1. Introduction

Bughound is a web-based bug recording and tracking software designed to streamline the management of bugs, employees, and program areas. The project is divided into three iterations:

- **Iteration 1**: Develop the overall application structure, including the GUI, dashboard, and navigation.
- **Iteration 2**: Add programs, areas, and employee forms to add new records into the system.
- **Iteration 3**: Implement bug page elements and associated forms to add, update, and search for bugs.

This document presents a comprehensive schedule for the Bughound project, including effort and duration estimates, task dependencies, milestones, critical path analysis, and a timeline for project completion.

2. Task Set for Bughound

Iteration 1: Application Structure and Navigation

- **Task 1.1**: Project Setup and Environment Configuration
    - Set up a development environment, tools, and version control systems.
- **Task 1.2**: Design User Interface (UI) and Dashboard
    - Create wireframes and design the application's look and feel.
- **Task 1.3**: Develop Home Screen and Dashboard
    - Implement the dashboard with navigation links.
- **Task 1.4**: Implement Navigation Logic
    - Develop routing and navigation between pages and forms.

Iteration 2: Add Programs, Areas, and Employee Forms

- **Task 2.1**: Develop Program Management Module
    - Create forms to add and manage program records.
- **Task 2.2**: Develop Area Management Module
    - Create forms to add and manage area records.
- **Task 2.3**: Develop Employee Management Module
    - Create forms to add and manage employee records.
- **Task 2.4**: Integrate Management Modules with Navigation

    - Link new modules to the dashboard and ensure smooth navigation.

Iteration 3: Bug Page Elements and Associated Forms

- **Task 3.1**: Develop New Bug Entry Form
    - Create a form to log new bugs with required fields.
- **Task 3.2**: Develop Bug Search Functionality
    - Implement search with filters to find bug reports.
- **Task 3.3**: Develop Bug Results Page
    - Display search results in a table format.
- **Task 3.4**: Develop Update Bug Form
    - Create a form to modify existing bug reports.
- **Task 3.5**: Integrate Bug Management with Navigation and Dashboard
    - Add bug management features to navigation and update the dashboard.

Additional Tasks

- **Task 4.1**: Develop Database Maintenance Page
    - Create an administrative page to manage data and export functionality.
- **Task 4.2**: Testing and Quality Assurance
    - Conduct unit, integration, and system testing.
- **Task 4.3**: Final Deployment and Documentation
    - Prepare for deployment and create user manuals and technical documentation.

---

3. Defining Parallel Work Activities

To optimize the project timeline, certain tasks can be executed in parallel:

- **Tasks 1.1 and 1.2**: Can start simultaneously since they are independent.
- **Tasks 2.1, 2.2, and 2.3**: After Task 1.4, these tasks can proceed in parallel as they involve different modules.
- **Tasks 3.1 and 3.2**: Can be performed in parallel after Task 2.4.
- **Task 4.1**: Can begin after Task 2.4 and run concurrently with Iteration 3 tasks.

4. Establishing Milestones

The following milestones are established to track the project's progress:

- **Milestone 1**: Completion of Application Structure and Navigation (End of Iteration 1)
- **Milestone 2**: Completion of Program, Area, and Employee Management Modules (End of Iteration 2)
- **Milestone 3**: Completion of Bug Management Features (End of Iteration 3)
- **Milestone 4**: Completion of Testing and Quality Assurance
- **Milestone 5**: Final Deployment and Documentation

4.1. Task Network with Effort and Duration

Task ID	Task Name	Effort (Person-Hours)	Duration (Days)	Predecessor(s)
1.1	Project Setup and Environment Config	16	1	None
1.2	Design UI and Dashboard	32	4	None
1.3	Develop Home Screen and Dashboard	24	2	1.2
1.4	Implement Navigation Logic	16	1	1.1, 1.3

Task ID	Task Name	Effort (Person-Hours)	Duration (Days)	Predecessor(s)
2.1	Develop Program Management Module	24	2	1.4
2.2	Develop Area Management Module	24	2	1.4
2.3	Develop Employee Management Module	32	3	1.4
2.4	Integrate Management Modules	16	1	2.1, 2.2, 2.3

Task ID	Task Name	Effort (Person-Hours)	Duration (Days)	Predecessor(s)
3.1	Develop New Bug Entry Form	32	3	2.4
3.2	Develop Bug Search Functionality	32	3	2.4
3.3	Develop Bug Results Page	24	2	3.2
3.4	Develop Update Bug Form	24	2	3.1
3.5	Integrate Bug Management Features	16	1	3.3, 3.4

Task ID	Task Name	Effort (Person-Hours)	Duration (Days)	Predecessor(s)
4.1	Develop Database Maintenance Page	24	2	2.4
4.2	Testing and Quality Assurance	40	4	3.5, 4.1
4.3	Final Deployment and Documentation	16	1	4.2

5. Determining the Critical Path

The critical path is the longest sequence of tasks that determines the minimum project duration.

