



**American International University-Bangladesh (AIUB)**

**Department of Computer Science**

**Faculty of Science & Technology (FST)**

**AI-Powered Project Management Assistant**

A Software Engineering Project Submitted

By





Semester: Fall_24_25		Section:	Group Number:	
SN	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
1	Anindita Bhattacharjee	22-48606-3		
2	Fariar Rahman Arpa	22-49040-3		
3	Ehtesham Ferdous	23-50022-1		
4	Sharfin Ahmed Shanto	22-49561-3		
5	Sadman Hossain Niloy	22-49629-3		

## PROJECT PROPOSAL

### *1.1 Background to the Problem*

In office work, critical project management activities such as scheduling, task assignment, resource monitoring, and progress reporting continue to rely on manual methods although technology has advanced. Traditional project management methods are not only time-consuming but also prone to human error, oversight, and inefficiencies.

The need for efficient project management is growing as businesses and organizations operate in increasingly complex and dynamic environments. With the rise of remote and cross-functional teams, Artificial Intelligence (AI) presents a powerful solution to optimize project management. AI's ability to analyze large volumes of data, recognize patterns, process language, provide offline services, and make predictions can automate many manual tasks, including:

-  Scheduling
-  Task allocation
-  Resource tracking
-  Risk management




The root cause of inefficiency in project management lies in the overwhelming volume of tasks and data, which often leads to delays, missed deadlines, poor resource utilization, and reporting errors, which negatively impact project success. This problem is critical to address because poor project management can result in severe financial losses, operational disruptions, and reputational damage. Additionally, inefficiencies in managing time, tasks, and resources reduce overall team and organizational productivity.

### *1.2 Solution to the Problem*







An AI-based personal assistant for project management can revolutionize how projects are planned, executed, and monitored. AI plays a crucial role in automating tasks, analyzing data, and providing real-time insights to enhance project workflows and decision-making.

#### ***Project Objective***

The primary objective is to develop an AI-driven project management assistant that:





-  Reduces manual workload by automating repetitive tasks.
-  Enhance decision-making through predictive analytics and real-time insights.
-  Improve efficiency in scheduling, resource allocation and risk management.

### ***Key Features of the proposed Model***

-  Automates task creation, assignment and tracking.
-  Optimizes workforce and budget distribution using AI-driven insights.
-  Identifies potential risks and suggests mitigation strategies.
-  Adjusts timelines dynamically based on project progress.
-  Provides data-driven recommendations.
-  Uses machine learning to refine process based on historical data.

### ***Review of Existing Models***

Current project management tools offer basic automation but lack advanced AI capabilities such as:

-  Predictive analytics for risk management
-  Dynamic resource reallocation based on real-time data
-  Natural Language Processing (NLP) for voice-assisted task management
-  Adaptive learning to improve efficiency over time

This proposed AI-based assistant addresses these gaps by integrating machine learning, NLP and predictive modeling to deliver smarter, more autonomous project management solutions.

## **2. SOFTWARE DEVELOPMENT LIFE CYCLE**


### ***2.1 Process Model***

The development of the AI-based project management assistant adopts the Iterative SDLC to support flexibility and continuous enhancement. The process begins with Requirement Analysis to define user expectations and AI functionalities. This is followed by System Design, detailing architecture, AI model selection, and the tech stack. During Development & AI Integration, software components and intelligent models are built and trained. Finally, Deployment & Monitoring ensures a smooth rollout, ongoing AI updates, and performance tracking for long-term optimization.

#### ***Selected Process Model: Iterative***

After careful analysis, the Iterative process model has been chosen as the most suitable software development methodology for this project.

#### ***Justification for Selected Process Model: Iterative***

-  **Iterative Development:** Iterative enables short, time-boxed that support continuous updates—ideal for training and refining AI models.

- ✚ **Flexibility:** Requirements and priorities can evolve based on findings from AI experiments and changing user needs.
- ✚ **User-Centric Approach:** Frequent stakeholder involvement ensures the solution aligns with real-world expectations and project management workflows.
- ✚ **Early & Incremental Delivery:** Core features like task automation and NLP can be released in early delivery, providing immediate value and allowing early feedback.
- ✚ **Cross-Functional Collaboration:** Encourages seamless integration between AI developers, backend engineers, UI designers, and testers.
- ✚ **Risk Reduction:** Continuous testing and feedback help identify issues early, reducing long-term project risks.
- ✚ **Adaptability to AI Needs:** AI projects demand experimentation and retraining—Iterative adaptability supports this ongoing evolution.

### *Comparison with Other Models*

- ✚ **Waterfall:** Not suitable for AI development due to its rigid, non-iterative structure.
- ✚ **Spiral:** Supports risk management but adds unnecessary complexity for small AI modules.
- ✚ **Incremental:** Less adaptive than Agile in handling frequent model retraining or uncertain outcomes from AI experimentation.

