SUMMER INTERNSHIP PROGRAM ON SOFTWARE DEVELOPMENT

An Industrial Internship Report

submitted by

Frank Mathew Sajan

23BCE8713

in partial fulfilment for the award of the degree of

Bachelor of Technology [B.Tech]

in

Computer Science and Engineering

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

SEPTEMBER 2025

DECLARATION BY THE CANDIDATE

I hereby declare that the Industrial Internship report entitled "**Summer Internship Program on Software Development"** submitted by me to VIT-AP University, Amaravati in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a record of bonafide industrial training undertaken by me under the supervision of **Dr. Santanu Mandal, TechtoGreen Drone & Robotics Pvt. Ltd.** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Frank

Frank Mathew Sajan
23BCE8713

OFFER LETTER

+91 8100053089
techtogreen02@gmail.com
https://www.techtogreen.com/

Date: 24/05/2025

Subject: Internship Confirmation at TECHTOGREEN DRONE & ROBOTICS PRIVATE LIMITED

Dear Frank Mathew Sajan,

Thank you for your application and interest in joining TECHTOGREEN DRONE & ROBOTICS PRIVATE LIMITED for your internship.

We are pleased to confirm your internship with us from **1st June 2025 to 15th July 2025**. During this period, you will have the opportunity to contribute to ongoing projects and gain valuable hands-on experience in the field of drone and robotics technology. We look forward to having you on board and seeing your contributions during your time with us.

Best regards,
Parnab Das

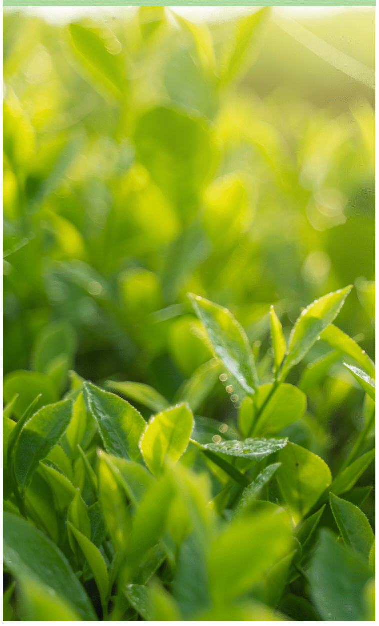
TECHTOGREEN DRONE & ROBOTICS PVT. LTD. ★ C.E.O.

Mr. Parnab Das
C.E.O. & Founder,
TECHTOGREEN DRONE & ROBOTICS PRIVATE LIMITED

TECHTOGREEN DRONE & ROBOTICS PRIVATE LIMITED

INTERNSHIP COMPLETION CERTIFICATE

Summer Internship Certificate



TECHTOGREEN
DRONE & ROBOTICS PVT. LTD.

is presenting this certificate to

Mr. Frank Mathew Sajan

for his completion of the Summer Internship Program on *Software Development* from 01/06/2025 to 15/07/2025. He showed outstanding commitment, technical curiosity, and a proactive attitude throughout the internship, completing all assigned tasks with sincerity and notable effort.

Parnab Das

Mr. Parnab Das
CEO

Dr. Santanu Mandal
Advisor, Technology & Inspection

ACKNOWLEDGEMENT

Place: Amaravati
Date: 05-09-2025 Frank Mathew Sajan

I express my sincere gratitude to my training officer and faculty supervisor for their guidance and invaluable support throughout the internship. Their expertise and encouragement greatly contributed to my learning experience and professional development. I also thank the organization for providing access to essential resources, facilities, and technologies necessary for carrying out my training. Additionally, I acknowledge the contributions of colleagues and research staff whose insights enriched my understanding of AI applications in energy management and legal analytics.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Abbreviations

VO: Visual Odometry
MAC-VO: Multi-Algorithm Combination Visual Odometry
GPS: Global Positioning System
IoT: Internet of Things
DO: Dissolved Oxygen
EC: Electrical Conductivity
NPK: Nitrogen, Phosphorus, Potassium (soil nutrients)
ML: Machine Learning
RMSE: Root Mean Squared Error
MAE: Mean Absolute Error
UI: User Interface
IMU: Inertial Measurement Unit

Symbols & Nomenclature

R: Rotation matrix recovered from Essential matrix decomposition
t: Translation vector recovered from Essential matrix decomposition
K: Camera intrinsic calibration matrix
E: Essential matrix
x, y, z: 3D coordinate axes
θ: Rotation angle

CHAPTER 1: INTRODUCTION

1.1 General

This internship provided the opportunity to work across two interdisciplinary domains: autonomous drone navigation and water quality monitoring using embedded systems and machine learning. The training combined both hardware and software aspects, ranging from computer vision for navigation to IoT-based rover tracking, market research of sensors, and predictive analytics for environmental monitoring. The intent was to bridge academic knowledge with real-world applications, thereby enhancing both technical depth and problem-solving capabilities.

