Contents

Project Background and Introduction

Project Overview

This project management plan outlines the approach for developing a software vulnerability detection system as part of the Innovation Project. The chosen project focuses on “Detecting software vulnerabilities” from the available project options.

The system will use machine learning technology to automatically identify potential security vulnerabilities in software code. This addresses the current challenge where manual code reviews are time-intensive and may miss emerging vulnerability patterns.

The project will integrate a trained AI model with a web-based interface to create a practical demonstration of AI technology applied to cybersecurity challenges.

Project Goals and Objectives

Primary Goals

1. Develop a comprehensive IT project management plan report, detailing the project scope, timeline, and risk management strategies.
2. Successfully train an AI model specifically tailored to detect software vulnerabilities in code.
3. Design and deploy a dynamic website that interacts with the trained AI model, enabling end users to test code for potential vulnerabilities.

Specific Objectives

- Plan and execute a structured development approach across four key phases
- Implement effective project management practices including scope, time, and resource management
- Develop technical components that meet the course requirements for AI integration
- Deliver comprehensive documentation and project reporting
- Demonstrate successful team collaboration and project delivery within specified timeframes

Expected Outcomes

This project management plan establishes the framework for delivering a software vulnerability detection system that meets the academic requirements while demonstrating practical application of project management principles.

Upon completion, this project will deliver:

1. **Technical Deliverables**:
   - Working prototype with AI model integration
   - React-based web frontend
   - Python/FastAPI backend implementation
   - Functional demonstration system
2. **Documentation**: Project management documentation including this plan, progress reports, risk management plans, and final project report as required by course assessments
3. **Prototype/Demo**: A working demonstration system that showcases the integration of the required technologies and validates the project approach
4. **Knowledge Contribution**: Practical experience in project management methodologies, team collaboration, and technology integration within an academic project framework

Team Introduction

Team Overview

Our team consists of 3 members, each bringing unique skills and perspectives to the Computing Technology Innovation Project. The team was formed on the 4th of August 2025, and has been meeting weekly to ensure effective collaboration and progress tracking.

Team Members

Team Member 1: Harrison Jones

- **Role**: [Insert role]
- **Student ID**: 100578691
- **Email**: 100578691@student.swin.edu.au
- **Background**: Computer Science student, majoring in Cybersecurity. Moderate experience in a variety of programming languages. Passion for all things linux.
- **Responsibilities**:
  - [Insert primary responsibility 1]
  - [Insert primary responsibility 2]
  - [Insert primary responsibility 3]

Team Member 2: Thisari Nethara Jayaweera

- **Role**: [Insert role]
- **Student ID**:
- **Email**: (**student.swin.edu.au?**)
- **Background**: [Brief description of relevant skills, experience, and academic background]
- **Responsibilities**:
  - [Insert primary responsibility 1]
  - [Insert primary responsibility 2]
  - [Insert primary responsibility 3]

Team Member 3: Vimeth Manchala Wijerathna

- **Role**: [Insert role]
- **Student ID**:
- **Email**: (**student.swin.edu.au?**)
- **Background**: [Brief description of relevant skills, experience, and academic background]
- **Responsibilities**:
  - [Insert primary responsibility 1]
  - [Insert primary responsibility 2]
  - [Insert primary responsibility 3]

Team Communication

Communication Channels

- **Primary Communication**: Zoom, WhatsApp
- **File Sharing**: Github
- **Project Management**: GitHub
- **Meeting Platform**: Zoom

Meeting Schedule

- **Regular Meetings**: Saturday of Sunday mornings
- **Duration**: 1.5 hours
- **Location**: Online platform

Team Collaboration Framework

Decision Making Process

[Describe how your team makes decisions - consensus, majority vote, etc.]