## *2.2 Project Role Identification and Responsibilities*

### ✚ **AI/ML Engineer (Anindita)**

- Design and train AI models for task allocation, scheduling, and risk prediction.
- Evaluate model performance and fine-tune algorithms for accuracy and reliability.
- Implement NLP models for voice-based input and task interpretation.
- Integrate models into backend systems and support ongoing AI improvements.

### ✚ **Data Engineer (Fariar)**

- Build and maintain data pipelines to support AI training and predictions.
- Ensure data quality and integrity for consistent model performance.
- Optimize storage and retrieval processes for both structured (PostgreSQL) and unstructured (MongoDB) data.
- Provide real-time data support for dashboards and reporting tools.

### ✚ **Management (Sharfin)**

- Define project goals, timelines, and strategic alignment with business objectives.
- Approve major scope changes and resource allocation.
- Set standards and compliance policies for AI implementation.
- Monitor overall project progress and AI integration.
- Support change management and adoption of the AI assistant across teams.

### ✚ **Quality Assurance (QA) Engineer (Ehtesham)**

- Develop test strategies for functional, performance, and AI-specific validation.

- Automate test cases for features such as task assignment and reporting accuracy.
- Perform load testing to ensure system performance under high task volume.
- Report defects and work with developers to ensure high-quality sprint deliverables.

#### **UI/UX Designer (Sadman)**

- Design user-friendly interfaces for dashboards, task views, and voice inputs.
- Ensure that AI features (e.g., risk alerts, predictive insights) are clearly presented.
- Conduct usability testing to validate interface efficiency and user satisfaction.
- Maintain consistency in visual elements and user interaction patterns.

#### **Customer/User Representative**

- Provide insights on current pain points in manual project management processes.
- Validate backlog items and offer feedback on prototypes and delivered features.
- Test automation tools and predictive suggestions in real-world project settings.
- Ensure the AI assistant aligns with actual workflows and team expectations

