JOBCONNECT

PROJECT PLAN

PRESENTED BY

MEMBER 1: WAJEEHA KAMRAN(LEADER)

MEMBER 2: TAHA FAYYAZ

MEMBER 3: NAILA HABIB

SUPERVISED BY

Dr. Onaiza Maqbool

Department of Computer Science

Quaid e Azam University, Islamabad

PROJECT PLAN DOCUMENT APPROVEL SIGNATURES

PROJECT MANAGER

_____ _____

Signatur Date

WAJEEHA KAMRAN

Student, QAU Islamabad

MEMBER 1

_____ _____

Signature Date

TAHA FAYYAZ

Student, QAU Islamabad

MEMBER 2

_____ _____

Signature Date

NAILA HABIB

Student, QAU Islamabad

STAKEHOLDER

_____ _____

Signature Date

DR. ONAIZA MAQBOOL

Professor, QAU Islamabad

CHANGE HISTORY :

VERSION NO.	DATE	MODIFIED BY	CHANGES
1	07/10/2024	Wajeeha Kamran	Original

REMARKS:

Signature

DR.ONAIZA MAQBOOL

Professor, QAU Islamabad

TABLE OF CONTENTS

List of figures
List of tables

1.Project overview

1.1 Project Summary

The main aim of this project is to develop a software that allows companies to post jobs on the platform and applicants to apply for the jobs by submitting their CV's. The system then automatically shortlists suitable applicants according to the job requirements and sets up appointments for them with the relevant companies for interviews. Jobs are automatically deleted from the platform after their deadlines.

1.2 Project Purpose, Scope and Objectives

Purpose:

To make a hiring platform where **companies post jobs** , applicants can **submit their cv or fill out a built-in form** to apply for job , **streamline shortlisting of candidates , appointment scheduling** and **automatic deletion of the job posts** once candidates are hired.

Scope :

>Context

This is a **standalone system** managing the recruitment process and overseeing the scheduling of interviews .

>Functions

Functional requirements:

The system will allow ：

- ✓ Companies to **post jobs** on the platform.
- ✓ Applicants to apply for jobs **by filling out a built-in form** acting as a CV that can **add, modify or remove** details.
- ✓ **Automatic shortlisting of candidates** according to job requirements.
- ✓ **Scheduling appointments** for the shortlisted candidates.
- ✓ **Feedback** provided to both company and the applicant about the appointment scheduling.
- ✓ **Automatic deletion** of job post after the deadline.

Non-Functional requirements:

- ✓ **Usability**: The user interface should be intuitive, enabling users to complete their tasks easily.
- ✓ **Scalability**: The architecture should allow for easy scaling to accommodate increased user load without major changes to the system.
- ✓ **Security**: The system should implement data encryption and user authentication to protect sensitive information.

Objectives:

- ✓ To provide a **centralized platform** for individuals and companies to easily communicate with each other
- ✓ To make the process of online recruitments process easy and efficient.
- ✓ Providing applicants a platform where they can easily apply **and track their application** status .
- ✓ **Automatic managing** of shortlisting applications and appointment scheduling reduces manual effort.

1.3 Assumptions and Constraints

Constraints :

- ✓ The software will be built using **Java programming language**.
- ✓ The system will be developed as an **app-based platform**.
- ✓ It will be provided as **a free software**.
- ✓ The platform will be available only in **English language** .
- ✓ The project must be completed by **14 weeks of development window .**
- ✓ The software will have **a limited initial user base** which may impact the efficiency of recruitment process.

Assumptions:

- ✓ Assuming that there is a **significant demand for recruitment software** and will also fulfill **core need in the job market**.
- ✓ Assuming that the companies and the applicants have the **basic knowledge of such software systems** allowing them to use and navigate through the platform easily.
- ✓ Assuming that the companies will **provide accurate job descriptions and criteria.**
- ✓ Assuming that the **applicants will maintain up-to-date profiles and submit relevant and accurate information**.

1.3 Project Deliverables

- ✓ A detailed document at each phase :
  - Project Plan: A document that specifies the overall software development strategy and timeline.
  - Software Requirements Specification (SRS): A document detailing the entire functionality of the software and use cases associated with each requirements.

- Analysis Document: A document specifying detailed features and providing mockups to illustrate interactions between companies and users.
- Design Document: A document describing the design prototypes and architectural details of the system.

✓ **Fully functional software** that will incorporate all the core features.
✓ **Quality assurance and testing reports , comprehensive reviews of work product** , phase or a milestone.
✓ Work products at each meeting internal to the team.
✓ **Design prototypes** representing the system visually
✓ **User guidelines , system manuals** to assist users in navigating the software.

1.3 Schedule Summary

✓ **Phase 1** (Planning): 1 week
✓ **Phase 2** (Analysis): 3 weeks
✓ **Phase 3** (Design): 4 weeks
✓ **Phase 4** (Development): 4 weeks
✓ **Phase 5** (Testing): 3 weeks

2.References

1. IEEE Software Construction Guideline. ISO/IEC/IEEE 16326:2019(E)
2. ProjectLibre manual version 0.1-October 6,2012.

3.Definitions

- ✓ IEEE: Institute of Electrical and Electronics Engineers
- ✓ ISO: The International Organization for Standardization
- ✓ CV: Curriculum Vitae, a document used by applicants to detail their qualifications, experience, and skills.
- ✓ Shortlisting: The process of filtering and selecting suitable candidates based on predefined criteria.
- ✓ Appointment Scheduling: Automated system for setting up interview times between candidates and companies.
- ✓ Deadline: a date or time before which something must be done.
- ✓ Deliverables: Product (outcome) after completion of a particular phase.