1.2 Objectives of the Internship

The primary objectives of the internship were:

- To develop a **visual odometry (VO) system** using MAC-VO for enabling drone navigation in GPS-denied environments.
- To design and implement an **IoT-based rover tracking system** using Raspberry Pi and GPS modules, supported by a real-time web interface.
- To perform **market research on sensors** relevant to Agri-tech and aqua-tech domains, documenting specifications, costs, and applicability.
- To contribute to a **predictive water quality monitoring model**, leveraging machine learning on large-scale datasets (2.8M+ records).

1.3 Problem Statement

Autonomous systems such as drones and rovers face significant challenges when operating in **GPS-denied environments** like indoor spaces, dense urban regions, or obstructed terrains. In such scenarios, accurate **pose estimation and localization** are critical for safe navigation. Similarly, in environmental monitoring, there is a growing need for **real-time sensing** of soil and water parameters, as well as **predictive models** that can forecast water quality trends weeks in advance. The specific problem areas addressed during this internship were:

- Developing a vision-based navigation system using **monocular visual odometry**.
- Building an application to display the **location of a rover and nearby obstacles** using Raspberry Pi.
- Conducting **sensor market research** to evaluate the most suitable hardware for agricultural and aquaculture applications.
- Providing a **preprocessed dataset and ML pipeline** for predicting water quality parameters.

1.4 Scope of Work

The scope of this internship spanned both **software development** and **applied research**:

- **Computer Vision & Robotics**: Implementing MAC-VO for visual odometry, testing feature detection, motion estimation, and trajectory mapping.
- **Embedded Systems & IoT**: Developing a rover tracking system using Raspberry Pi, Arduino, GPS modules, and a Django-based web interface for real-time monitoring.
- **Sensor Technology**: Evaluating sensors for soil, nutrient, and water monitoring, comparing integrated kits and individual modules based on performance and cost.
- **Machine Learning & Data Science**: Building preprocessing pipelines, normalizing large datasets, and testing regression models for predictive water quality analytics.

This diverse scope ensured exposure to multiple technologies, enabling a holistic understanding of how **computer vision, IoT, and data science** converge in real-world applications.

CHAPTER 2: ABOUT THE ORGANIZATION

TechToGreen Drone and Robotics Pvt. Ltd., founded in 2025 by faculty, scholars, and students of VIT-AP University, specializes in delivering autonomous and cost-effective drone and robotic solutions for real-world challenges. The company focuses on sectors such as precision agriculture, surveillance, waste management, worker safety, and environmental monitoring, combining advanced research with practical engineering.

TechToGreen's mission is to create eco-friendly, reliable, and intelligent systems that enhance operational efficiency while maintaining sustainability. The company provides solutions that enable real-time problem solving, precise operations, risk reduction, and automation at a competitive cost.

The organization has achieved 10+ patents, 15+ projects, and 20+ awards, reflecting its commitment to research-driven innovation. Its technology stack integrates autonomous hardware, embedded systems (Raspberry Pi, Arduino), IoT platforms, AI algorithms, and cloud-based analytics, ensuring products are adaptable, precise, and scalable.

Operating from its registered office in Kolkata and operational office at VIT-AP University, Amaravati, TechToGreen fosters a culture of innovation, collaboration, and sustainability. During my internship, I gained hands-on exposure to the company's core domains, working on projects in autonomous drone navigation, rover GPS tracking, sensor research, and water quality monitoring, which provided invaluable practical experience aligned with academic learning.

CHAPTER 3: SKILLSET PRIOR TO INTERNSHIP

Prior to the internship, I had a strong foundation in full-stack development, with hands-on experience in Python and JavaScript, including frameworks like Django for REST APIs, ORM, and authentication. I was also familiar with mobile app development using React Native and Expo, as well as desktop applications through Cordova and ElectronJS. My database knowledge spanned PostgreSQL, Firebase, SQLite, and MongoDB, enabling me to manage and query data effectively.