### **3.1: Requirements**

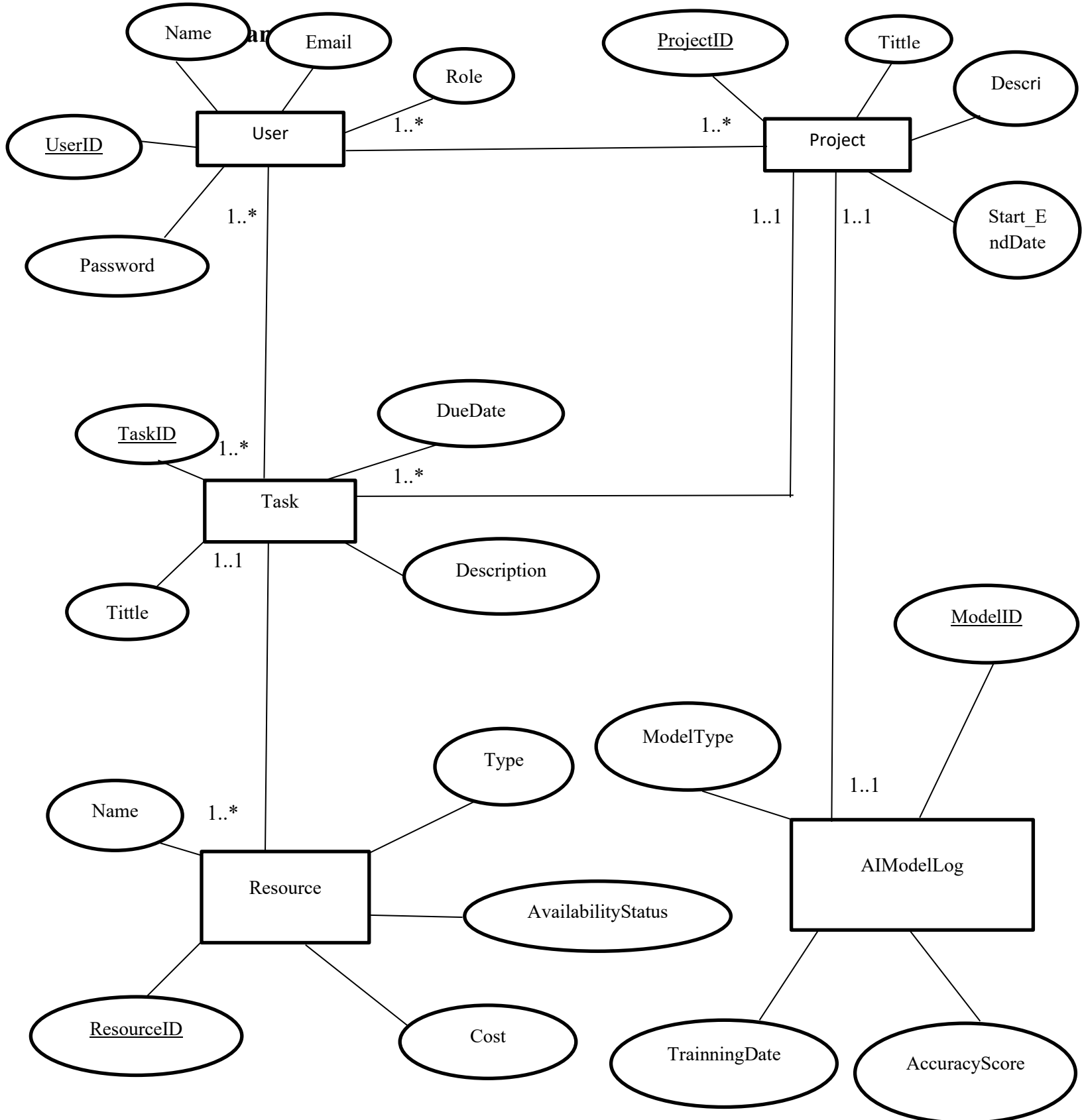
#### **Functional-Requirements**

- User Authentication and Authorization: Only authorized users can access the system.
  - ✓ 1.1. Valid user credentials, Login successful
  - ✓ 1.2. Invalid username or password, Login failed
  - ✓ 1.3. Empty fields, Show error message
  - ✓ 1.4. Unregistered username or password, Login failed
  - ✓ 1.5. More than 20 seconds to give details for login, Login failed
- Project and Task Management: Create, update, delete projects and tasks. Assign tasks to team members automatically or manually.
  - ✓ 2.1. Authorized user creates a new project, Project added successfully
  - ✓ 2.2. Project managers assign tasks to a team member manually, Assignment successful
  - ✓ 2.3. Unauthorized users try to delete a project, Access denied
- AI-Driven Task Assignment: Based on member availability, skill level, and deadlines.
  - ✓ 3.1. AI assigns task based on availability and skills, Assignment matches profile
  - ✓ 3.2. AI avoids assigning overburdened users, Workload balanced
- Resource Management: Manage people, budget, tools, everything. AI suggests reallocation when needed.
  - ✓ 4.1. Add team members, tools, and budget, Resources updated
  - ✓ 4.2. Assign resources to project, Linked correctly
- Risk Identification and Mitigation: Predict potential project risks (budget overruns, deadline slips) early using machine learning.
  - ✓ 5.1. AI predicts deadline slip, Warning issued
  - ✓ 5.2. Mitigation suggestions are generated, Suggestions shown in dashboard

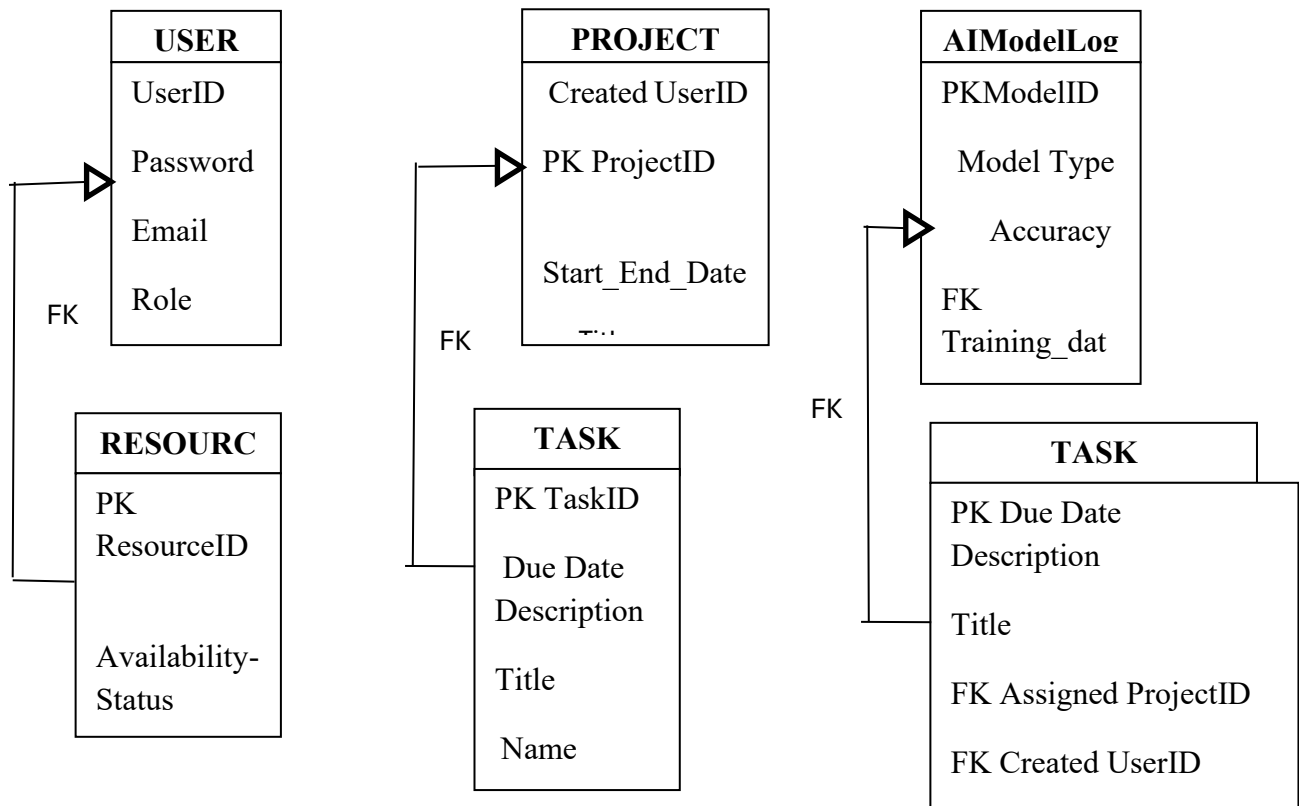
- Real-Time Notifications: Notify users of task updates, risks detected, deadline changes, or suggestions.
  - ✓ 6.1. Task status updated, Assigned user gets notification
  - ✓ 6.2. Risk not detected, System not giving alert
- Continuous learning Module: System learns from past project data to improve. Continuously retrains models using new data.
  - ✓ 8.1. AI model retrains after project closure, Training log recorded
  - ✓ 8.2. Accuracy improves over time, Prediction metrics show improvement
  - ✓ 8.3. New patterns are recognized, Assistant adapts responses
  - ✓ 8.4. System stores training history, History is accessible for auditing