Conflict Resolution

[Describe your approach to resolving conflicts within the team]

Knowledge Sharing

[Describe how team members share knowledge and expertise]

Skills Matrix

Team Member	Technical Skills	Soft Skills	Project Contribution
[Name 1]	[List skills]	[List skills]	[Primary contribution areas]
[Name 2]	[List skills]	[List skills]	[Primary contribution areas]
[Name 3]	[List skills]	[List skills]	[Primary contribution areas]
[Name 4]	[List skills]	[List skills]	[Primary contribution areas]

Team Development Plan

Training and Development

- [List any planned training or skill development activities]
- [Identify areas where team members need additional support]

Mentorship and Support

- [Describe any mentorship arrangements within the team]
- [Identify external support resources available]

Project Requirements List and Description

Functional Requirements

Core Functionality

FR1: AI Model Integration

- **Description**: The system must integrate a trained machine learning model capable of analyzing software code for potential vulnerabilities
- **Priority**: High
- **Source**: Course requirements and project objectives
- **Acceptance Criteria**:
  - AI model can process code input and identify potential vulnerabilities
  - Model provides confidence scores for detected vulnerabilities
  - Integration with backend API is successful and functional

FR2: Code Analysis Processing

- **Description**: System must accept software code input and process it through the vulnerability detection model
- **Priority**: High
- **Source**: Core business requirement
- **Acceptance Criteria**:
  - System accepts multiple programming language inputs
  - Processing completes within reasonable time limits
  - Results are returned in structured format

FR3: Vulnerability Reporting

- **Description**: System must generate detailed reports of detected vulnerabilities with relevant information
- **Priority**: High
- **Source**: User requirements and project objectives
- **Acceptance Criteria**:
  - Reports include vulnerability type, location, and severity
  - Reports provide recommended remediation steps
  - Reports can be exported in multiple formats

User Interface Requirements

FR4: React Web Frontend

- **Description**: Develop a responsive web interface using React framework for user interaction with the vulnerability detection system
- **Priority**: High
- **Source**: Course technology requirements
- **Acceptance Criteria**:
  - Interface built using React framework
  - Responsive design works on desktop and mobile devices
  - User can upload/input code for analysis

FR5: User Experience Flow

- **Description**: Provide intuitive user journey from code input through analysis to results display
- **Priority**: Medium
- **Source**: User experience best practices
- **Acceptance Criteria**:
  - Clear navigation and workflow guidance
  - Progress indicators during processing
  - Results displayed in user-friendly format

Data and Integration Requirements

FR6: Python Backend with FastAPI

- **Description**: Implement backend services using Python and FastAPI framework to handle API requests and model integration
- **Priority**: High
- **Source**: Course technology requirements
- **Acceptance Criteria**:
  - Backend implemented using Python and FastAPI
  - RESTful API endpoints for frontend communication
  - Secure handling of code input and analysis results

FR7: Database Integration

- **Description**: Implement data storage solution for analysis history and user sessions
- **Priority**: Medium
- **Source**: Technical requirements for data persistence
- **Acceptance Criteria**:
  - Database stores analysis results and metadata
  - Data retrieval is efficient and reliable
  - Database schema supports required data structures

Scope Management

Project Scope Statement

Project Scope Description

This project involves the development of an innovation project for detecting software vulnerabilities using machine learning technology. The system will integrate a trained Large Language Model (LLM) with a web-based interface to provide automated vulnerability detection capabilities.

The project scope encompasses the development of a comprehensive software vulnerability detection system, including:

- Planning and requirements analysis with UI/UX design
- Implementation of web architecture with Python backend API
- Monitoring and control systems with quality assurance processes
- Deployment and user acceptance testing with final documentation

Work Breakdown Structure (WBS)

