Software Engineering Lab
Assignment 1

DeepBlue EcoGuard: Underwater Waste Detection and Water Quality Assessment

DeepBlue EcoGuard tackles the rising issue of underwater waste with three key solutions: a YoloV8 Algorithm for precise waste detection, a rule-based classifier for assessing aquatic habitat, and a Machine Learning model to classify water quality. The YoloV8 Algorithm, trained on 5000 images, detects waste, while the rule-based classifier uses guidelines from the US EPA and WHO for habitat assessment. The Machine Learning model, trained on a dataset of over 6 million rows, reliably classifies water as fit or unfit. This project offers a comprehensive approach to safeguard our oceans and marine life.

Functionalities:

- Underwater Waste Detection: The YoloV8 Algorithm accurately detects and identifies underwater waste, providing real-time insights into the presence and distribution of pollutants.

- Aquatic Habitat Assessment: The rule-based classifier assesses the habitat of aquatic life based on chemical properties, using guidelines from authoritative sources like the US EPA and WHO. This functionality helps in understanding the impact of pollutants on marine ecosystems.

- Water Quality Classification: The Machine Learning model evaluates water quality, classifying it as either fit or unfit for use. This assessment is crucial for monitoring and managing water resources and ensuring the well-being of both marine life and human communities.

Applications:

- Environmental Monitoring: DeepBlue EcoGuard can be deployed for continuous monitoring of underwater environments, providing valuable data for environmental agencies, researchers, and policymakers to track and address pollution issues.

- Marine Conservation: By accurately detecting and assessing underwater waste, the project contributes to marine conservation efforts. Understanding the impact on aquatic habitats helps in designing targeted conservation strategies.

- Water Resource Management: The water quality classification functionality is vital for managing and preserving water resources. It can be employed in areas where access to clean water is a concern, guiding efforts to maintain or improve water quality.

- Early Warning System: The real-time capabilities of the YoloV8 Algorithm make DeepBlue EcoGuard an effective early warning system for pollution events. This can be crucial for rapid response and mitigation measures.
- Research and Education: The project serves as a valuable tool for researchers and educators studying marine ecosystems, pollution dynamics, and water quality. It can be integrated into educational programs to raise awareness about environmental issues.

PAJAMA PADHAI

Overall, DeepBlue EcoGuard's functionalities and applications make it a versatile and powerful tool in the realm of underwater waste detection, habitat assessment, and water quality management.

Resources Requirements:

1. Hardware Requirements:
   - High-performance servers or cloud computing resources for real-time processing and analysis.
   - GPUs (Graphics Processing Units) for efficient training and execution of deep learning models, particularly for YoloV8 and Dark Channel Prior Algorithm.

2. Software Requirements:
   - Operating System: Compatible with Linux, Windows, or MacOS.
   - Python Environment: Python 3.x for coding and implementation.
   - Development Tools: Edrawmax, StarUML for designing system architecture, flowcharts, and documentation.

3. Functional Requirements:
   - Data Collection: Capability to collect diverse underwater images and chemical properties datasets.
   - Algorithm Development: Implementation of YoloV8, Dark Channel Prior Algorithm, and Xgboost-Classifier for precise waste detection, habitat assessment, and water quality classification.
   - Integration: Ability to integrate the developed algorithms into a cohesive system for seamless functionality.
   - Real-Time Monitoring: Implement a real-time monitoring system for continuous analysis and display of results.

4. Non-Functional Requirements:
   - Performance: The system should exhibit high performance, providing real-time results for waste detection and analysis.
   - Scalability: The architecture should be scalable to handle an increasing amount of data and users.
   - Reliability: Ensure the system's reliability through robust testing, minimizing errors and downtime.
   - Usability: Design a user-friendly interface for easy interaction and accessibility.
   - Security: Implement security measures to protect sensitive data and ensure the integrity of the system.

5. Frontend and Backend Requirements:
   - Frontend: Streamlit (version 1.11.1) for developing a user-friendly interface.
   - Backend: Python-based backend using libraries such as matplotlib, numpy, opencv_python, pandas, pycaret, scipy, seaborn, torch, ultralytics, and xgboost.