I possessed solid algorithmic thinking and competitive programming skills, alongside introductory experience in AI and machine learning using TensorFlow and OpenCV. I had explored applications of AI in projects but was still in the early stages of hands-on deployment. Additionally, I had exposure to embedded systems and IoT, including basic programming on Raspberry Pi and Arduino, providing a foundation for sensor interfacing and data acquisition.

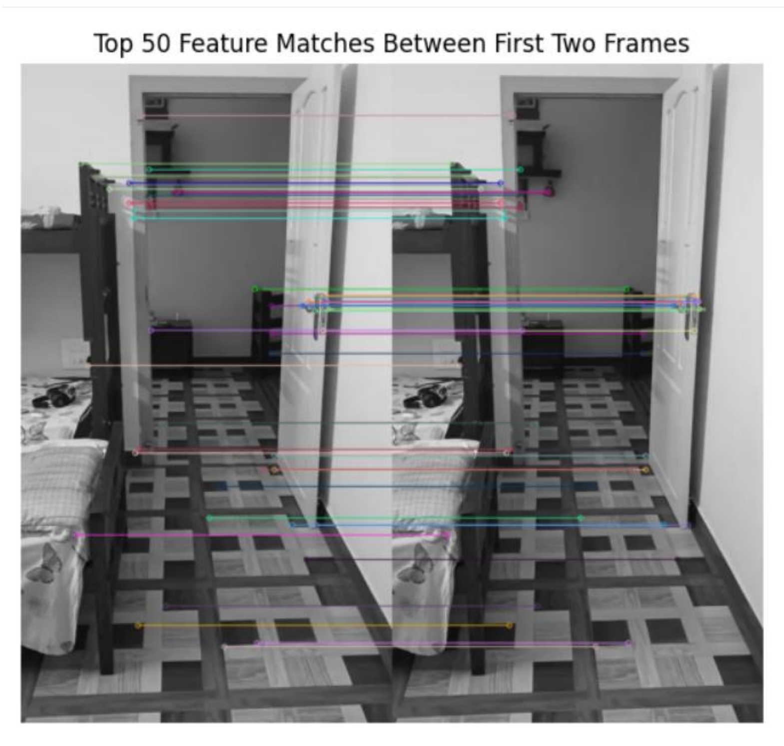
Alongside technical skills, I actively pursued research in Biotechnology, and Robotics, striving to bridge academic learning with real-world applications. While I had not yet achieved major accomplishments, my focus was on exploring emerging technologies, maintaining academic excellence, and preparing for practical, interdisciplinary projects such as those undertaken during this internship.