#### **Non-Functional Requirements**

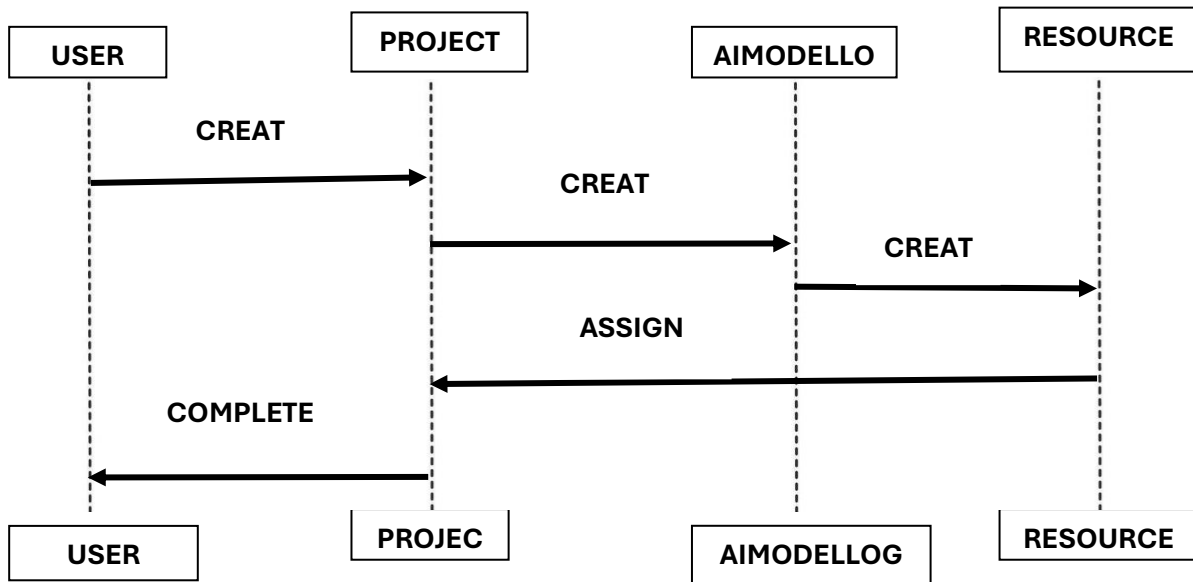
- Performance
- Reliability and Availability
- Security
- Usability
- Maintainability