6. Time Requirements:
   - Development Time: Depending on the complexity, the development phase may take several months.
   - Training Time: Training deep learning models (YoloV8, Dark Channel Prior, Xgboost) may require significant computation time.
   - Testing and Validation: Allocate time for thorough testing and validation to ensure the robustness and reliability of the developed algorithms and the integrated system.

PAJAMA PADHAI

Modules of Work:

1. Data Collection:
   - Gather diverse underwater images for training the YoloV8 Algorithm.

- Collect chemical properties data for training the rule-based classifier and Machine Learning model.

2. YoloV8 Algorithm Development:
    - Implement and train the YoloV8 Algorithm on the collected image dataset for precise underwater waste detection.

3. Rule-Based Classifier Design:
    - Develop a rule-based classifier using guidelines from the US EPA and WHO to assess aquatic habitat based on chemical properties.

4. Machine Learning Model Training:
    - Train the Machine Learning model on a dataset of over 6 million rows to classify water quality as fit or unfit for use.

5. Integration of Functionalities:
    - Integrate the three functionalities into the DeepBlue EcoGuard system, ensuring seamless communication and cooperation among modules.

6. Real-Time Monitoring System:
    - Implement a real-time monitoring system to continuously analyze and display results, providing instant feedback on underwater waste, habitat assessment, and water quality.

7. Validation and Testing:
    - Validate the accuracy and reliability of each module through rigorous testing, ensuring the system's effectiveness in different underwater environments.

8. Optimization and Fine-Tuning:
    - Optimize algorithms and models based on feedback from testing, improving the efficiency and accuracy of the overall system.

9. User Interface Development:
    - Create a user-friendly interface for easy interaction with the DeepBlue EcoGuard system, facilitating access to real-time data and insights.

10. Deployment and Maintenance:
    - Deploy the system in relevant underwater environments and establish a maintenance plan for continuous functionality and performance.

Suitable process model:

PAJAMA PADHAI

For the DeepBlue EcoGuard project, the most suitable process model is the **Incremental Model**.

Reasoning:

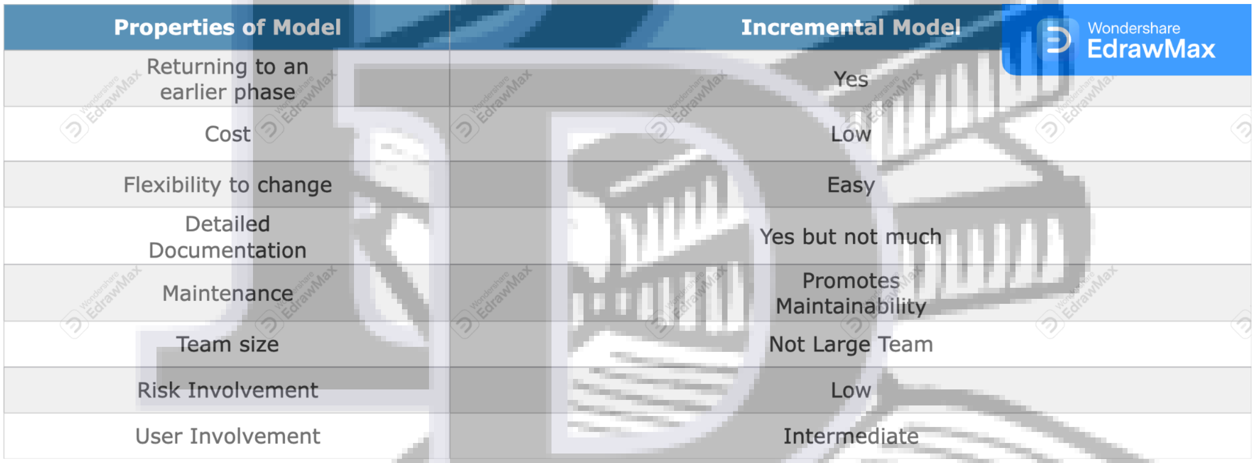
- Iterative Development: The Incremental Model allows for iterative development, which aligns well with the project's complex nature. The development of underwater waste detection, habitat assessment, and water quality classification modules can be tackled incrementally.