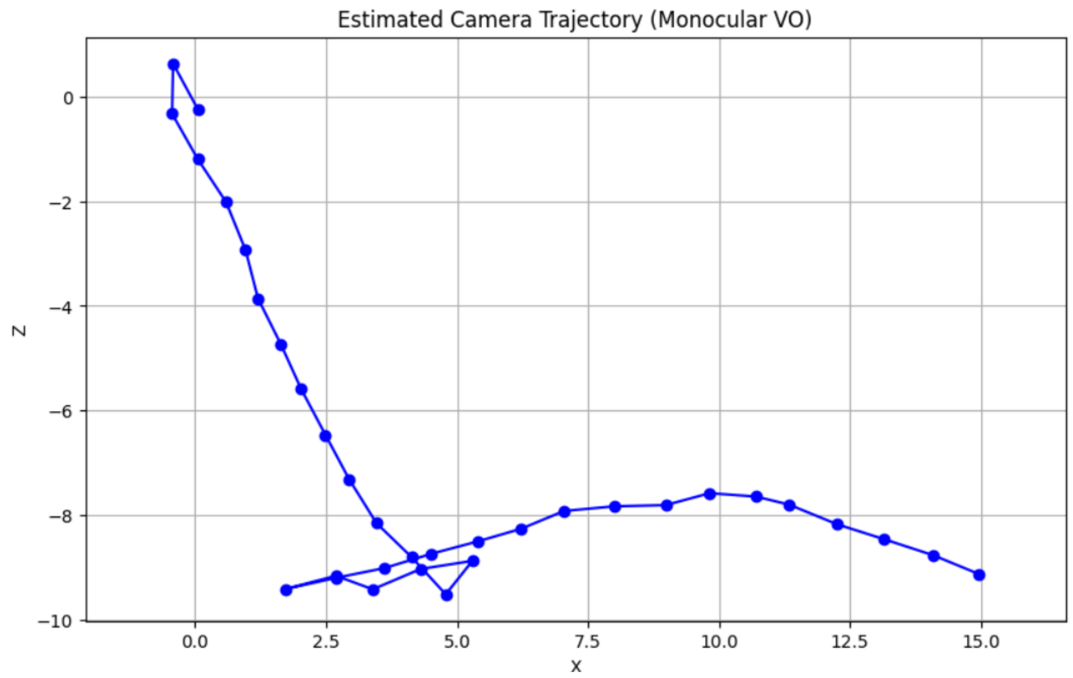
CHAPTER 4: KNOWLEDGE ACQUIRED FROM IN-PLANT TRAINING

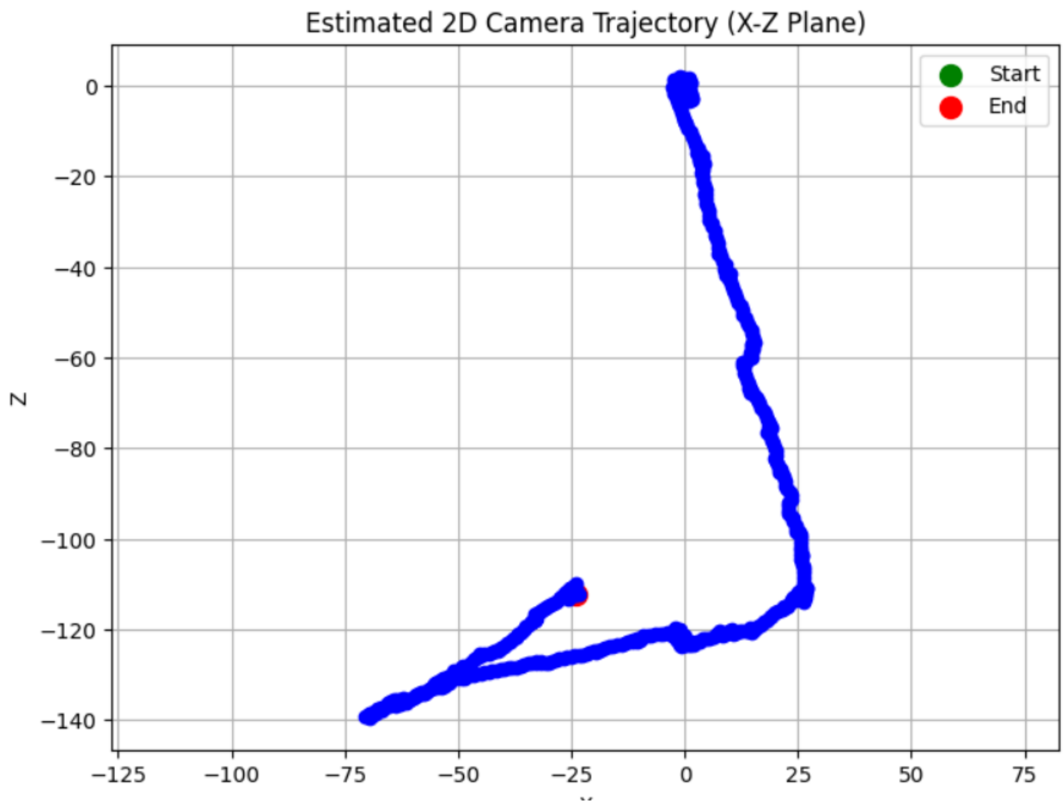
During the internship at TechToGreen, I acquired hands-on experience in autonomous drone navigation, IoT-based rover monitoring, sensor evaluation, and water quality modeling. The training enabled me to apply theoretical concepts from computer science, electronics, and data science to practical real-world projects, enhancing both my technical and analytical capabilities.

4.1 Visual Odometry System (MAC-VO)

I developed a monocular visual odometry (VO) system to estimate the motion of drones in GPS-denied environments. Using Python, OpenCV, and the Essential Matrix, I implemented feature detection and matching with ORB, estimated camera egomotion, and visualized the drone’s trajectory in 2D plots. Advanced features included creating overlay videos displaying the live feed with persistent trajectory maps, start/end markers, and dynamic frame skipping to optimize performance. This work strengthened my understanding of computer vision, motion estimation, and trajectory visualization.