**3.2: Diagrams**

## 2. Class Diagram

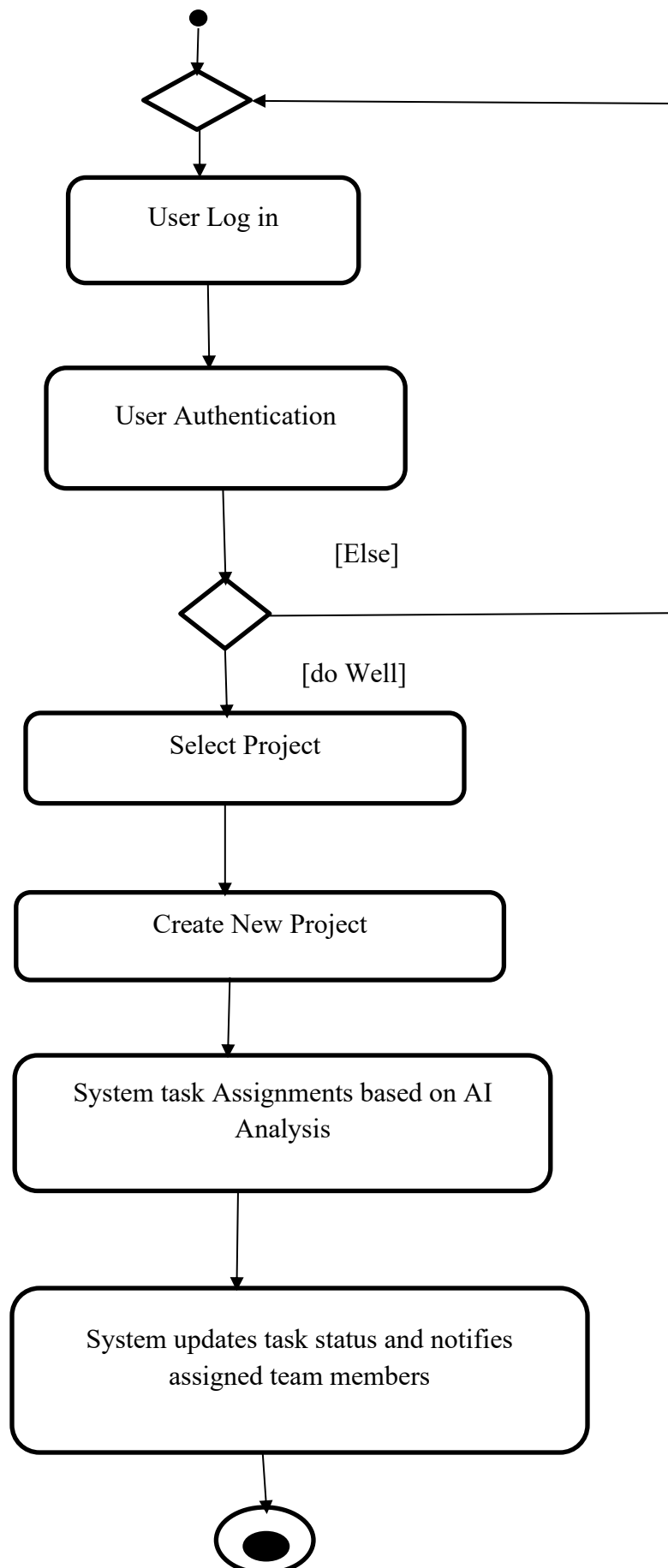


## 3. Sequence Diagram





#### 4. Activity Diagram



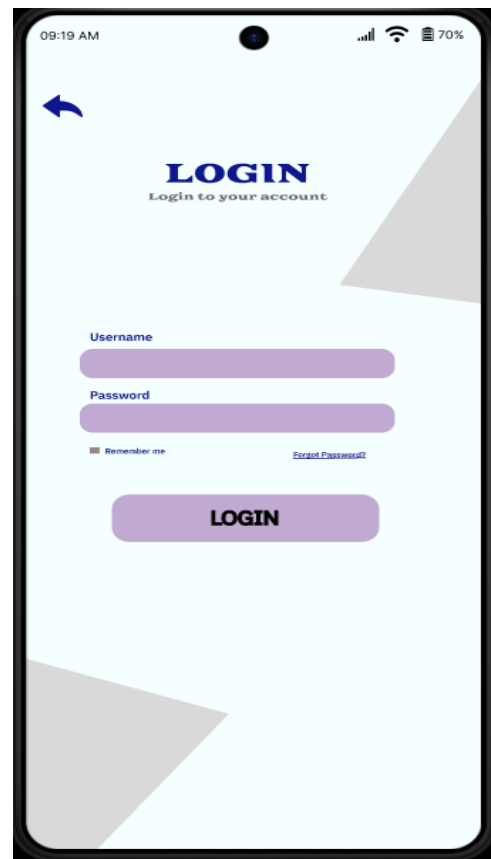
## ➤ *UI/UX Design:*

For the AI-Powered Project Management Assistant (AIPPMA), we propose designing a prototype for both mobile and web-based applications to ensure optimal experience for project management people. The design will follow UI/UX principles such as consistency, user control, and reducing memory load. Key considerations include:

### ❖ **Mobile Application (Android/iOS):**

**Layout and Design:** The interface will be designed for ease of use with simple navigation, ensuring representatives can login and visits, view their schedules, and access necessary information with minimal effort. Key functionalities like project requirements list, task allocation and chatbots.