- Progressive Building of Functionality: Each increment in the Incremental Model adds new functionality to the system. This aligns with the DeepBlue EcoGuard project, where distinct functionalities, such as waste detection, habitat assessment, and water quality classification, can be developed and integrated incrementally.

- Early Delivery of Partial System: The Incremental Model facilitates the early delivery of a partial system with each increment. This is beneficial for the DeepBlue EcoGuard project, as it allows stakeholders to start using and benefiting from certain functionalities while others are still under development.

- Feedback Mechanism: The model incorporates a feedback mechanism after each increment, allowing for continuous improvement based on user feedback and changing requirements. This is crucial for a project like DeepBlue EcoGuard, where real-world conditions and user interactions may necessitate adjustments and enhancements.

- Risk Mitigation: Incremental development helps in mitigating risks by allowing for the identification and resolution of issues early in the development process. Given the complexity of the project and the integration of various algorithms, this iterative approach reduces the likelihood of major setbacks.



Properties of Model	Incremental Model
Returning to an earlier phase	Yes
Cost	Low
Flexibility to change	Easy
Detailed Documentation	Yes but not much
Maintenance	Promotes Maintainability
Team size	Not Large Team
Risk Involvement	Low
User Involvement	Intermediate

Wondershare EdrawMax

Work Break-Down Structure:

1. Project Initiation

1.1 Define Project Scope:

1.1.1 Identify Key Project Objectives: Clearly articulate the specific goals and objectives the DeepBlue EcoGuard project aims to achieve.
1.1.2 Establish Project Boundaries: Clearly delineate the limits and constraints of the project scope, defining what is included and excluded.

PAJAMA PADHAI

1.2 Conduct Stakeholder Analysis:

1.2.1 Identify Project Stakeholders: Enumerate and identify all individuals or groups that may be affected by or affect the project.
1.2.2 Assess Stakeholder Influence and Interests: Evaluate the level of influence and the interests of each stakeholder to understand their impact on the project.

1.3 Establish Project Objectives and Success Criteria:

1.3.1 Define Clear Project Goals: Explicitly state the desired outcomes and achievements of the project.

1.3.2 Develop Measurable Success Criteria: Create specific and measurable criteria to assess whether the project has met its objectives.

1.4 Develop Project Charter:

1.4.1 Outline Project Scope and Objectives: Summarize the project's scope, objectives, stakeholders, risks, and deliverables.

1.4.2 Identify Project Team Members: List and define the roles and responsibilities of each team member.

1.4.3 Obtain Stakeholder Sign-off: Secure formal approval from key stakeholders, indicating their acceptance of the project charter.

2. Data Collection

2.1 Gather Underwater Image Dataset:

2.1.1 Identify Suitable Image Sources: Determine appropriate sources for diverse underwater images, considering authenticity and relevance.

2.1.2 Obtain Necessary Permissions and Licenses: Ensure compliance with legal and ethical considerations by obtaining permissions and licenses for image usage.

2.2 Collect Chemical Properties Dataset:

2.2.1 Compile Chemical Properties Guidelines (US EPA, WHO): Assemble guidelines from authoritative sources such as US EPA and WHO to guide the collection of chemical properties data.

2.2.2 Extract Relevant Data from Available Sources: Gather chemical properties data from reliable sources, aligning with the established guidelines.

2.3 Verify Data Quality and Relevance:

2.3.1 Conduct Data Quality Checks: Assess the accuracy and reliability of collected data through systematic quality checks.

2.3.2 Cross-Verify Data Relevance for Algorithm Training: Ensure that the collected data is relevant and suitable for training the algorithms.

3. Algorithm Development

3.1 YoloV8 Algorithm Implementation:

3.1.1 Set Up YoloV8 Development Environment: Prepare the development environment with the necessary tools and libraries for YoloV8 implementation.

3.1.2 Train YoloV8 on Sample Data: Utilize sample data to train the YoloV8 algorithm for accurate underwater waste detection.

3.2 Dark Channel Prior Algorithm Integration:

3.2.1 Incorporate Dark Channel Prior into Image Processing Pipeline: Integrate the Dark Channel Prior algorithm into the image processing pipeline to enhance image quality.