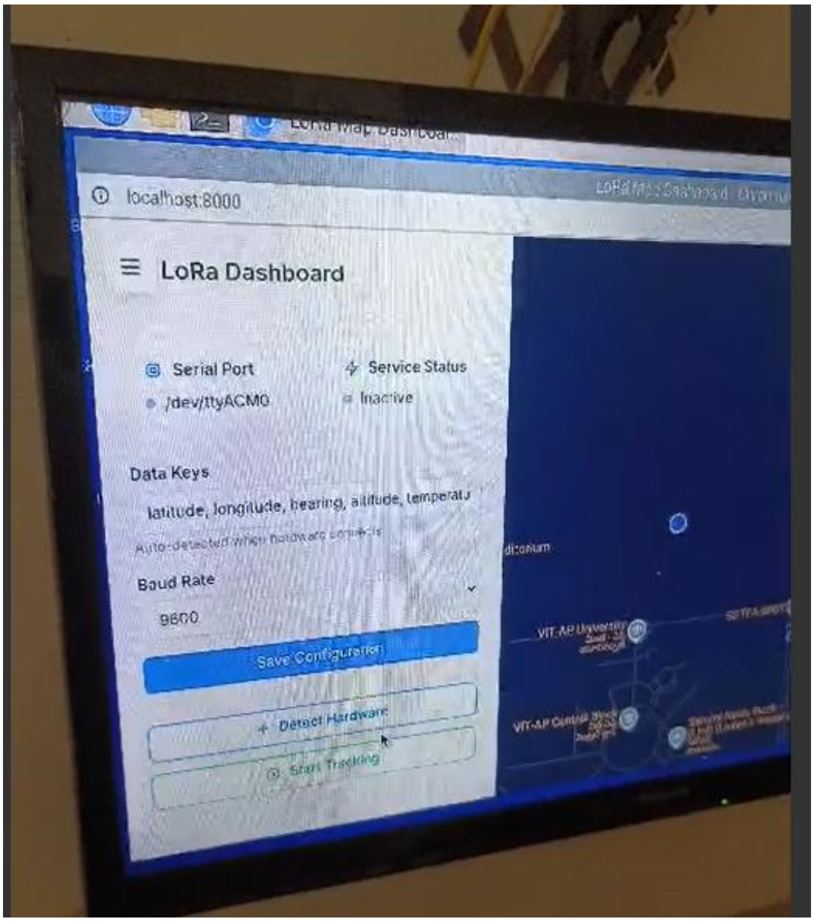



Estimated Camera Trajectory (Monocular VO)
Z
X


Estimated 2D Camera Trajectory (X-Z Plane)
Start
End
Z

4.2 Rover Monitoring System

I built an embedded system using Raspberry Pi 2 and Arduino to track a rover's position via GPS. A web-based interface displayed real-time location data and obstacle mapping, while field testing verified system responsiveness and GPS accuracy. Through this project, I gained practical experience in IoT integration, real-time monitoring, and web application deployment.



4.3 Sensor Market Research

I conducted detailed market research on sensors for both Agri-tech and aqua-tech applications. This included evaluating soil moisture, EC, NPK sensors for agriculture, and DO, ammonia, pH, chlorophyll-a sensors for water monitoring. I compared individual sensors versus integrated kits, documented specifications, costs, and supplier links, and prepared a structured report. This enhanced my skills in technical evaluation, product comparison, and documentation for engineering decision-making.

Table 1: Summary of Selected Smart Agriculture and Aquaculture Sensors

Sensor Model	Function	Working Principle	Approx. Price ()	Source
IISc/IIT-B Soil Moisture Sensor	Soil moisture	Heat pulse–based alternative, low power	1,000–3,000	Link
Argus Smart Agriculture Sensor	Soil moisture, temp., humidity	IoT-enabled probes, wireless data transmission	2,500	Link
NPK Soil Sensor Kit	Soil nutrients (N, P, K)	Ion-selective electrodes with analog output	3,000–7,000	Amazon / Local
Farmonaut IoT Sensor	Soil moisture & temperature with AI insights	IoT + satellite integration for crop advisory	5,000–12,000	Link
Smart Marine Aquaculture System	Water quality (pH, DO, salinity)	ESP32 wireless sensor network, cloud monitoring	15,000–20,000	Link

4.4 Water Quality Prediction Model

I assisted in building a predictive model for water quality parameters using a real-world dataset of 2.8 million records. Tasks included data preprocessing, cleaning, normalization, and preparation for regression modeling. Early experiments indicated promising predictive capability, demonstrating the application of machine learning in environmental monitoring and data-driven decision-making.