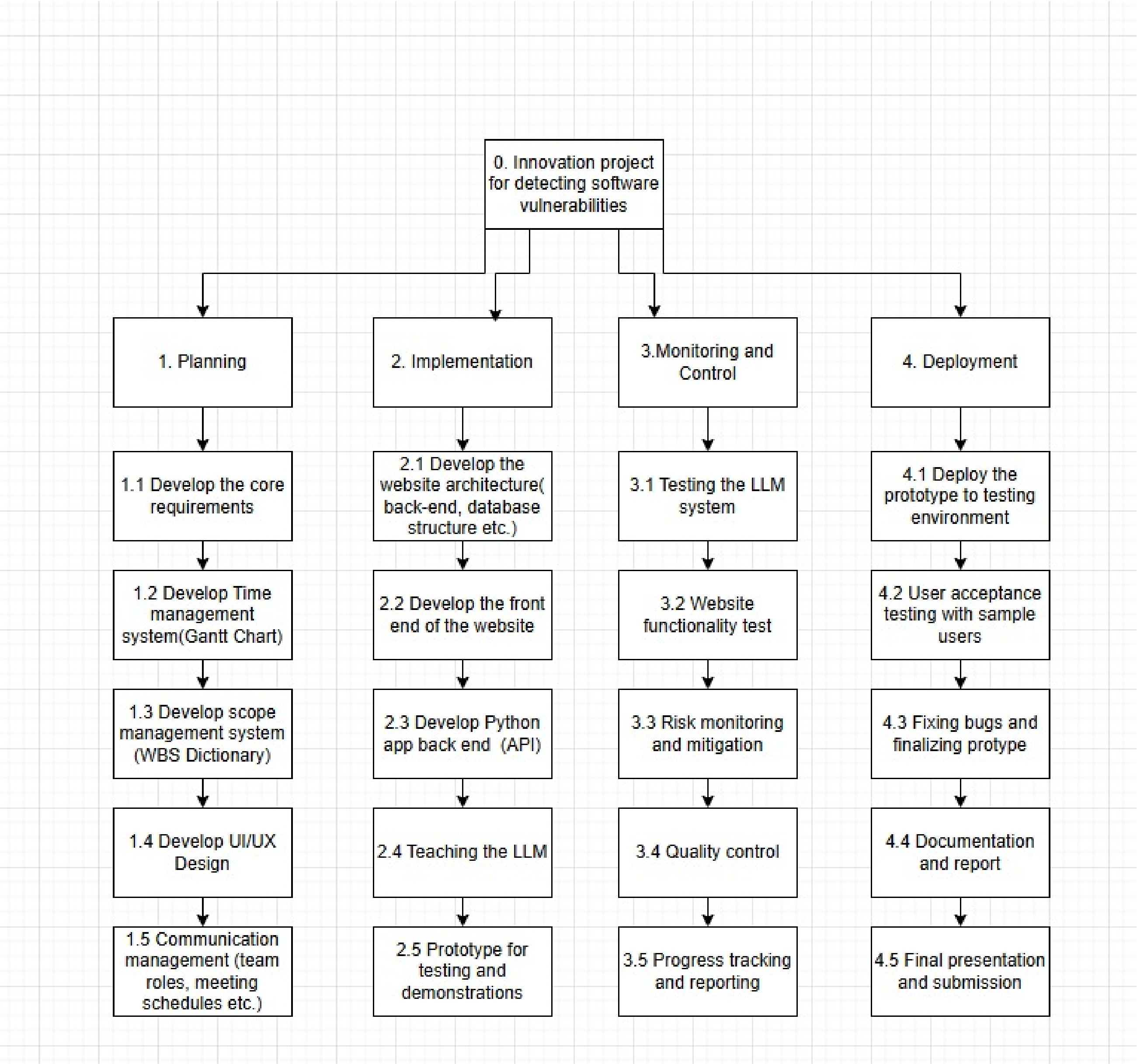

0. Innovation project for detecting software vulnerabilities
1. Planning
2. Implementation
3.Monitoring and Control
4. Deployment
1.1 Develop the core requirements
1.2 Develop Time management system(Gantt Chart)
1.3 Develop scope management system (WBS Dictionary)
1.4 Develop UI/UX Design
1.5 Communication management (team roles, meeting schedules etc.)
2.1 Develop the website architecture(back-end, database structure etc.)
2.2 Develop the front end of the website
2.3 Develop Python app back end (API)
2.4 Teaching the LLM
2.5 Prototype for testing and demonstrations
3.1 Testing the LLM system
3.2 Website functionality test
3.3 Risk monitoring and mitigation
3.4 Quality control
3.5 Progress tracking and reporting
4.1 Deploy the prototype to testing environment
4.2 User acceptance testing with sample users
4.3 Fixing bugs and finalizing protype
4.4 Documentation and report
4.5 Final presentation and submission

Figure 1: Work Breakdown Structure diagram showing the hierarchical decomposition of project deliverables

WBS Dictionary

1. Planning

WBS ID: 1.0
Description: Project planning and preparation phase including requirements analysis and design.