3.2.2 Test Algorithm with Sample Images: Evaluate the performance of the Dark Channel Prior algorithm using sample images.

3.3 Xgboost-Classifier Development:

3.3.1 Preprocess Data for Xgboost: Prepare the data for Xgboost classification through cleaning, normalization, and feature engineering.

PAJAMA PADHAI

3.3.2 Train Xgboost-Classifier: Train the Xgboost-Classifier using preprocessed data for water quality classification.

3.4 Algorithm Unit Testing:

3.4.1 Verify YoloV8 Detection Accuracy: Assess the accuracy of the YoloV8 algorithm in detecting underwater waste.

3.4.2 Validate Dark Channel Prior Image Enhancement: Confirm the effectiveness of the Dark Channel Prior algorithm in image enhancement.

3.4.3 Evaluate Xgboost-Classifier Performance: Gauge the performance of the Xgboost-Classifier in water quality classification.

4. Integration

4.1 System Architecture Design:

4.1.1 Define Overall System Architecture: Outline the structure and components of the entire system, including how modules will interact.

4.1.2 Identify Communication Protocols: Specify the communication protocols that will facilitate seamless interaction between system components.

4.2 Algorithm Integration:

4.2.1 Combine YoloV8, Dark Channel Prior, and Xgboost-Classifier: Integrate individual algorithms to create a unified system for comprehensive underwater assessment.

4.2.2 Ensure Compatibility and Consistency: Verify that integrated algorithms work together cohesively, maintaining consistency and compatibility.

4.3 Develop Data Communication Channels:

4.3.1 Establish Efficient Data Exchange Protocols: Create protocols to ensure efficient and secure exchange of data between integrated modules.

4.3.2 Implement Secure Data Transmission: Incorporate security measures to protect data during transmission.

4.4 Integration Testing:

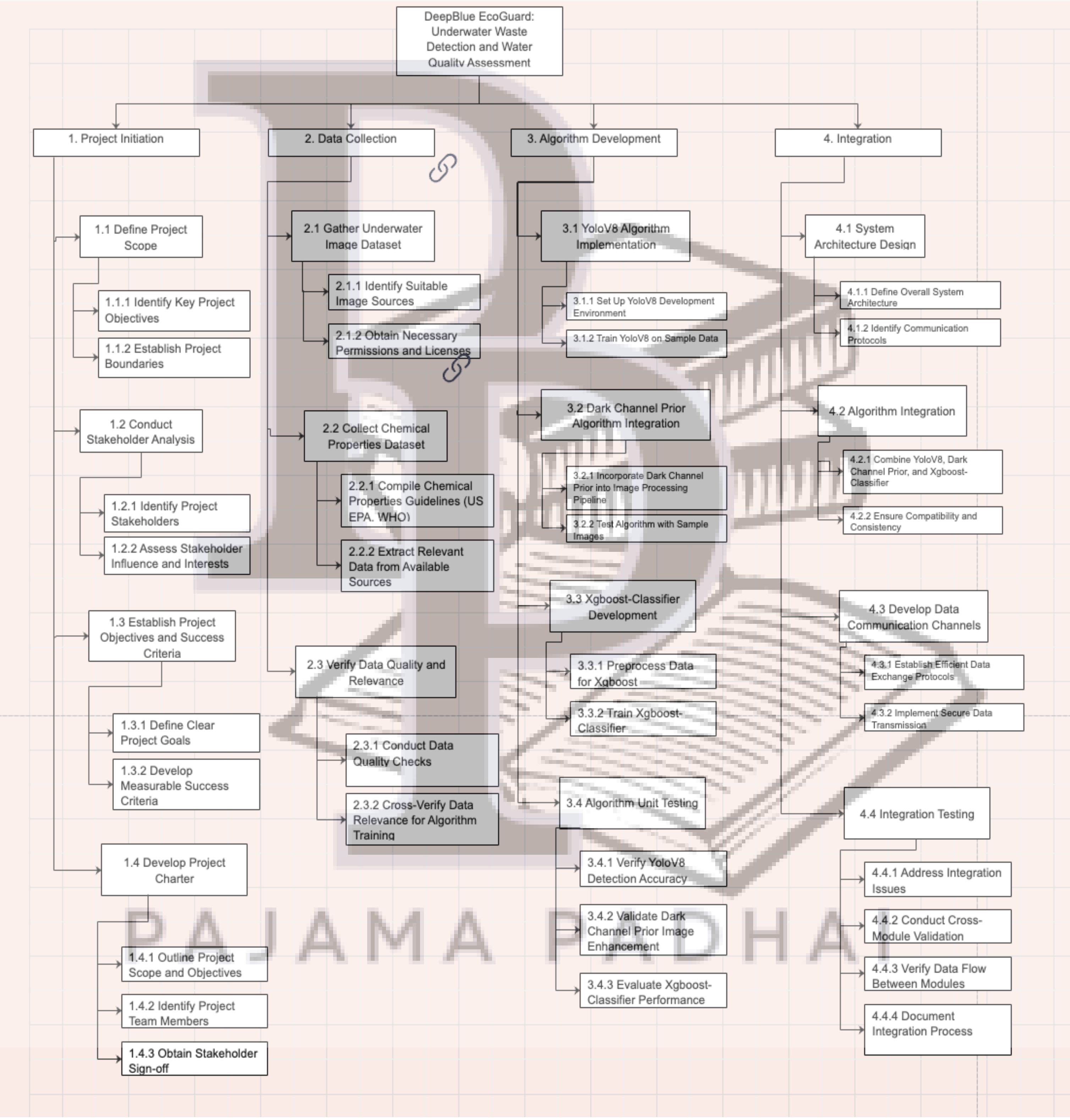
4.4.1 Address Integration Issues: Identify and resolve any issues that arise during the integration process.