**Usability:** To minimize the learning curve, intuitive icons, labels, and a clear color scheme will guide users through each task. Shortcuts and default settings will streamline common actions.





09:19 AM

←

## SIGNUP

Create an account to continue

Email

Username

Password

Confirm Password

Register



09:19 AM

←

## Project Requirement Lists

Task Management

Data Security & Confidentiality

Risk Management

Quality Standards

ChatBot



09:19 AM

←

## Task Allocation

Data Security

## Scheduling

10 June



09:19 AM

←

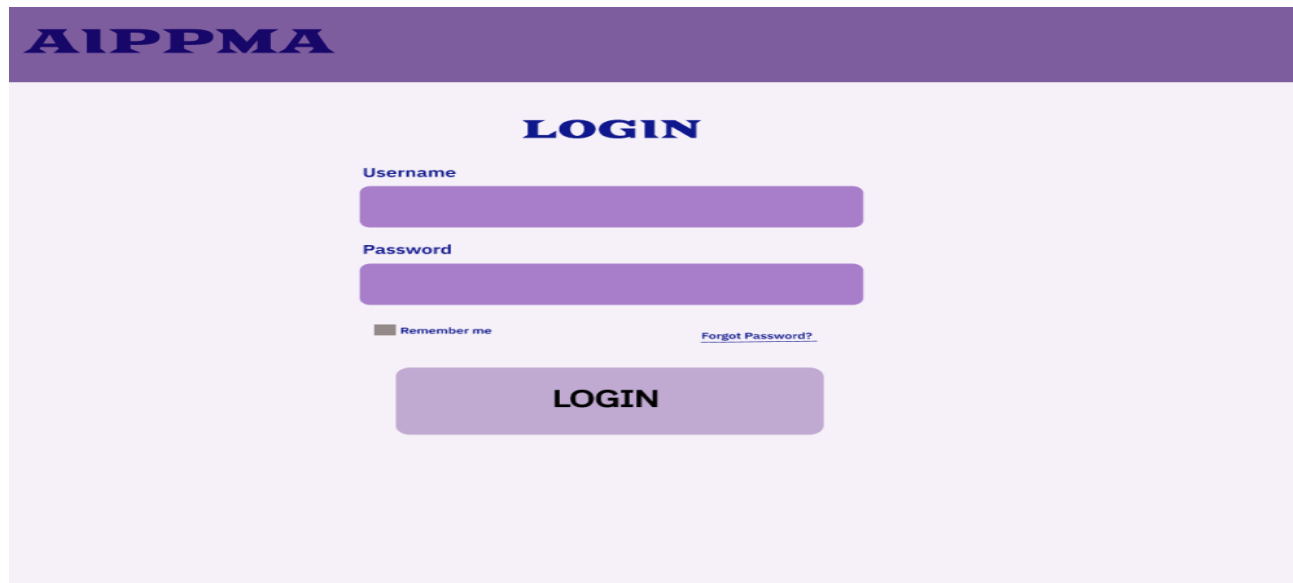
## ChatBot

Hello!  
How Can I help you?

### ❖ Web-Based Admin Panel:

**Dashboard Design:** Managers will be presented with a comprehensive dashboard displaying real-time data on representative activities, visit logs, and team lists.

**Interaction Flexibility:** Managers can quickly assign tasks, monitor performance, and communicate with representatives. The interface will also allow for multitasking, making interactions efficient and flexible.



The image shows a login form for AIPPMA. It features a purple header with the AIPPMA logo. Below the header, the word "LOGIN" is centered in a bold, dark blue font. The form includes two input fields: "Username" and "Password", both with purple borders and light purple backgrounds. Below the "Password" field, there is a "Remember me" checkbox and a "Forgot Password?" link. A large, light purple "LOGIN" button is centered at the bottom of the form.



The image shows a table titled "Work List" for AIPPMA. It features a purple header with the AIPPMA logo. Below the header, the title "Work List" is centered in a bold, dark blue font. To the left of the title is a blue left-pointing arrow, and to the right is a blue right-pointing arrow. The table has two columns: "Action" and "Code". It contains three rows of data, each with a light purple background.

Action	Code
Task Management	2332
Quality Management	1330
Data Security	1332

AIPPMMA		
 <b>Teams List</b>		
Code	Name	Role
2332	Ovi	Task Manager
1330	Sneha	Quality Assurance Manager
1332	Farhaan	Data Analyst

### ➤ *Project Testing*

We have selected functional requirements for testing in order to determine how the system should be tested. To do this, we have chosen the following requirements:

- Valid user credentials – Login should be successful.
- Invalid username or password – Login should fail.
- Empty fields – An error message should be displayed.
- Unregistered username or password – Login should fail.
- Taking more than 20 seconds to enter login details – Login should fail.
- Verifying that authorized users can successfully create a new project.
- Verifying that unauthorized users cannot delete a project.
- Verifying that the AI assigns tasks based on availability and skills.
- Verifying that no alert is triggered when no risk is detected.