4.Project Context

4.1 Process Model

- The project will adopt an **iterative and incremental process model** for developing our Software **Job Connect**. This approach ensures **flexibility, adaptability** throughout the development cycle and allows **continuous feedback from stakeholders** throughout the development cycle.

- The **Scrum** model used, ensure regular communication with the stakeholders and rapid, incremental and iterative delivery of key features of our Software.

Project Activities:

- ✓ The Major Project Activities include **Gathering of Requirements,**

System Design, Development, Coding and **Testing** and **Deployment** which will be carried out in **sprints** an advantage of using **Scrum** model which will deliver **functional increments** that includes:

- Job Posting from Companies
- CV Submissions from Job Seekers
- Shortlisting Candidates by Companies
- Scheduling of Appointments
- Notifying after Shortlisting the Candidates
- Deleting Posts after deadline

with regular **reviews** at the end of each **sprint** for gathering **feedback** and for doing **refinements** if required.

Information Flow and Work Products:

- ✓ Flow of Information and Work Products will be shared in each **sprint** and to track the core features listed in the document.
- ✓ We will have Key Milestones to show the completion of these core features.
- ✓ At the end of each **sprint,** to ensure the deliverables meet the required technical standards, Reviews and Approvals will take place.

Project Initiation and Termination:

- ✓ The initiation phase of the project involves defining **objectives, timelines and resource allocations.**
- ✓ The termination phase of the project involves **user acceptance and successful deployment** of the project. To ensure all the deliverables have met the expectations of user and documentation is processed.

- ✓ A **final Review** will be assessing **project outcomes for project closure.**

4.2 Methods, Tools and Techniques

Methods:

- ✓ **Scrum Methodology** for managing development through **sprints**.
- ✓ **Object Oriented Programming (OOP) and Java** for managing software architecture.

Tools:

- ✓ **Java** as the Programming language and **Eclipse** as our Code Editor.
- ✓ **MS Word** for the documentation of our project.
- ✓ **Project Libre** for measuring the progress of our project.
- ✓ Git and GitHub to keep crucial versions of our project.
- ✓ **MySQL** as database management to store **Job Listings, Applicant and Registered Users data.**
- ✓ **Figma** for designing our **User Interface and for sketching the whole App.**
- ✓ **Apache Tomcat** that will work on server side and used as a deployer for deploying **Java Applications to end-users.**
- ✓ **JDBC is an API** for creating connection between **MySQL and Eclipse**.

Techniques:

- ✓ Meetings with **Stakeholder** to gather **requirements and feedback, like Job Posting Features, Application Management, Way of Shortlisting Candidates and Application Management**.
- ✓ **Weekly Meetings with the team** to report the progress and remove unnecessary **blockers** during the development.

4.3 Product Acceptance Plan

Acceptance Criteria:

>User Registration and Authentication

- ✓ Users should be able to register with valid Credentials (**username, email address, phone number and password**).
- ✓ Users Authentication must be secure and reliable which insures that only **registered users** can log in to their accounts.

>Job Postings By Companies

- ✓ Companies should be able to create Job Postings with details such as **Job title, Job Description, requirements needed for Job selection and deadlines.**
- ✓ The System should allow Companies to **modify or delete** the Job Posts.
- ✓ Job Posts should be automatically **deleted after the deadline of Job Posts** is met.

>CV Submission by Job Seekers:

- ✓ Applicants must be able to submit their **CV's** to posted jobs with relevant details that is needed in the job post, including applicant's personal information for contacting.
- ✓ Applicants should be allowed to edit or remove their CV's if necessary.
- ✓ The System must confirm successful submission of CV's and should notify the applicant.

>Shortlisting the Candidates:

- ✓ The platform should automatically shortlist the candidates those have included their CV's in that particular post jobs based on predefined criteria set by the Company (e.g. skills, experience).

- ✓ Companies should be able to view the list of shortlisted candidates, and they must be allowed to manually adjust the list if necessary.

>Scheduling Appointments:

- ✓ Companies should be able to schedule interviews with the shortlisted candidates directly through the platform.
- ✓ Both the Company and the Applicant should be able to receive notifications about the scheduled appointments.

>Job and Management of Applicants:

- ✓ Companies should have access to a dashboard to view and manage job postings and list of Applicants.
- ✓ Applicants should be able to track the status of their applications (**e.g. submitted, shortlisted, interview scheduling**).

>Alerts and Notifications:

- ✓ The system should be able to notify the applicants about their job status (e.g., shortlisted, rejected) and schedules of their interviews with the respective Companies.
- ✓ Companies should be able to receive the alerts when the job post deadline approaches.

>Management of Data and Security:

- ✓ All user data, including job listings and CVs, should be securely stored in the MySQL database.
- ✓ The system should be able to adhere data protection **policies, ensuring integrity and confidentiality**.

>Performance and Usability:

- ✓ The platform should be able **to handle multiple concurrent users efficiently**, ensuring a seamless experience for both companies and applicants.

- ✓ The user interface should be intuitive, providing easy navigation for both job postings and CV submissions.

5.Project Planning

This section describes the project's general management plan. Important points in planning are:

>Process Model (Scrum model).

5.1 Project Work Plan.

The project will use an iterative and incremental Scrum process model with sprints that are clearly defined.

Planning, analysis, design, development, and testing phases will all be included in each sprint to guarantee a well-organized flow.

Sprint reviews and meetings will help in tracking progress. At the conclusion of every sprint, milestones will be set to assess deliverables and make any changes for the next sprint.