4.4.2 Conduct Cross-Module Validation: Verify that each module works correctly when integrated with others.

4.4.3 Verify Data Flow Between Modules: Ensure smooth data flow between integrated modules.

4.4.4 Document Integration Process: Record the steps taken during the integration process for reference and future maintenance.

PAJAMA PADHAI



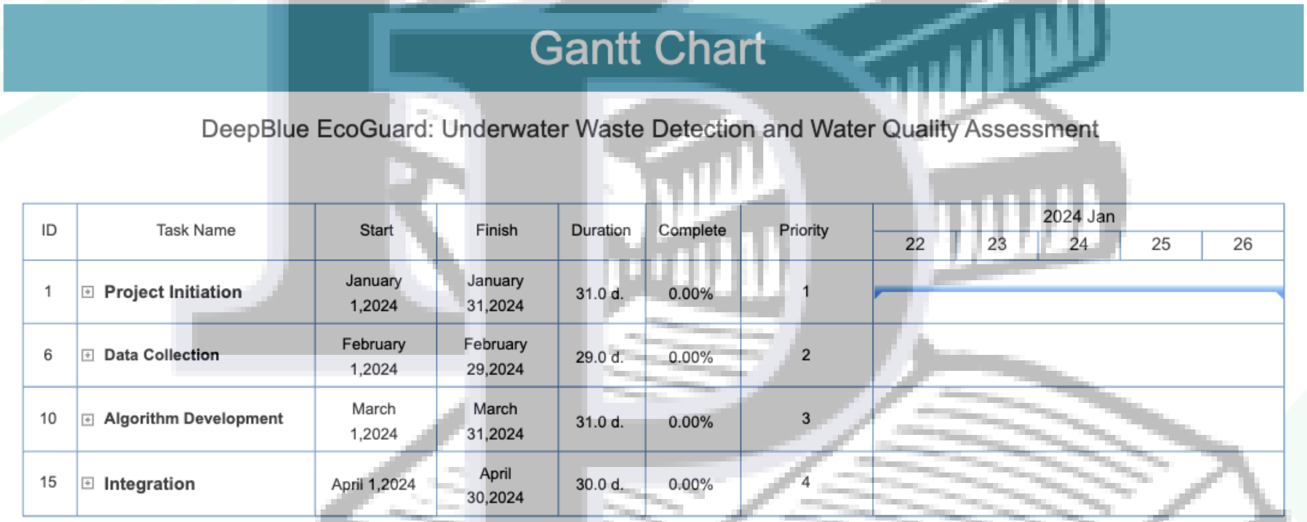
DeepBlue EcoGuard: Underwater Waste Detection and Water Quality Assessment