1.1 Develop the Core Requirements WBS ID: 1.1
Description: Analyze and document functional and non-functional requirements for the vulnerability detection system.
Deliverables: Requirements specification document
Duration: 1 week

1.2 Develop Time Management System (Gantt Chart) WBS ID: 1.2
Description: Create comprehensive project timeline with milestones and dependencies.
Deliverables: Gantt chart and project schedule
Duration: 3 days

1.3 Develop Scope Management System (WBS Dictionary) WBS ID: 1.3
Description: Define project scope boundaries and create detailed work breakdown structure.
Deliverables: WBS Dictionary and scope statement
Duration: 3 days

1.4 Develop UI/UX Design WBS ID: 1.4
Description: Design user interface mockups and user experience workflows for the web application.
Deliverables: UI mockups, wireframes, and UX documentation
Duration: 1 week

1.5 Communication Management WBS ID: 1.5
Description: Establish team roles, communication protocols, and meeting schedules.
Deliverables: Communication plan and team charter
Duration: 2 days

2. Implementation

WBS ID: 2.0
Description: Development and implementation of the software vulnerability detection system.

2.1 Develop the Website Architecture WBS ID: 2.1
Description: Build backend infrastructure, database structure, and system architecture.
Deliverables: Backend system, database schema, system architecture documentation
Duration: 2 weeks

2.2 Develop the Front End of the Website WBS ID: 2.2
Description: Implement user interface and frontend functionality based on UI/UX designs.
Deliverables: Functional web frontend
Duration: 2 weeks

2.3 Develop Python App Back End (API) WBS ID: 2.3
Description: Create API endpoints and backend services for system integration.
Deliverables: RESTful API, backend services
Duration: 2 weeks

2.4 Teaching the LLM WBS ID: 2.4
Description: Train and fine-tune the Large Language Model for vulnerability detection.
Deliverables: Trained LLM model, training documentation
Duration: 3 weeks

2.5 Prototype for Testing and Demonstrations WBS ID: 2.5
Description: Integrate all components into a working prototype system.
Deliverables: Functional prototype system
Duration: 1 week

3. Monitoring and Control

WBS ID: 3.0
Description: Testing, quality assurance, and project monitoring activities.

3.1 Testing the LLM System WBS ID: 3.1
Description: Conduct comprehensive testing of the machine learning model performance.
Deliverables: Test results, performance metrics
Duration: 1 week

3.2 Website Functionality Test WBS ID: 3.2
Description: Test all web application features and user interface functionality.
Deliverables: Test reports, bug tracking documentation
Duration: 1 week

3.3 Risk Monitoring and Mitigation WBS ID: 3.3
Description: Monitor project risks and implement mitigation strategies.
Deliverables: Risk register updates, mitigation reports
Duration: Ongoing

3.4 Quality Control WBS ID: 3.4
Description: Ensure all deliverables meet quality standards and requirements.
Deliverables: Quality assurance reports
Duration: Ongoing

3.5 Progress Tracking and Reporting WBS ID: 3.5
Description: Monitor project progress and generate status reports.
Deliverables: Progress reports, status updates
Duration: Ongoing

4. Deployment

WBS ID: 4.0
Description: System deployment, testing, and project closure activities.

4.1 Deploy the Prototype to Testing Environment WBS ID: 4.1
Description: Deploy the completed system to a testing environment for validation.
Deliverables: Deployed system in testing environment
Duration: 3 days

4.2 User Acceptance Testing with Sample Users WBS ID: 4.2
Description: Conduct user acceptance testing with representative users.
Deliverables: UAT results, user feedback documentation
Duration: 1 week

4.3 Fixing Bugs and Finalizing Prototype WBS ID: 4.3
Description: Address identified issues and finalize the system for delivery.
Deliverables: Final system version, bug fix documentation
Duration: 1 week