### Test Case 1.1: Verify login with valid username and password

Project Name: AI Driven Project Management System		Test Designed by:		
Test Case ID:1.1		Test Designed date:		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Login Session		Test Execution date:		
Test Title	Verify login with valid username and password			
Description:	Test login functionality with valid credentials.			
Precondition (If any):	User has a valid registered account.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Enter valid username. 3. Enter valid password. 4. Click on the login button.	Username: user123  Password: pass123	<ul style="list-style-type: none"><li>Login page is displayed.</li><li>User is logged in and redirected to the dashboard.</li></ul>	As expected	Pass
Post Condition: User is validated and successfully logged into the application.				

### Test Case 1.2: Verify login fails with invalid credentials

Project Name:				
Test Case ID:				
Test Priority (Low, Medium, High): High				
Module Name:				
Test Title	Verify login fails with invalid username or password			
Description:	Test login functionality with invalid username or password.			
Precondition (If any):	User account exists in the system.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Enter invalid username. 3. Enter the invalid password. 4. . Click on the login button.	Username: invalid User Password: wrong Pass	<ul style="list-style-type: none"><li>Login fails and error messages are displayed.</li></ul>	As expected	Pass
Post Condition: User remains on the login page with error displayed.				

### Test Case 1.3: Verify error message when login fields are empty

Project Name:				
Test Case ID:				
Test Priority ((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify error messages when login fields are empty			
Description:	Test login functionality when both username and password fields are empty.			
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Navigate to the login page. 2.Leave username and password fields empty. 3. Click on the login button.	Username: blank Password:blank	Error message prompting to enter username and password is displayed.	As expected	Pass
Post Condition: No login is performed; users are prompted to enter the required credentials.				

### Test Case 1.4: Verify login fails for unregistered user

Project Name:				
Test Case ID:				
Test Priority ((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify login fails for unregistered user			
Description:	Test login functionality with an unregistered username.			
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Navigate to the login page. 2.Enter unregistered username. 3.Click on the login button.	Username: unknown User Password: any Password	<ul style="list-style-type: none"><li>Login page is displayed.</li><li>Login fails and 'User not found' error is displayed.</li></ul>	As expected	Pass
Post Condition: User remains on the login page with error displayed.				

### Test Case 1.5: Verify login timeout after 20 seconds

Project Name:				
Test Case ID:				
Test Priority((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify login timeout after 20 seconds			
Description:	Test that the login session times out if credentials are not entered within 20 seconds.			
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Navigate to the login page. 2.Wait for 25 seconds without entering credentials. 3.Enter valid username. 4.Enter valid password and click login.	Username: user123 Password: pass123	Session expires and login fails; error message is displayed.	Login page is displayed.	Pass
Post Condition: No session is created; users must start login process again.				

### Test Case 1.6: Verify authorized users can create a new project successfully

Project Name:				
Test Case ID:				
Test Priority((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify authorized users can create a new project successfully			
Description:	Test project creation functionality for an authorized user.			
Precondition (If any):	User is logged in with permission to create projects.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Login as an authorized user. 2. Navigate to the 'Create New Project' page. 3.Enter project details and submit the form 4.Confirm creation of the project.	Username: admin, Password: admin123	<ul style="list-style-type: none"><li>Dashboard is displayed.</li><li>New Project form is displayed.</li><li>Project is added successfully and appears in the project list.</li></ul>	As expected	Pass
Post Condition: New project exists in the project management system.				



### Test Case 1.7: Verify unauthorized user cannot delete project

Project Name:				
Test Case ID:				
Test Priority((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify unauthorized user cannot delete a project			
Description:	Test delete functionality for a user without delete permissions.			
Precondition (If any):	User is logged in without delete privileges.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Login as an unauthorized user. 2. Navigate to the project list page. 3. Attempt to delete a project. 4.Observe the system response.	Username: user1 Password: user1 pass	<ul style="list-style-type: none"><li>Dashboard is displayed.</li><li>List of projects is displayed.</li><li>Access denied message is displayed; project is not deleted.</li></ul>	As Expected	Pass
Post Condition: Project is not deleted; unauthorized user action is blocked				

### Test Case 1.8: Verify AI assigns tasks based on availability and skills

Project Name:				
Test Case ID:				
Test Priority((Low, Medium, High): Medium				
Module Name:				
Test Title	Verify AI assigns tasks based on availability and skills			
Description:	Test that AI assigns tasks to team members based on their skills and availability			
Precondition (If any):	Tasks and team member skill profiles are available in the system.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Login as a project manager 2.Navigate to the AI task assignment section. 3.Trigger the AI task assignment algorithm. 4.Verify the task assignments.	Username: manager, Password: manager123 Task: Test Login Feature (Testing); Task: Develop API Endpoint (Backend)	<ul style="list-style-type: none"><li>Dashboard is displayed.</li><li>AI Task Assignment page is displayed.</li><li>Each task is assigned to a team member with</li></ul>	As Expected	Pass

		matching skills and availability		
Post Condition: Tasks are correctly assigned in the system according to the AI algorithm.				

