University College Dublin

COMP47250 Team Software Project
Project Plan - HAB Detection System
Team: Gradient Descent

Team Members:
Daniel Ilyin [18327256]
Sagar Satish Poojary [24205485]
Dharmik Arvind Vara [24215287]
Karthika Garikapati [24206355]
Kruthi Jangam [24206846]
Roshan Palem [24209558]

Contents

1 Project Objectives

This project aims to develop a web-based Harmful Algal Bloom (HAB) detection and prediction system based on the HABNet architecture proposed by Hill et al. (2020). HABs pose significant risks to marine ecosystems, aquaculture operations, and public health, with traditional detection methods relying on periodic manual sampling that leads to delayed response times.
Our system will implement a spatiotemporal "datacube" approach combined with deep neural networks to achieve high-accuracy HAB detection and prediction capabilities. The primary objectives include:
Core Technical Objectives:

- Develop an automated datacube generator for processing remote sensing data from MODIS-Aqua/Terra satellites and Sentinel-3 sensors
- Implement and train deep learning models (NASNet-Mobile backbone with LSTM) for HAB event classification
- Create a minimum viable reproduction achieving ¿90% detection accuracy and 80% prediction accuracy
- Build a real-time web-based dashboard for HAB monitoring and prediction

User-Centered Objectives:

- Provide marine biologists and environmental agencies with early warning capabilities (up to 8 days ahead)
- Enable aquaculture operators to implement timely mitigation measures
- Support public health officials in issuing timely advisories for recreational water use

Success will be measured through technical performance metrics (detection/prediction accuracy), user evaluation feedback, and system scalability demonstrations using cloud infrastructure.

2 Project Plan

2.1 Sprint Overview

Our development follows an agile methodology with 2-week sprints aligned with project milestones:

Sprint	Dates	Deliverables & Goals
Sprint 1	20/5 - 2/6	Team formation, environment setup, initial data source exploration
Sprint 2	3/6 - 16/6	MVP development, basic datacube pipeline, simple classifier

Sprint 3	17/6 - 30/6	Model training, interim presentation preparation, web interface prototype
Sprint 4	1/7 - 14/7	Advanced model implementation, cloud deployment, user testing framework
Sprint 5	15/7 - 28/7	System integration, performance optimization, comprehensive evaluation
Sprint 6	29/7 - 11/8	Final testing, documentation, presentation preparation
Sprint 7	12/8 - 19/8	Final report completion, system refinements

2.2 Key Dates

- **9/6/2025:** Project Plan submission
- **23/6/2025:** Interim Presentation & MVP Demo
- **4/8/2025:** Final Presentation & Complete System Demo
- **19/8/2025:** Final Report submission

3 Roles

- **Project Manager (Kruthi):** Tracks milestones, maintains sprint board, leads team syncs.
- **ML Lead (Roshan):** Leads model selection, training, and tuning.
- **Data Engineer (Karthika):** Manages dataset acquisition, cleaning, and ETL.
- **Frontend Developer (Dharmik):** Builds the Streamlit dashboard and integrates backend.
- **DevOps & Cloud Lead (Daniel):** Sets up and maintains cloud infrastructure and deployment.
- **UX & Testing Lead (Sagar):** Handles UI testing, SHAP analysis, and final user evaluations.

4 Architecture

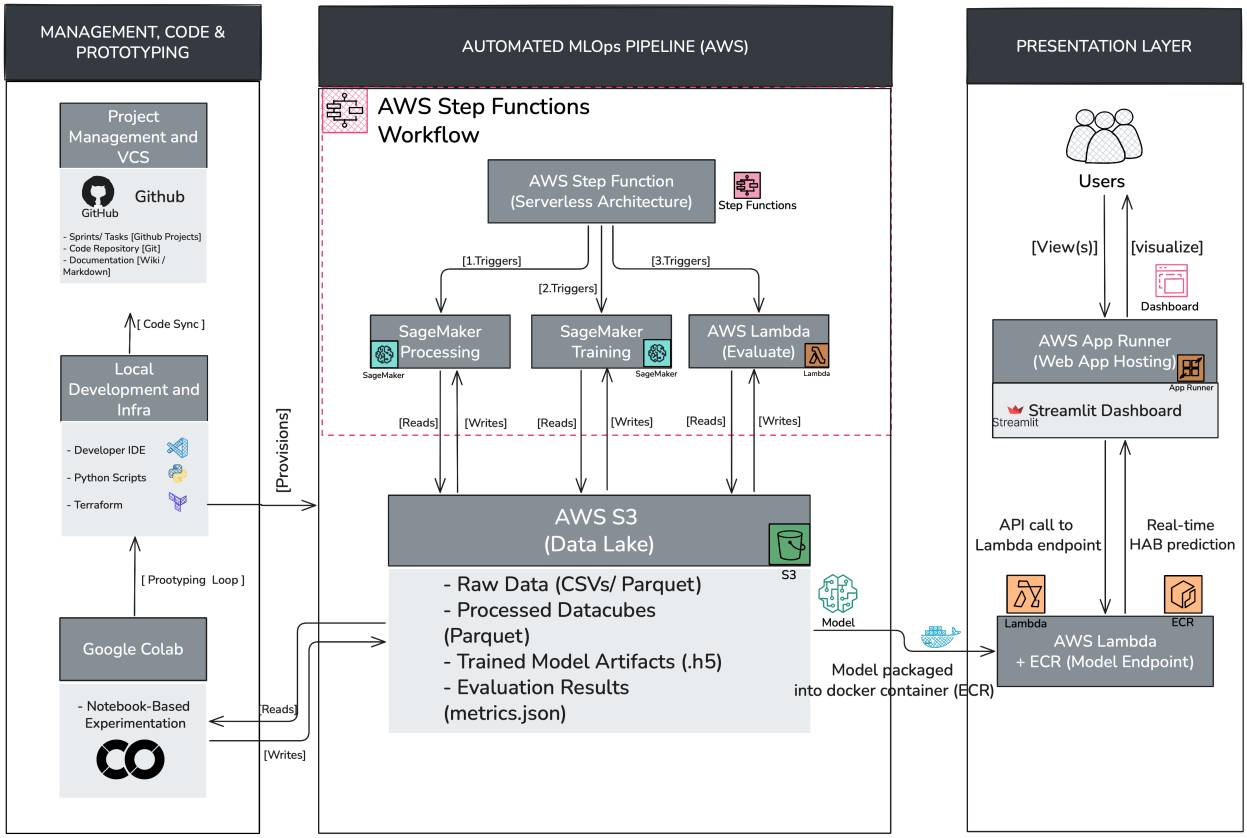

Figure 1: System architecture for HAB detection and prediction. The pipeline integrates local development, version control, and notebook experimentation with an automated AWS-based MLOps workflow using Step Functions, SageMaker, and Lambda. Outputs are stored in S3 and served through a Streamlit dashboard hosted on AWS App Runner for real-time HAB prediction.

5 Data Plan

6 GitHub Repository

Our GitHub repository can be found at https://github.com/danieli1245/Harmful-Algal-Bloom-Detection-System

7 Team Management

The team follows a sprint-based approach using GitHub Projects to track progress and assign tasks. Weekly sync-up meetings are held to review milestones, address blockers, and redistribute work as needed.

We maintain a shared GitHub repository where all members regularly commit updates to ensure version control and collaboration. Communication is done through a dedicated group chat and short catch-up calls when necessary.
Everyone is accountable for their role, but we also coordinate cross-functionally to ensure smooth integration of all components.