the MySQL database.
- **Weather Reporting**: Allows farmers to input weather data (e.g., temperature, humidity).
- **Dashboard**: Displays weather reports and aggregated data to users.
- **API Integration**: Retrieves current weather forecasts and reports from the OpenWeather API.

API Documentation

- **OpenWeather API**:
  - Endpoint: `https://api.openweathermap.org/data/2.5/weather?lat={latitude}&lon={longitude}&appid={api_key}`
  - Response: JSON data containing weather parameters (e.g., temperature, humidity, wind speed).

Setup and Usage Instructions

- **Deploy on Vultr**:
  - Set up a Vultr account and deploy the MySQL instance.
  - Set up your database to store weather data.
- **Configure Bubble**:
  - Build the frontend using Bubble's drag-and-drop interface.
  - Set up API connections (for weather data) and workflows (for user input and data storage).

---

4. Prototype/Codebase (1 page)

GitHub Repository Link

https://github.com/krtannu/ClimatePulse.git

Codebase Details:

- Source Code: All relevant files for the platform's development.
- Build Instructions: How to deploy and run the project (e.g., setting up MySQL on Vultr, Bubble workflows, and API integrations).
- Deployment Instructions: How to set up Vultr and integrate MySQL with Bubble for full functionality.

---

5. Conclusion & Future Work (Optional)

- **Challenges**:
  - Data validation and ensuring accurate weather reports.
  - Scalability concerns as more farmers join the platform.
  - Implementing advanced forecasting (yet to be done).
- **Future Work**:
  - **Collaborative Forecasting**: Develop algorithms for more accurate weather predictions using the collected data.
  - **Mobile App**: Extend the platform to a mobile version for better accessibility.