CHAPTER 5: APPLICATION OF GAINED KNOWLEDGE

The internship provided practical experience in integrating **computer vision, embedded systems, IoT, and machine learning** to solve real-world challenges. Each project involved applying theoretical concepts, troubleshooting hardware-software interactions, and building end-to-end pipelines for demonstration and testing purposes.

5.1 Sample Demonstration: Visual Odometry (Colab)

A monocular visual odometry system was implemented using MAC-VO in Google Colab. The workflow included video data loading, ORB feature detection, feature matching, and estimation of camera motion using the Essential Matrix and camera intrinsic parameters. The trajectory of the drone was visualized through 2D plots and overlay videos with persistent minimaps showing start/end points and gridlines. This demonstration illustrated practical motion estimation, trajectory tracking, and real-time visualization techniques in GPS-denied environments.

5.2 Rover GPS Tracking & Web Interface

An IoT-based rover monitoring system was developed using Raspberry Pi 2, Arduino, and GPS modules. A Python-Django web interface displayed the rover's real-time location and mapped obstacles. Field testing confirmed accurate GPS readings and minimal latency in updates. This project reinforced skills in embedded system integration, real-time data handling, and web-based monitoring solutions.

5.3 Sensor Research Documentation

Comprehensive research was conducted to evaluate sensors for agriculture and water quality applications. Key parameters studied included soil moisture, EC, NPK (Agri), and DO, ammonia, pH, chlorophyll-a (aqua). A comparison between integrated kits and individual sensors was made, documenting specifications, prices, and supplier information. The resulting report facilitated hardware selection and procurement decisions, showcasing the importance of thorough technical evaluation and documentation.

5.4 Water Quality Dataset Preprocessing & Modeling

A water quality prediction model was prepared using a 2.8 million record dataset. Tasks included data cleaning, missing value handling, normalization, and preparation for regression modeling. Preliminary experiments demonstrated promising predictions for 1–2 weeks in advance, emphasizing the applicability of machine learning for environmental monitoring. This project enhanced understanding of large-scale data preprocessing, model training, and predictive analytics.

CHAPTER 6: COMPETENCY COMPARISON AND SELF-EVALUATION

Before the internship, my technical foundation included Python and JavaScript programming, basic AI/ML knowledge, full-stack development, and embedded system concepts. I had experience with web development frameworks (React, Django), database management (PostgreSQL, Firebase, MongoDB), and algorithmic problem solving. However, my exposure to real-world interdisciplinary projects combining drones, IoT, and environmental monitoring was limited.

During the internship, I developed hands-on skills in visual odometry, IoT-based rover monitoring, sensor evaluation, and large-scale data preprocessing for predictive modeling. I learned to integrate hardware and software systems, work with GPS modules, Raspberry Pi, Arduino, and develop web interfaces for real-time monitoring. I also enhanced my understanding of computer vision algorithms, trajectory estimation, and predictive analytics using real-world datasets.

Comparing my competency levels before and after the internship, the growth is significant in terms of practical problem-solving, project integration, and application of AI/ML in interdisciplinary domains. I am now confident in tackling projects that require the fusion of embedded systems, robotics, and environmental data analysis. This experience has strengthened my self-learning abilities, adaptability, and collaborative skills, which are critical for future research and professional work in tech-driven domains.

7-Week Progress Plan

Week 1: Orientation & Setup

- Joined TechToGreen and attended induction sessions to understand organizational workflow and ongoing projects.
- Studied the objectives of autonomous drone navigation and water quality monitoring projects.
- Set up local and cloud environments for coding and data processing (Google Colab, Python libraries, Raspberry Pi).
- Installed necessary libraries for computer vision, data analysis, and machine learning.
- Explored the datasets and sensor specifications relevant to Agri-tech and aqua-tech applications.

Week 2: Visual Odometry – Data Exploration

- Loaded sample video datasets and explored video frames.
- Implemented feature detection and matching using ORB features.
- Tested essential matrix computation for monocular visual odometry.
- Began understanding camera intrinsics, egomotion estimation, and trajectory visualization.
- Documented initial observations and noted challenges with low-texture frames.