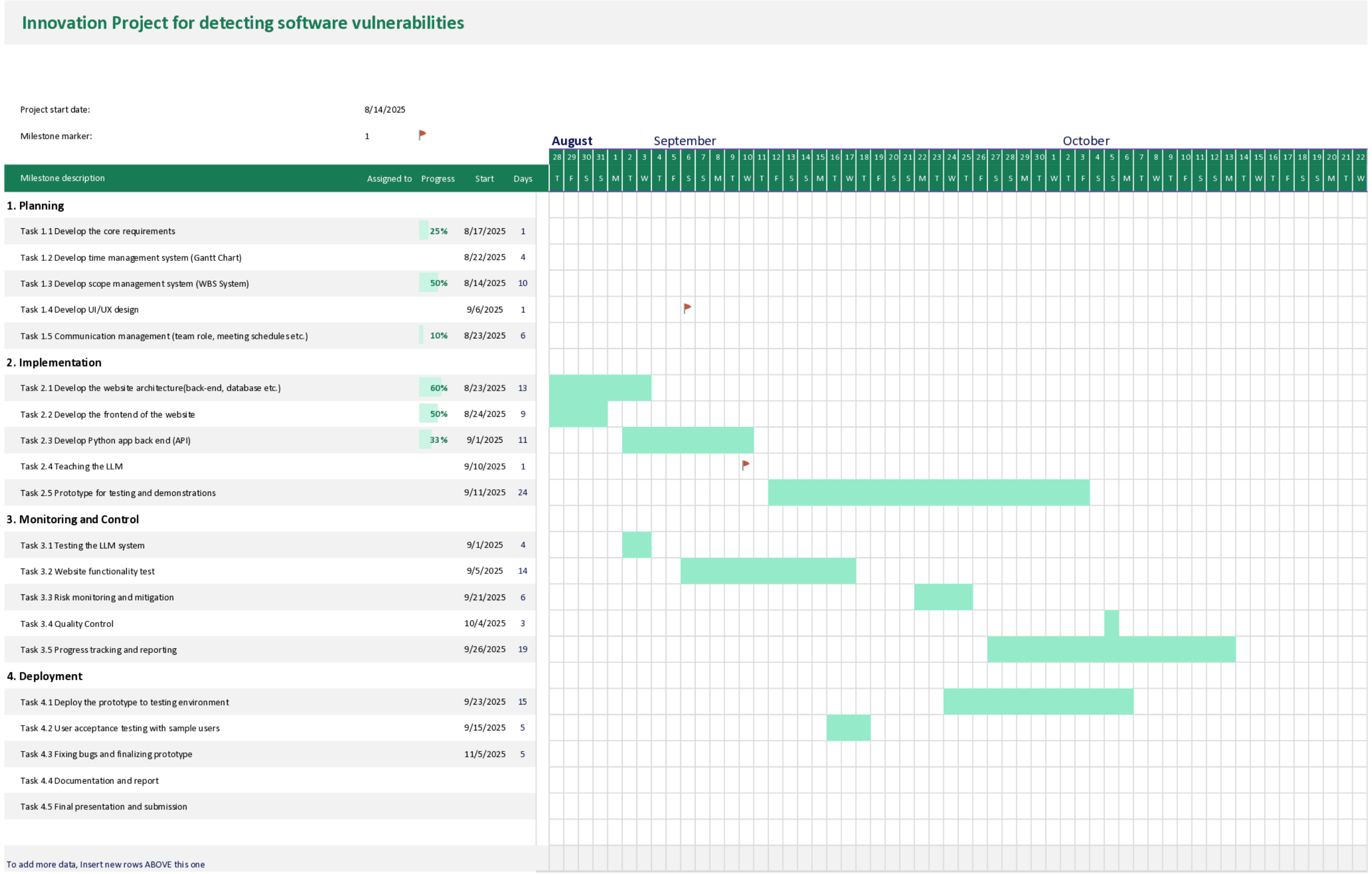
4.4 Documentation and Report WBS ID: 4.4
Description: Complete all project documentation and final report.
Deliverables: Technical documentation, user manual, project report
Duration: 1 week

4.5 Final Presentation and Submission **WBS ID:** 4.5
Description: Prepare and deliver final project presentation and submission.
Deliverables: Presentation materials, final project submission
Duration: 3 days

Time Management

Gantt Chart

Below is the Gantt chart illustrating the expected time allocation for the project. This visual representation outlines the key project phases and milestones leading up to the submission deadline.