>Project phases:

Our project consists of these main phases. They will also be the deliverables:

- Planning.
- Analysis.
- Designing.
- Coding/Development.
- Testing.
- Deployment.

>Work activities/tasks in every sprint:

- ✓ Sprint Planning (each sprint needs to be planned before working)
- ✓ Analysis (gathering requirements for each sprint)
- ✓ Design (designing solution for each sprint before development)
- ✓ Development (developing the sprint)
- ✓ Testing (testing each sprint)

>Reviews / Regular meetings:

- ✓ Sprint meeting (At the start of sprint for planning)
- ✓ Reviews (Reviewing every sprint)
- ✓ After-action Review (Check what went well or what needs to be improved)

>Major Milestones:

Completion of each phase and can be delivered.

>Schedule Allocation:

It includes start and finish dates, dependencies and milestones. A Gantt chart will be used to see task sequences and checks the progress. Here time will be given to every activity.

>Resource Allocation:

Resources will be allocated to every phase of our project life-cycle.

→The Backlog and Deliverables:

The high-level phases:

- Planning
- Analysis
- Design
- Development
- Testing

→The Sprint Backlogs:

They are the sub-tasks and in our project we are giving 1 week to each sprint backlog. They can be sprinted, and we work on them

- Development of the project plan, review
- Defining use-cases
- Development of analysis models
- SRS development, review and refinement
- Design principles, data design
- Interface design
- Detailed design
- Review of design and refinement
- Database connectivity
- Refining the front-end
- Development of classes
- Refinement of classes and traceability with design
- Testing of the software system
- Deployment
- Project Presentations

8.Supporting Process Plans

8.1 Risk management:

The plan for Risk Management uses the approach for **Identifying, Analyzing, Prioritizing and Mitigating the risks** throughout the lifecycle of the project. By Addressing potential risk factors, our aim is to minimize their impacts on project success.

Risk Identification:

The Project team will identify the risks that could impact the project using the following methods.

- ✓ **Team sessions** to discuss potential challenges that can occur.
- ✓ **Stakeholder meetings** to gather input from external.
- ✓ **Review the data which is historical** from similar projects for common **risk patterns**.

  The following risk categories have been identified:
  - **Technological Risks:**
    - Issues in Integrating various technologies like not familiar with **Java, GitHub, MySQL, Eclipse, JDBC** or problems with platform scalability.
  - **Risks in Scheduling and Budget:**
    - Delays in development phases, like in analysis or especially in deployment or testing or cost overruns because of not analyzing one of the important tasks.
  - **Personnel Risks:**
    - Unavailability of the team member or turnover, leading to skill gaps or loss of key project knowledge.
  - **Size and Complexity Risks:**
    - Complexity of the Project, especially in shortlisting algorithms, could introduce unanticipated challenges.

Risk Analysis and Prioritization:

Each Risk will be assessed for its probability of occurrence and its impact on the project, using a risk matrix to categorize risks as low, medium, or high priority.

- ✓ **High-priority risks:**
  - (e.g., integration failure or major security vulnerabilities) will be addressed immediately.
- ✓ **Medium-priority risks:**
  - (e.g., minor delays or user integration feedback) will be monitored closely with contingency plans in place.
- ✓ **Low-priority risks:**

  (e.g., minor bugs or less critical delays) will be tracked and mitigated if necessary.

Mitigation of Risks and Contingency Planning:

For each risk that is identified, the following strategies will be used:

- ✓ Technological Risks:
  - Regular reviews of code, testing must be conducted automatically, and robust security measures will be implemented.
- ✓ Risks in Scheduling and Budget:
  - Project Libre will be used to track Milestones. Any delays will trigger immediate review and reallocation of resources.
- ✓ Personnel Risks:
  - Team members will be doing cross-training to cover multiple roles and by developing clear documentation, it will minimize the impact of team changes.
- ✓ Complexity Risks:
  - To break down Complex tasks into manageable parts, a phased, modular development approach will be used, that will reduce the overall risks of system-wide issues.

Tracking Risks and its Evaluation:

Throughout the Project Lifecycle, Risks will be continuously monitored and reassessed, Our Team will:

- ✓ Maintain a **Risk Register** to check all **Identified Risks, Mitigation Strategies, and their Status.**
- ✓ Conduct **Weekly Risk Review Meetings** to evaluate Risk Levels.
- ✓ Use of **GitHub and Project Libre** will be done to track progress and identify emerging risks.