- 1. Project Initiation
  - 1.1 Define Project Scope
    - 1.1.1 Identify Key Project Objectives
    - 1.1.2 Establish Project Boundaries
  - 1.2 Conduct Stakeholder Analysis
    - 1.2.1 Identify Project Stakeholders
    - 1.2.2 Assess Stakeholder Influence and Interests
  - 1.3 Establish Project Objectives and Success Criteria
    - 1.3.1 Define Clear Project Goals
    - 1.3.2 Develop Measurable Success Criteria
  - 1.4 Develop Project Charter
    - 1.4.1 Outline Project Scope and Objectives
    - 1.4.2 Identify Project Team Members
    - 1.4.3 Obtain Stakeholder Sign-off
- 2. Data Collection
  - 2.1 Gather Underwater Image Dataset
    - 2.1.1 Identify Suitable Image Sources
    - 2.1.2 Obtain Necessary Permissions and Licenses
  - 2.2 Collect Chemical Properties Dataset
    - 2.2.1 Compile Chemical Properties Guidelines (US EPA, WHO)
    - 2.2.2 Extract Relevant Data from Available Sources
  - 2.3 Verify Data Quality and Relevance
    - 2.3.1 Conduct Data Quality Checks
    - 2.3.2 Cross-Verify Data Relevance for Algorithm Training
- 3. Algorithm Development
  - 3.1 YoloV8 Algorithm Implementation
    - 3.1.1 Set Up YoloV8 Development Environment
    - 3.1.2 Train YoloV8 on Sample Data
  - 3.2 Dark Channel Prior Algorithm Integration
    - 3.2.1 Incorporate Dark Channel Prior into Image Processing Pipeline
    - 3.2.2 Test Algorithm with Sample Images
  - 3.3 Xgboost-Classifier Development
    - 3.3.1 Preprocess Data for Xgboost
    - 3.3.2 Train Xgboost-Classifier
  - 3.4 Algorithm Unit Testing
    - 3.4.1 Verify YoloV8 Detection Accuracy
    - 3.4.2 Validate Dark Channel Prior Image Enhancement
    - 3.4.3 Evaluate Xgboost-Classifier Performance
- 4. Integration
  - 4.1 System Architecture Design
    - 4.1.1 Define Overall System Architecture
    - 4.1.2 Identify Communication Protocols
  - 4.2 Algorithm Integration
    - 4.2.1 Combine YoloV8, Dark Channel Prior, and Xgboost-Classifier
    - 4.2.2 Ensure Compatibility and Consistency
  - 4.3 Develop Data Communication Channels
    - 4.3.1 Establish Efficient Data Exchange Protocols
    - 4.3.2 Implement Secure Data Transmission
  - 4.4 Integration Testing
    - 4.4.1 Address Integration Issues
    - 4.4.2 Conduct Cross-Module Validation
    - 4.4.3 Verify Data Flow Between Modules
    - 4.4.4 Document Integration Process

PAJAMA PADHAI

Gantt Chart:

The Gantt chart provides a visual representation of the DeepBlue EcoGuard project timeline, outlining key tasks and their corresponding start and end dates. The project is organized into distinct phases, such as Project Initiation, Data Collection, Algorithm Development, and Integration. Each phase consists of specific tasks, subtasks, and milestones, allowing for a structured and comprehensive overview of the project's progression. The timeline highlights the sequential nature of tasks and dependencies between different project elements. This Gantt chart serves as a valuable tool for project managers and team members to track progress, manage resources, and ensure timely completion of each project component. Regular updates to the Gantt chart will facilitate effective project monitoring, enabling the team to adapt to changes and maintain project timelines.