Week 3: Visual Odometry – Implementation & Testing

- Developed the monocular VO system using MAC-VO pipeline.
- Implemented frame-to-frame rotation and translation accumulation for trajectory estimation.
- Created 2D trajectory plots and overlaid minimaps on video for visual validation.
- Debugged feature tracking errors and optimized frame-skipping for performance.
- Tested the system on sample videos and evaluated drift accuracy.

Week 4: Rover Monitoring System Development

- Designed a GPS-based rover tracking system using Raspberry Pi and Arduino.
- Implemented backend using Python and Django REST API for data collection.
- Developed a basic web interface for live location visualization.
- Conducted field tests to verify GPS accuracy and system responsiveness.
- Documented sensor readings, latency issues, and UI improvements.

Week 5: Sensor Market Research

- Identified sensors suitable for Agri-tech (soil moisture, EC, NPK) and aqua-tech (DO, ammonia, pH, chlorophyll-a).
- Compared integrated kits versus individual sensors in terms of cost, accuracy, and ease of integration.
- Compiled a structured report with detailed sensor specifications, pricing, and links.
- Discussed findings with the team to finalize hardware selection for pilot deployments.

Week 6: ***Water Quality Prediction Modeling***

- Acquired and preprocessed a real-world dataset containing 2.8M+ water quality records.
- Handled missing values, normalized input features, and explored key parameters.
- Built baseline regression models to predict water quality parameters 1–2 weeks in advance.
- Evaluated initial performance and identified areas for model improvement.
- Coordinated with mentors to align preprocessing techniques with future predictive modeling plans.

Week 7: Integration, Testing & Documentation

- Conducted end-to-end testing of VO system, rover tracking, and sensor interfaces.
- Created sample applications to validate communication stacks.
- Documented project workflows, system limitations, and learning outcomes.
- Prepared final reports, presentations, and shared code repositories with mentors.
- Reflected on overall internship experience, key learnings, and potential future enhancements.

APPENDICES

Appendix 1: Sample Code for Visual Odometry (MAC-VO)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load video from Google Drive
cap = cv2.VideoCapture('/content/drive/MyDrive/video.mp4')

# ORB detector and matcher
orb = cv2.ORB_create(2000)
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

# Dummy camera intrinsics (you can estimate better later)
focal_length = 718.8560  # use approx. width of video in pixels
center = (640, 360)      # for 1280x720 or similar
K = np.array([[focal_length, 0, center[0]],
              [0, focal_length, center[1]],
              [0, 0, 1]])

# Trajectory storage
trajectory = []
R_total = np.eye(3)
t_total = np.zeros((3, 1))

frame_idx = 0
step = 15  # Skip 15 frames between VO steps

def get_keypoints_descriptors(gray):
    return orb.detectAndCompute(gray, None)

def get_matched_points(kp1, kp2, matches):
    pts1 = np.float32([kp1[m.queryIdx].pt for m in matches])
    pts2 = np.float32([kp2[m.trainIdx].pt for m in matches])
    return pts1, pts2

while True:
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)
    ret1, frame1 = cap.read()
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx + step)
    ret2, frame2 = cap.read()

    if not (ret1 and ret2):
        break

    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    gray2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
```

Appendix 2: Rover GPS Tracking and Django Web Interface

```python
def start(request):
    global serial_thread
    config = AppConfig.get_config()

    if not all([config.baud_rate, config.serial_port]):
        print("Configuration is incomplete.")
        return JsonResponse({'error': 'Configuration not complete'}, status=400)

    def serial_worker():
        channel_layer = get_channel_layer()
        print("Serial worker thread started.")
        try:
            with serial.Serial(config.serial_port, config.baud_rate, timeout=1) a
                while True:
                    try:
                        line = ser.readline().decode('utf-8').strip()
                        if line:
                            print(f"Received line from serial port: {line}")
                            try:
                                data = json.loads(line)
                                print(f"Parsed JSON data: {data}")

                                # Send all data to WebSocket clients
                                try:
                                    async_to_sync(channel_layer.group_send)(
                                        "lora_updates",
                                        {
                                            "type": "lora_message",
                                            "data": data  # Send the entire data
                                        }
                                    )
                                    print("Data sent to WebSocket clients")
                                except Exception as e:
                                    print(f"Error sending to WebSocket: {str(e)}"