Critical Path Tasks:

To generate the critical path for this project, we need to identify the longest sequence of dependent tasks with the highest cumulative duration. Here's the analysis:

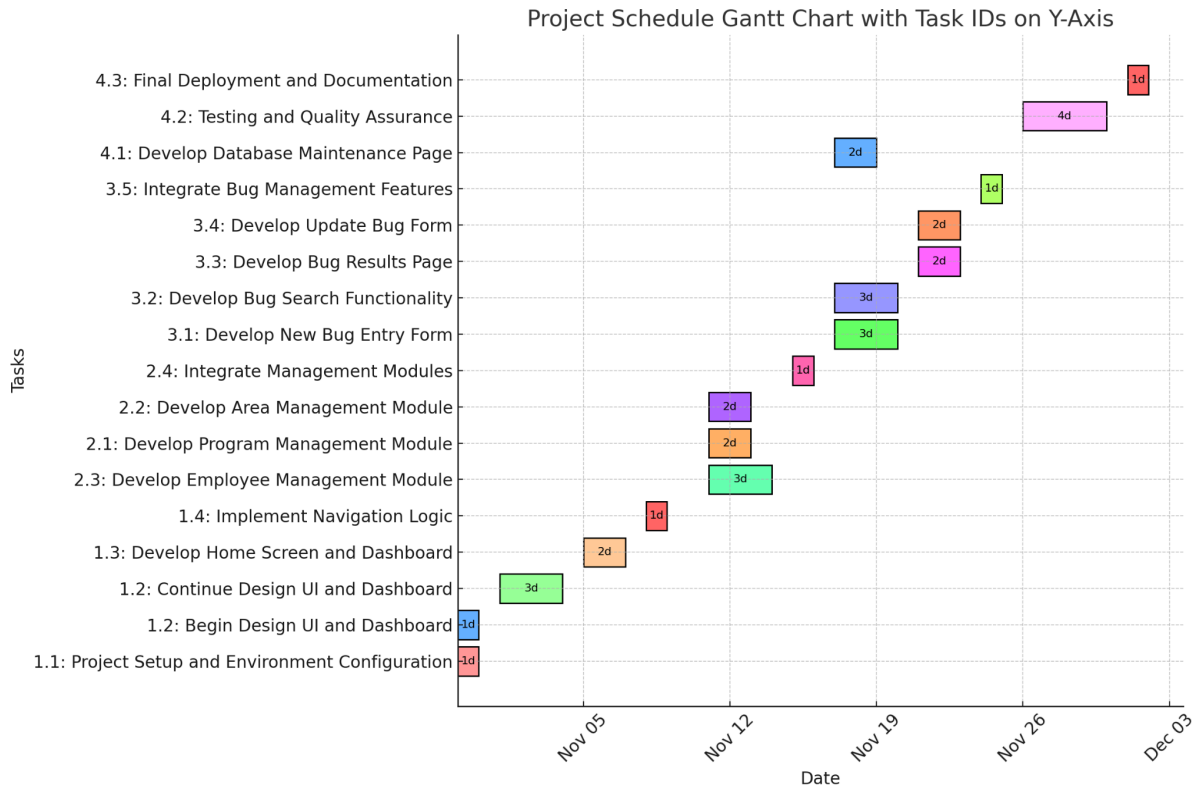
1. Identify Paths
    - Path A: 1.1 → 1.4 → 2.1 → 2.4 → 3.1 → 3.4 → 3.5 → 4.2 → 4.3
    - Path B: 1.2 → 1.3 → 1.4 → 2.1 → 2.4 → 3.2 → 3.3 → 3.5 → 4.2 → 4.3
2. Calculate Duration for Each Path
    - Path A Duration = 1 + 1 + 2 + 1 + 3 + 2 + 1 + 4 + 1 = 16 days
    - Path B Duration = 4 + 2 + 1 + 2 + 1 + 3 + 2 + 1 + 4 + 1 = 21 days
3. Critical Path
   The longest path is Path B with a total duration of 21 days. This is the critical path, as it determines the minimum project completion time.

Total Duration on Critical Path:

- **Total Days**: 4 + 3 + 2 + 4 + 2 + 4 + 3 + 2 + 5 + 2 = **31 days**

6. Timeline for Bughound Completion

Assuming a 5-day workweek, the project spans approximately **6.5 weeks**.



Project Setup and Initial Design

- 1.1: Project Setup and Environment Configuration - **1 day** (Nov 3)
- 1.2: Begin Design UI and Dashboard - **1 day** (Nov 4)
- 1.2: Continue Design UI and Dashboard - **3 days** (Nov 5 - Nov 7)
- 1.3: Develop Home Screen and Dashboard - **2 days** (Nov 7 - Nov 8)

Develop Core Functionality

- 1.4: Implement Navigation Logic - **1 day** (Nov 9)
- 2.1: Develop Program Management Module - **2 days** (Nov 11 - Nov 12)
- 2.2: Develop Area Management Module - **2 days** (Nov 12 - Nov 13)
- 2.3: Develop Employee Management Module - **3 days** (Nov 13 - Nov 15)
- 2.4: Integrate Management Modules - **1 day** (Nov 15)

Bug Management

- 3.1: Develop New Bug Entry Form - **3 days** (Nov 16 - Nov 18)
- 3.2: Develop Bug Search Functionality - **3 days** (Nov 19 - Nov 21)
- 3.3: Develop Bug Results Page - **2 days** (Nov 21 - Nov 22)
- 3.4: Develop Update Bug Form - **2 days** (Nov 22 - Nov 23)
- 3.5: Integrate Bug Management Features - **1 day** (Nov 24)

Finalization and Deployment

- 4.1: Develop Database Maintenance Page - **2 days** (Nov 26 - Nov 27)
- 4.2: Testing and Quality Assurance - **4 days** (Nov 27 - Nov 30)
- 4.3: Final Deployment and Documentation - **1 day** (Dec 1)