Gantt Chart

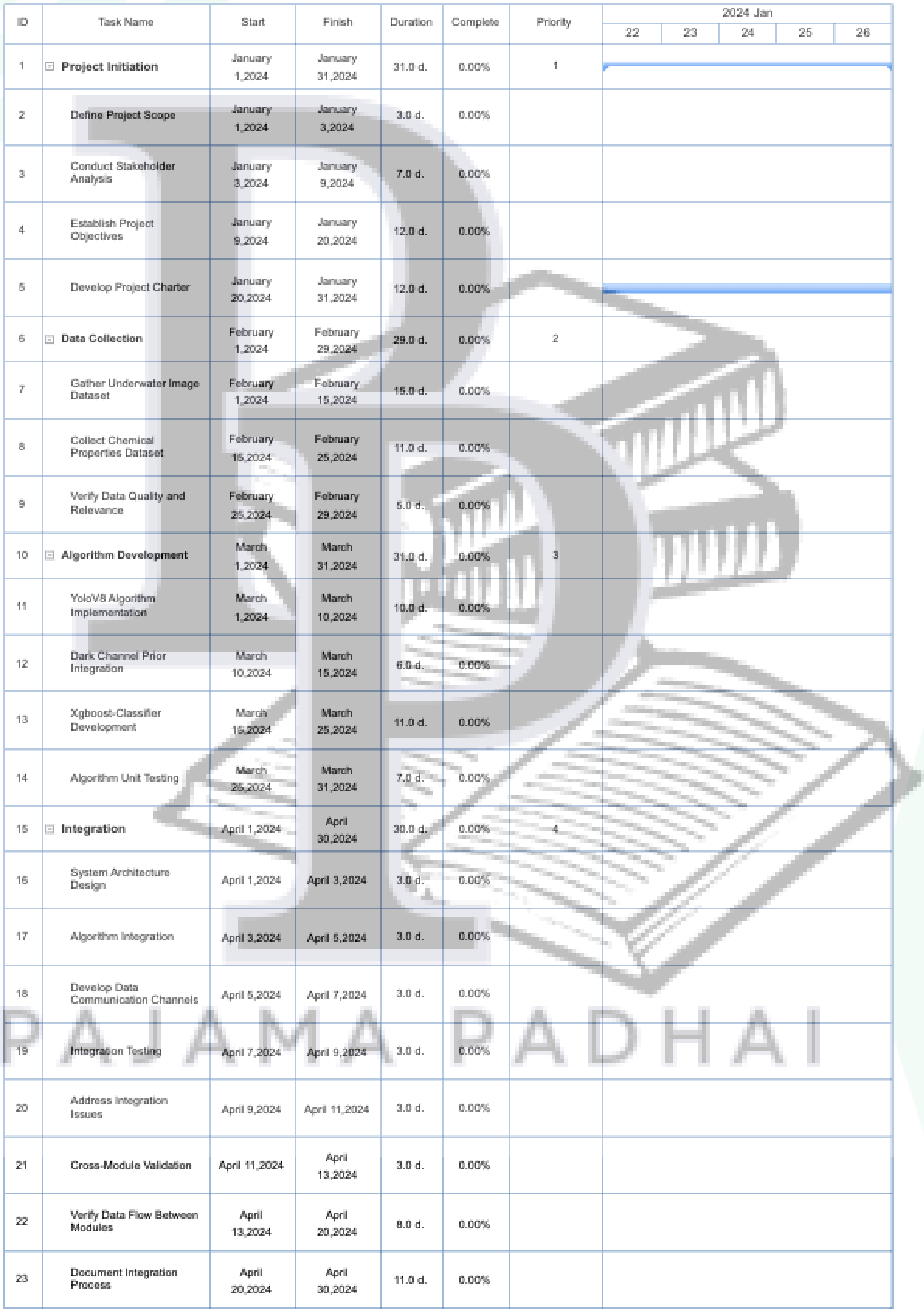
DeepBlue EcoGuard: Underwater Waste Detection and Water Quality Assessment

ID	Task Name	Start	Finish	Duration	Complete	Priority	2024 Jan				
							22	23	24	25	26
1	⊞ **Project Initiation**	January 1,2024	January 31,2024	31.0 d.	0.00%	1					
6	⊞ **Data Collection**	February 1,2024	February 29,2024	29.0 d.	0.00%	2					
10	⊞ **Algorithm Development**	March 1,2024	March 31,2024	31.0 d.	0.00%	3					
15	⊞ **Integration**	April 1,2024	April 30,2024	30.0 d.	0.00%	4					