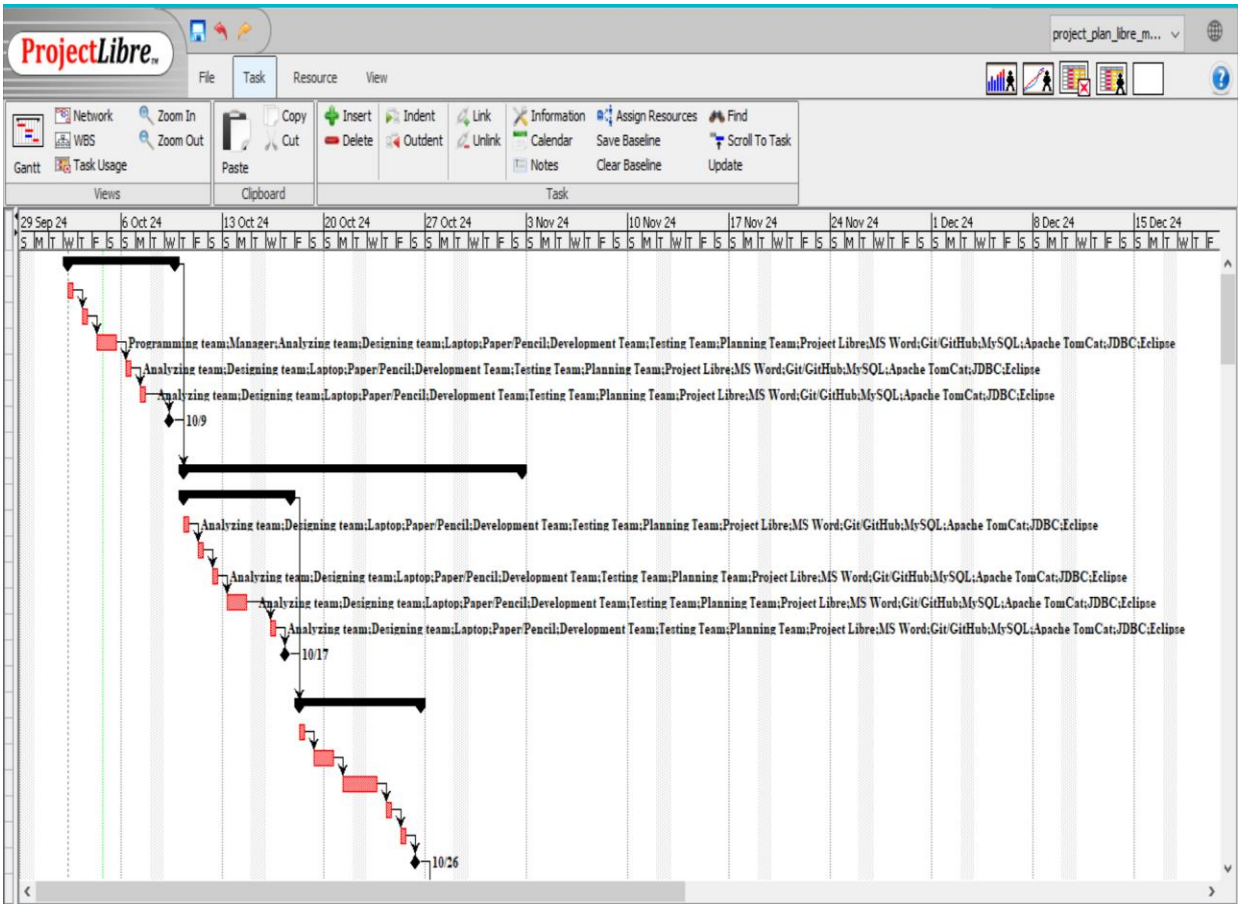
Risks Communication:

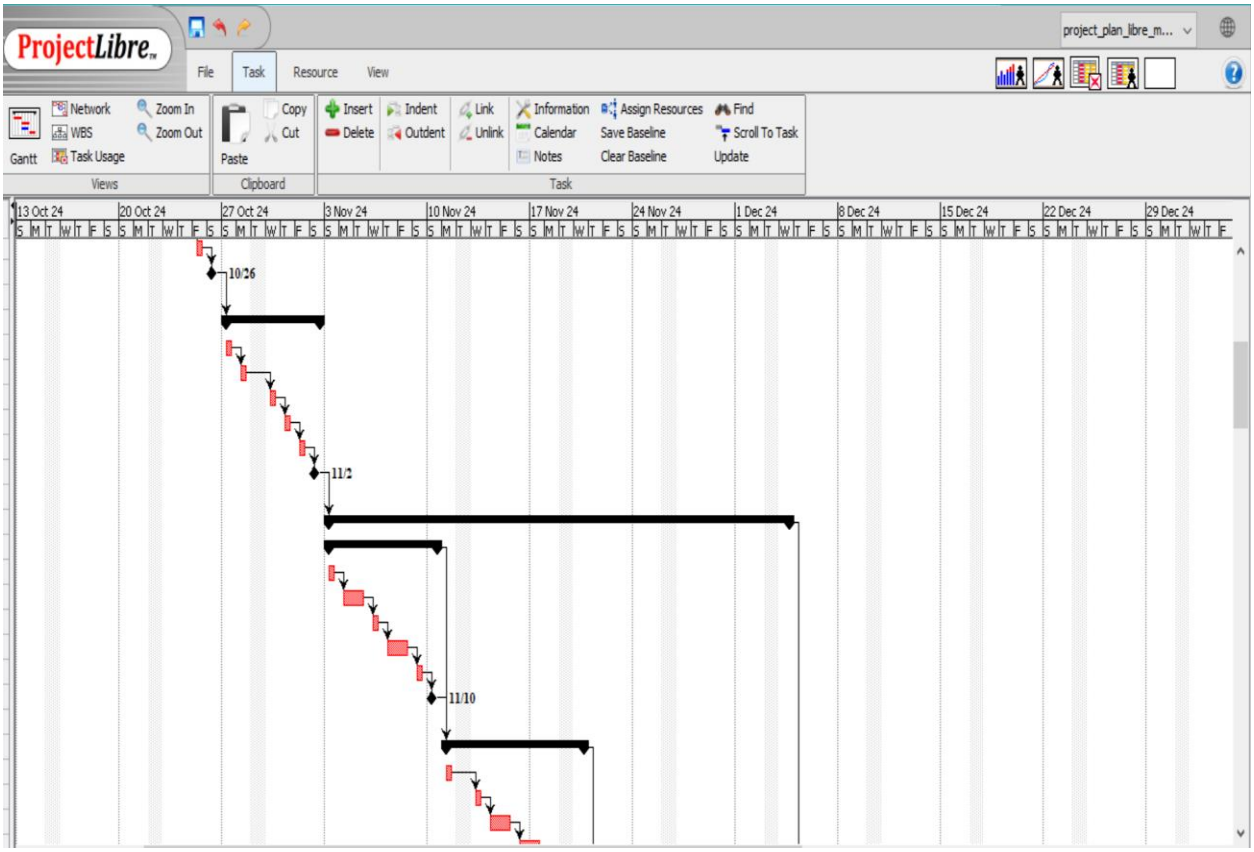
The manager of the Project will be responsible for communicating the status of Risks to the project **stakeholders**. It will include:

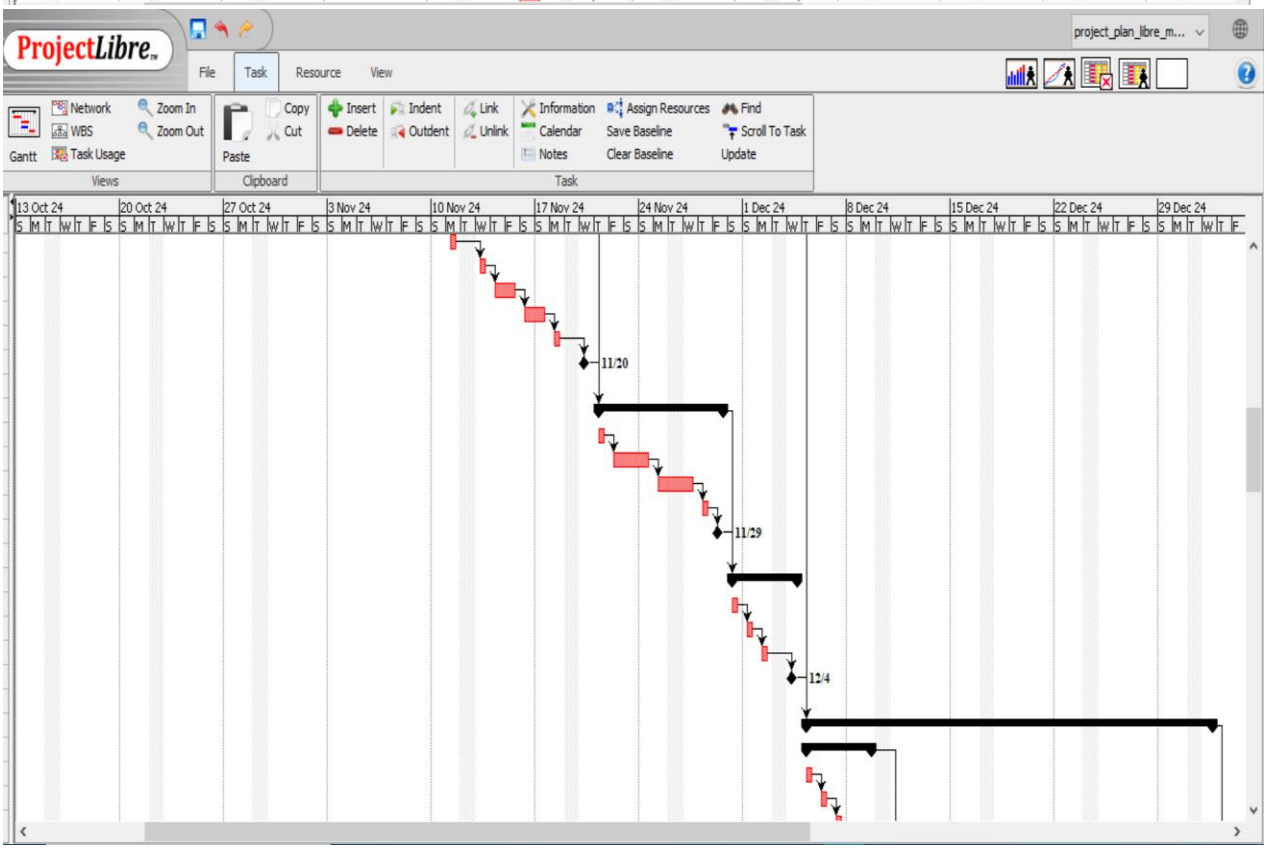
- ✓ Providing **Weekly Reports** during team and stakeholders meetings.
- ✓ Sharing **real-time updates** on critical risks via Project Communication channels (e.g., email address).