                            except json.JSONDecodeError as e:
                                print(f"JSON decoding error: {str(e)}")
                            except KeyError as e:
                                print(f"Missing key error: {str(e)}")
                    except UnicodeDecodeError as e:
                        print(f"Error decoding serial data: {str(e)}")
                        continue

        except serial.SerialException as e:
            print(f"Serial error: {str(e)}")

    # Start the thread if it's not already running
    if serial_thread is None or not serial_thread.is_alive():
        serial_thread = threading.Thread(target=serial_worker, daemon=True)
        serial_thread.start()
        print("Serial worker thread has been started.")

    return JsonResponse({'status': 'Serial communication started'}, status=200)
```

Appendix 3: Sensor Research Report with Comparison

Conclusion

Yes, sensors for all four parameters exist, with robust options for both integrated and separate deployment. Key challenges: fouling, calibration, and interference are mitigated through modern features like automatic wipers, smart calibration, and algorithmic compensation. For continuous monitoring:

- **Opt for multiparameter sondes** (e.g., YSI EXO, BOQU MPG-6099) for comprehensive data with anti-fouling.
- **Choose modular sensors** (e.g., Renke, LZ) if targeting specific parameters or harsh conditions.
  Always validate sensor specifications against water matrix (e.g., salinity, turbidity) and prioritize field-serviceable designs.

2. Top 11 water quality sensors for water treatments easily - Renke

3. MPG-6099 Multiparameter Water Quality Meter | Boqu

4. WATER QUALITY CONTINUOUS MONITORING SYSTEM - EXO SONDES

5. TOP 13 COMMONLY USED WATER QUALITY SENSORS IN AQUACULTURE

7. MULTI-PARAMETER WITH PH EC TURBIDITY TSS COD AMMONIA TDS WATER QUALITY SENSORS

9. TOP 11 WATER QUALITY SENSORS FOR WATER TREATMENTS EASILY-ATECH SENSOR CO., LTD.

11. DISSOLVED OXYGEN SENSORS FOR STREAM AND RIVER MONITORING

Appendix 4: Water Quality Dataset Preprocessing Pipeline and Modelling Outputs

One of my primary responsibilities during the internship was to conduct **dataset discovery and curation** for the water quality prediction task. After reviewing multiple repositories and academic sources, I identified a highly valuable dataset:

- **A Comprehensive Surface Water Quality Monitoring Dataset (1940–2023)** hosted on *Figshare*.
- Contains over **2.82 million records**, making it one of the largest public datasets for water quality monitoring.
- Includes parameters such as dissolved oxygen (DO), temperature, pH, and other CCME indices.
- Targets of interest: **CCME_Values** and **CCME_WQI**, both widely used in water quality assessment.
-

This dataset was shortlisted due to its **scale, longitudinal coverage, and open availability** for academic research.

Although my role did not involve direct model training, I assisted the team by:

1. **Evaluating dataset quality**: Ensured metadata consistency and verified parameter definitions.
2. **Preliminary preprocessing guidance**: Suggested handling of missing values (e.g., forward-fill for time series).
3. **Organizing subsets**: Extracted and catalogued samples of the dataset (~10k records) for model prototyping.
4. **Documentation**: Shared references and related literature (e.g., PMC article on ML-based water quality prediction) with the team.

REFERENCES

- OpenCV: Open-source computer vision library for image processing and feature tracking: https://docs.opencv.org
- MAC-VO: Monocular visual odometry framework for drone navigation: https://github.com/MAC-VO/MAC-VO
- Python: Programming language used for data processing, modeling, and scripting: https://www.python.org
- Pandas: Data analysis and manipulation library: https://pandas.pydata.org
- NumPy: Numerical computation library for matrix and vector operations: https://numpy.org
- Scikit-learn: Machine learning library for model building and evaluation: https://scikit-learn.org
- Matplotlib & Seaborn: Data visualization libraries for plotting trajectories and sensor data: https://matplotlib.org, https://seaborn.pydata.org
- Raspberry Pi Documentation: Embedded platform for IoT and rover applications: https://www.raspberrypi.com/documentation/
- Arduino Documentation: Microcontroller programming and interfacing with sensors: https://www.arduino.cc/en/Guide/HomePage
- GPS Modules (e.g., Neo-6M) Datasheets and Documentation: Real-time location tracking for embedded systems
- Fig share Water Quality Dataset: Public dataset used for predictive modeling: https://figshare.com/
- IoT Sensor Market References: Product catalogs and datasheets for soil moisture, EC, NPK, pH, DO, ammonia, and chlorophyll-a sensors