PAJAMA PADHAI

Gantt Chart

DeepBlue EcoGuard: Underwater Waste Detection and Water Quality Assessment



ID	Task Name	Start	Finish	Duration	Complete	Priority	2024 Jan				
							22	23	24	25	26
1	⊟ **Project Initiation**	January 1,2024	January 31,2024	31.0 d.	0.00%	1					
2	Define Project Scope	January 1,2024	January 3,2024	3.0 d.	0.00%						
3	Conduct Stakeholder Analysis	January 3,2024	January 9,2024	7.0 d.	0.00%						
4	Establish Project Objectives	January 9,2024	January 20,2024	12.0 d.	0.00%						
5	Develop Project Charter	January 20,2024	January 31,2024	12.0 d.	0.00%						
6	⊟ **Data Collection**	February 1,2024	February 29,2024	29.0 d.	0.00%	2					
7	Gather Underwater Image Dataset	February 1,2024	February 15,2024	15.0 d.	0.00%						
8	Collect Chemical Properties Dataset	February 15,2024	February 25,2024	11.0 d.	0.00%						
9	Verify Data Quality and Relevance	February 25,2024	February 29,2024	5.0 d.	0.00%						
10	⊟ **Algorithm Development**	March 1,2024	March 31,2024	31.0 d.	0.00%	3					
11	YoloV8 Algorithm Implementation	March 1,2024	March 10,2024	10.0 d.	0.00%						
12	Dark Channel Prior Integration	March 10,2024	March 15,2024	6.0 d.	0.00%						
13	Xgboost-Classifier Development	March 15,2024	March 25,2024	11.0 d.	0.00%						
14	Algorithm Unit Testing	March 25,2024	March 31,2024	7.0 d.	0.00%						
15	⊟ **Integration**	April 1,2024	April 30,2024	30.0 d.	0.00%	4					
16	System Architecture Design	April 1,2024	April 3,2024	3.0 d.	0.00%						
17	Algorithm Integration	April 3,2024	April 5,2024	3.0 d.	0.00%						
18	Develop Data Communication Channels	April 5,2024	April 7,2024	3.0 d.	0.00%						
19	Integration Testing	April 7,2024	April 9,2024	3.0 d.	0.00%						
20	Address Integration Issues	April 9,2024	April 11,2024	3.0 d.	0.00%						
21	Cross-Module Validation	April 11,2024	April 13,2024	3.0 d.	0.00%						
22	Verify Data Flow Between Modules	April 13,2024	April 20,2024	8.0 d.	0.00%						
23	Document Integration Process	April 20,2024	April 30,2024	11.0 d.	0.00%						

PAJAMA PADHAI

Conclusion:

The DeepBlue EcoGuard project represents a significant stride towards addressing the escalating issue of underwater waste and enhancing water quality assessment. Through the meticulous execution of tasks outlined in the comprehensive Work Breakdown Structure and the strategic scheduling captured in the Gantt chart, the project is poised to deliver a robust and integrated solution.

The project's focus on algorithm development, including the implementation of YoloV8, integration of the Dark Channel Prior algorithm, and the creation of the Xgboost-Classifier, demonstrates a commitment to employing cutting-edge technologies for precise waste detection and water quality classification. The iterative development approach, reflected in the Incremental Model, ensures adaptability to evolving requirements and facilitates early deliveries of valuable functionalities.

The Gantt chart serves as a vital tool for project management, offering a visual roadmap for task scheduling, dependencies, and progress tracking. It provides a dynamic framework for team coordination and resource allocation, enabling effective decision-making throughout the project lifecycle.

As the project advances through its phases, the emphasis on data quality, algorithm validation, and integration testing will contribute to the creation of a reliable and accurate system. The inclusion of user-friendly interfaces and real-time monitoring mechanisms enhances the practicality and usability of the DeepBlue EcoGuard solution.

In conclusion, the DeepBlue EcoGuard project aligns with the imperative to safeguard our underwater ecosystems. By integrating advanced algorithms, robust testing methodologies, and an iterative development approach, the project aspires to make a positive impact on environmental conservation. The success of this endeavor will not only mark a technological achievement but also signify a commitment to the responsible stewardship of our aquatic environments.

PAJAMA PADHAI