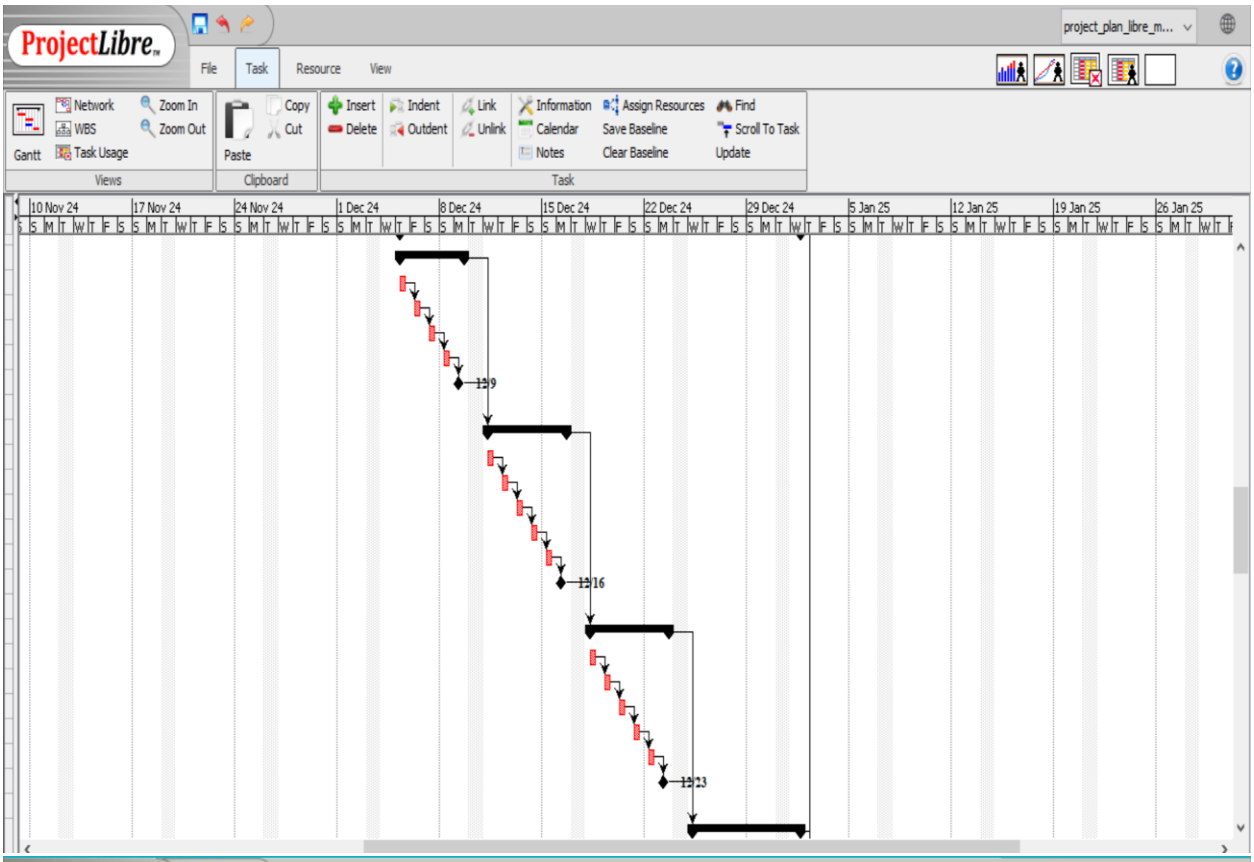
Project Schedule:

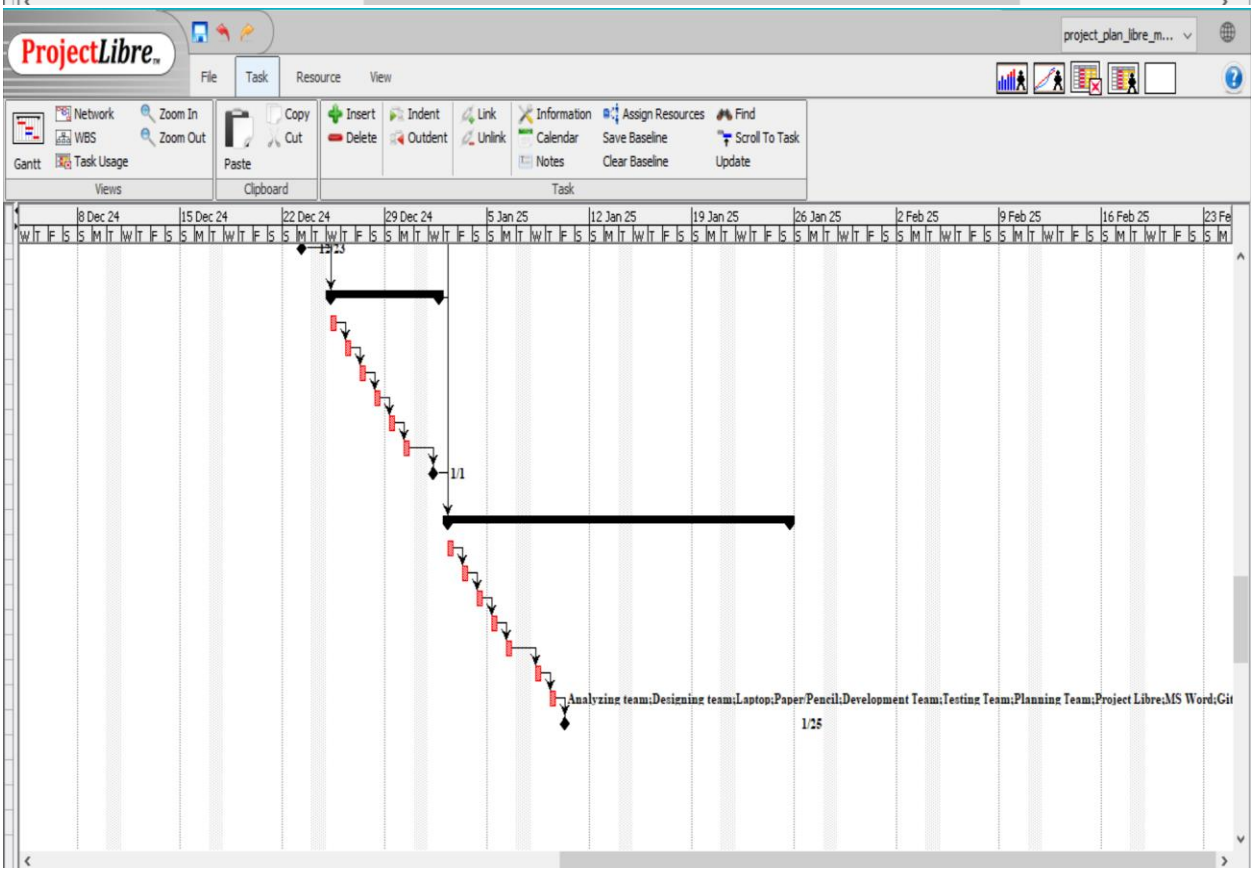
>Gantt Chart:



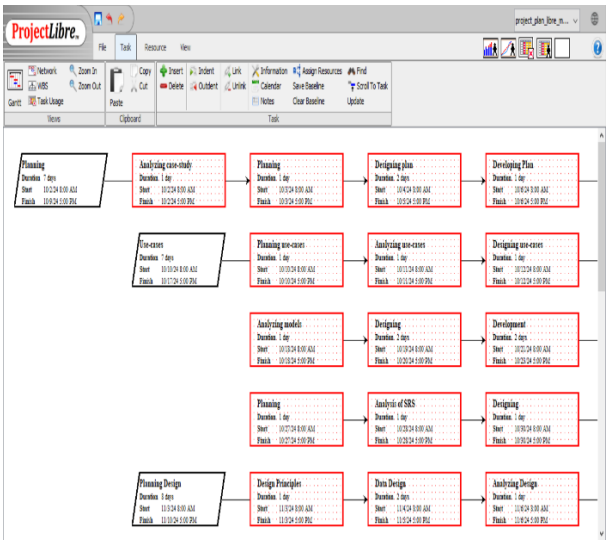

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Copy
Cut
Paste
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
13 Oct 24
20 Oct 24
27 Oct 24
3 Nov 24
10 Nov 24
17 Nov 24
24 Nov 24
1 Dec 24
8 Dec 24
15 Dec 24
22 Dec 24
29 Dec 24
10/26
11/2
11/10

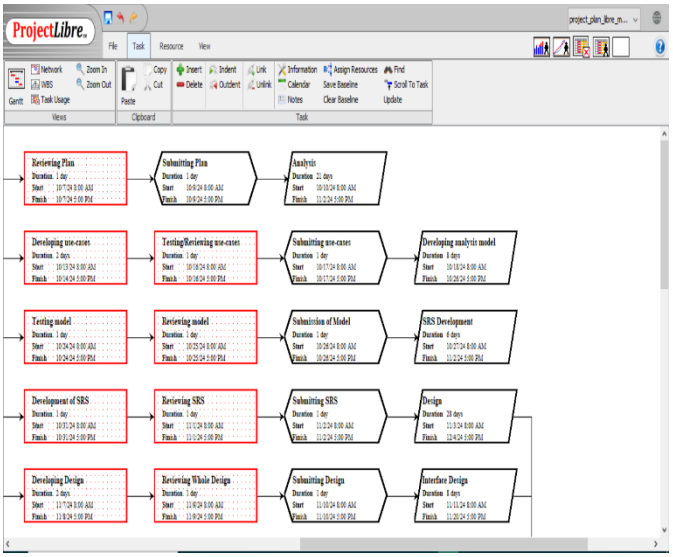

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Copy
Cut
Paste
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
13 Oct 24
20 Oct 24
27 Oct 24
3 Nov 24
10 Nov 24
17 Nov 24
24 Nov 24
1 Dec 24
8 Dec 24
15 Dec 24
22 Dec 24
29 Dec 24
11/20
11/29
12/4

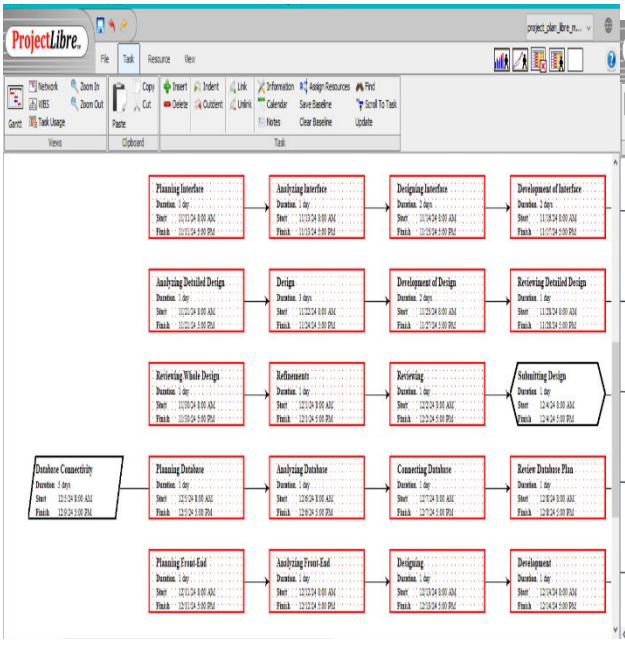

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Gantt
Views
Zoom In
Zoom Out
Paste
Copy
Cut
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
10 Nov 24
17 Nov 24
24 Nov 24
1 Dec 24
8 Dec 24
15 Dec 24
22 Dec 24
29 Dec 24
5 Jan 25
12 Jan 25
19 Jan 25
26 Jan 25