Innovation Project for detecting software vulnerabilities

Project start date: 8/14/2025

Milestone marker: 1 🚩

Milestone description	Assigned to	Progress	Start	Days
1. Planning				
Task 1.1 Develop the core requirements		25%	8/17/2025	1
Task 1.2 Develop time management system (Gantt Chart)			8/22/2025	4
Task 1.3 Develop scope management system (WBS System)		50%	8/14/2025	10
Task 1.4 Develop UI/UX design			9/6/2025	1
Task 1.5 Communication management (team role, meeting schedules etc.)		10%	8/23/2025	6
2. Implementation				
Task 2.1 Develop the website architecture(back-end, database etc.)		60%	8/23/2025	13
Task 2.2 Develop the frontend of the website		50%	8/24/2025	9
Task 2.3 Develop Python app back end (API)		33%	9/1/2025	11
Task 2.4 Teaching the LLM			9/10/2025	1
Task 2.5 Prototype for testing and demonstrations			9/11/2025	24
3. Monitoring and Control				
Task 3.1 Testing the LLM system			9/1/2025	4
Task 3.2 Website functionality test			9/5/2025	14
Task 3.3 Risk monitoring and mitigation			9/21/2025	6
Task 3.4 Quality Control			10/4/2025	3
Task 3.5 Progress tracking and reporting			9/26/2025	19
4. Deployment				
Task 4.1 Deploy the prototype to testing environment			9/23/2025	15
Task 4.2 User acceptance testing with sample users			9/15/2025	5
Task 4.3 Fixing bugs and finalizing prototype			11/5/2025	5
Task 4.4 Documentation and report				
Task 4.5 Final presentation and submission				

To add more data, Insert new rows ABOVE this one

Figure 2: Gantt Diagram showing the planned timeline for the software vulnerability project

Risk Management

Monitor and Control

Closure Plan

Project Design

Design Overview

This section outlines the technical design approach for the Computing Technology Innovation Project, including front-end prototypes, user interface design, and alignment with established usability principles.

Front-end Prototype Development

Meeting Minutes

Team Information

- **Team Name:** Session 21 (Tutor: Zehang Deng) - Group 3
- **Meeting No:** 1
- **Location:** BA408
- **Date:** Monday 11/08/2025
- **Attending:** Whole group

Previous Meeting

- **Matters arising from previous minutes:** No (First meeting)
- **Confirmation of minutes from last meeting:** N/A (First meeting)

Meeting Agenda & Outcomes

1. Communication Platform

Discussion: Need to establish primary communication channels for quick responses and collaboration.

Decision: - Primary: WhatsApp for quick daily communication - **Secondary:** Discord for group calls and file sharing

Action Items: - All team members to join WhatsApp group - Set up Discord server with appropriate channels - **Responsible:** Whole group - **Timeline:** Ongoing

2. Regular Meeting Schedule

Discussion: Establish consistent meeting times to monitor project progress.

Decision: Weekly meetings on Saturday or Sunday mornings at school.

Action Items: - Confirm specific day and time for weekly meetings - **Responsible:** Whole group - **Timeline:** Ongoing

3. File Sharing & Version Control

Discussion: Determine platforms for collaborative work and version management.

Decision: - Version Control: GitHub for file sharing and version control - **Live Collaboration:** VS Code Live Share extension for simultaneous editing

Action Items: - Set up GitHub repository with team access - Install VS Code Live Share extension - **Responsible:** Whole group - **Timeline:** Ongoing

Summary of Action Items

1. Join WhatsApp group and Discord server (All members)
2. Finalize weekly meeting schedule (All members)
3. Set up GitHub repository and VS Code Live Share (All members)

Meeting Closure

- **Meeting Ended:** [Time to be filled]
- **Next Meeting:** [Date, time, and location to be scheduled]
- **Minutes Recorded By:** [Name to be filled]