

## Project Initialization and Planning Phase

Date	15 March 2024
Team ID	739798
Project Title	Virtual Eye – Lifeguard for Active Swimming Drowning Detection
Maximum Marks	3 Marks

### Project Proposal (Proposed Solution)

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	The objective of the "Virtual Eye" project is to provide real-time underwater safety detection by leveraging a deep learning model (YOLO) to analyze and detect potential hazards in underwater environments. The system allows users to upload images and videos for automated safety assessments, enhancing monitoring and response to underwater threats.
Scope	Analyze uploaded underwater imagery to detect swimming and drowning risks.
Problem Statement	
Description	The model is a real-time object detection in underwater environments. It uses Flask to host a web application that allows users to upload images or videos. These uploads are processed using the YOLO model, which scans for safety hazards. After processing, the system provides immediate feedback, including hazard alerts and detected object labels.
Impact	The project improves underwater safety by providing immediate analysis of visual data, helping to detect potential hazards in real-time. By automating the detection of underwater threats, it reduces the response time and enhances the overall safety of swimmers.
Proposed Solution	

Approach	Gather a diverse dataset of swimming images and videos. Train the YOLO model on this dataset to identify and classify swimming and drowning. Develop a Flask-based web application where users can upload images and videos for detection and receive results in real-time.
Key Features	Utilizes the YOLO deep learning model for fast and accurate identification of objects within the underwater images and videos. Simple and intuitive Flask-based web application to upload and display the results of drowning detection. Supports both images and videos, ensuring flexibility in usage.

### Resource Requirements

Resource Type	Description	Specification/Allocation
<b>Hardware</b>		
Computing Resources	CPU/GPU specifications, number of cores	Intel(R) Core (TM) i5-1135G7, 4 cores
Memory	RAM specifications	8 GB
Storage	Disk space for data, models, and logs	1 TB SSD
<b>Software</b>		
Frameworks	Python frameworks	Flask
Libraries	Additional libraries	Scikit-learn, pandas, Numpy, tensorflow
Development Environment	IDE	Google Collab, Spyder
<b>Data</b>		
Data	Source, size, format	Kaggle dataset, 9000 images