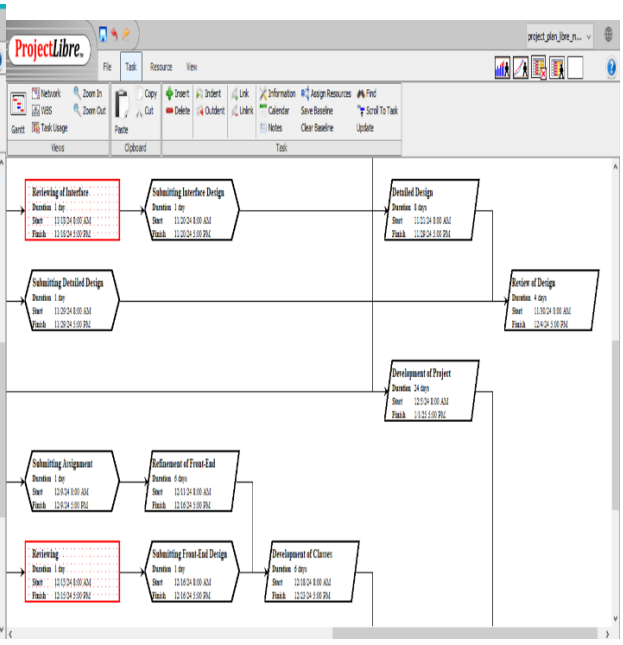

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Gantt
Views
Zoom In
Zoom Out
Paste
Copy
Cut
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
8 Dec 24
15 Dec 24
22 Dec 24
29 Dec 24
5 Jan 25
12 Jan 25
19 Jan 25
26 Jan 25
2 Feb 25
9 Feb 25
16 Feb 25
1/1
1/25

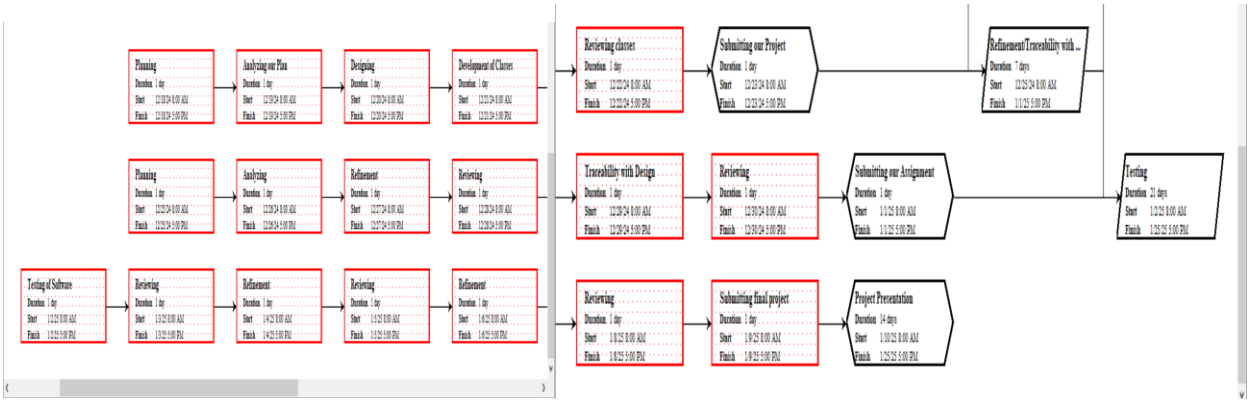
>Task Network:










Planning
Duration 1 day
Start 12/18/24 8:00 AM
Finish 12/18/24 5:00 PM
Analyzing our Plan
Duration 1 day
Start 12/19/24 8:00 AM
Finish 12/19/24 5:00 PM
Designing
Duration 1 day
Start 12/20/24 8:00 AM
Finish 12/20/24 5:00 PM
Development of Classes
Duration 1 day
Start 12/21/24 8:00 AM
Finish 12/21/24 5:00 PM
Reviewing classes
Duration 1 day
Start 12/22/24 8:00 AM
Finish 12/22/24 5:00 PM
Submitting our Project
Duration 1 day
Start 12/23/24 8:00 AM
Finish 12/23/24 5:00 PM
Refinement/Traceability with ...
Duration 7 days
Start 12/25/24 8:00 AM
Finish 1/1/25 5:00 PM
Planning
Duration 1 day
Start 12/20/24 8:00 AM
Finish 12/20/24 5:00 PM
Analyzing
Duration 1 day
Start 12/26/24 8:00 AM
Finish 12/26/24 5:00 PM
Refinement
Duration 1 day
Start 12/27/24 8:00 AM
Finish 12/27/24 5:00 PM
Reviewing
Duration 1 day
Start 12/28/24 8:00 AM
Finish 12/28/24 5:00 PM
Traceability with Design
Duration 1 day
Start 12/29/24 8:00 AM
Finish 12/29/24 5:00 PM
Reviewing
Duration 1 day
Start 12/30/24 8:00 AM
Finish 12/30/24 5:00 PM
Submitting our Assignment
Duration 1 day
Start 1/1/25 8:00 AM
Finish 1/1/25 5:00 PM
Testing
Duration 21 days
Start 1/2/25 8:00 AM
Finish 1/23/25 5:00 PM
Testing of Software
Duration 1 day
Start 1/2/25 8:00 AM
Finish 1/2/25 5:00 PM
Reviewing
Duration 1 day
Start 1/3/25 8:00 AM
Finish 1/3/25 5:00 PM
Refinement
Duration 1 day
Start 1/4/25 8:00 AM
Finish 1/4/25 5:00 PM
Reviewing
Duration 1 day
Start 1/5/25 8:00 AM
Finish 1/5/25 5:00 PM
Refinement
Duration 1 day
Start 1/6/25 8:00 AM
Finish 1/6/25 5:00 PM
Reviewing
Duration 1 day
Start 1/8/25 8:00 AM
Finish 1/8/25 5:00 PM
Submitting final project
Duration 1 day
Start 1/9/25 8:00 AM
Finish 1/9/25 5:00 PM
Project Presentation
Duration 14 days
Start 1/10/25 8:00 AM
Finish 1/29/25 5:00 PM