### Test Case 1.9: Verify no alert when no risk is detected

Project Name:				
Test Case ID:				
Test Priority((Low, Medium, High): Medium				
Module Name: Notification System				
Test Title	Verify no alert when no risk is detected			
Description:	Test that the notification system does not generate alerts when no risk conditions are present.			
Precondition (If any):	No high-risk tasks or issues are present in the system.			
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Login as a user 2.Navigate the project monitoring page. 3.Ensure there are no tasks flagged as high-risk. 4.Check no risk alert	Username: manager, Password: manager123 No alert is created since there are no detected risks.	<ul style="list-style-type: none"><li>• Dashboard is displayed.</li><li>• Project monitoring dashboard is displayed.</li><li>• No risk alerts or notifications are generated.</li></ul>	As expected	Pass
Post Condition: No alert is created since there are no detected risks.				

## ➤ Project Effort Estimation

### Effort Estimation Using COCOMO Model

To estimate the effort and timeline for our AI-based project management assistant, we applied the COCOMO (Constructive Cost Model) using the Semi-Detached mode due to the intermediate complexity and mixed experience levels in the team.

#### Assumptions:

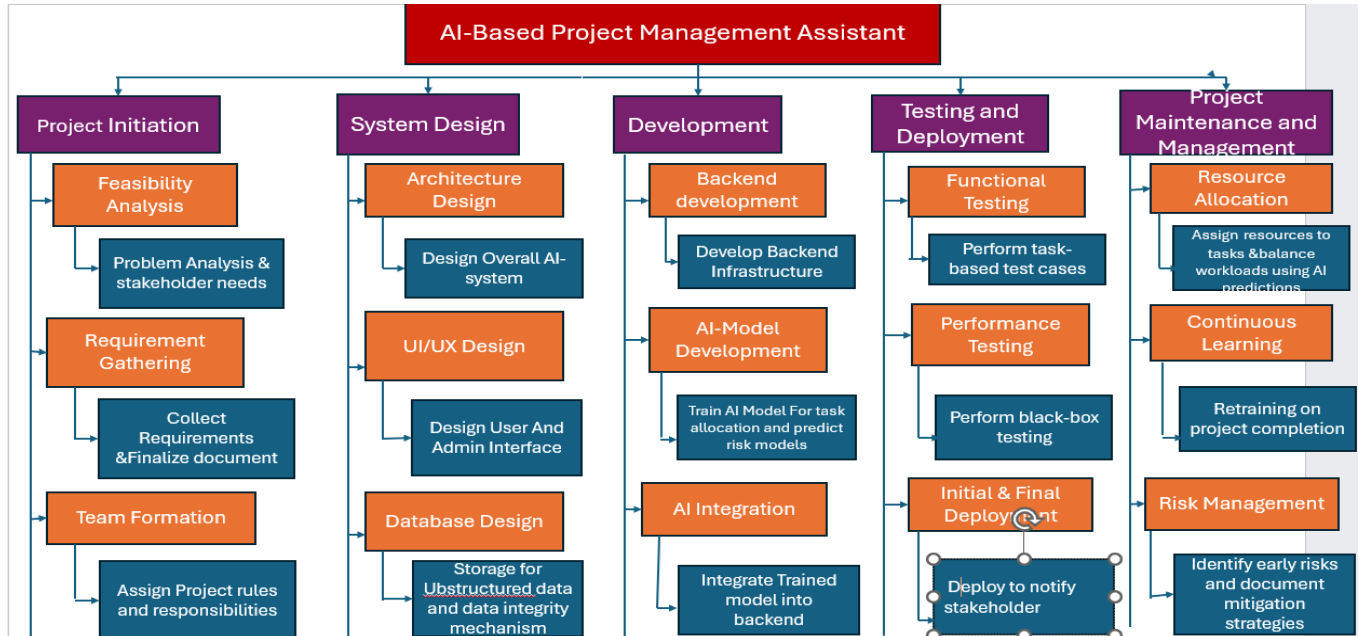
- Source Lines of Code (SLOC): 9827
- Project Type: Semi-Detached
- Coefficient (a): 3.0
- Complexity Exponent (P): 1.12
- Time Exponent (T): 0.35

#### Calculations:

- Effort (PM) =  $3.0 \times (9.827)^{1.12} = 38.78$
- Development Time (DM) =  $2.5 \times (38.78)^{0.35} = 8.9$
- Average Staffing (ST) =  $PM / DM = 38.78 / 8.99 = 4.31$  (5 people)

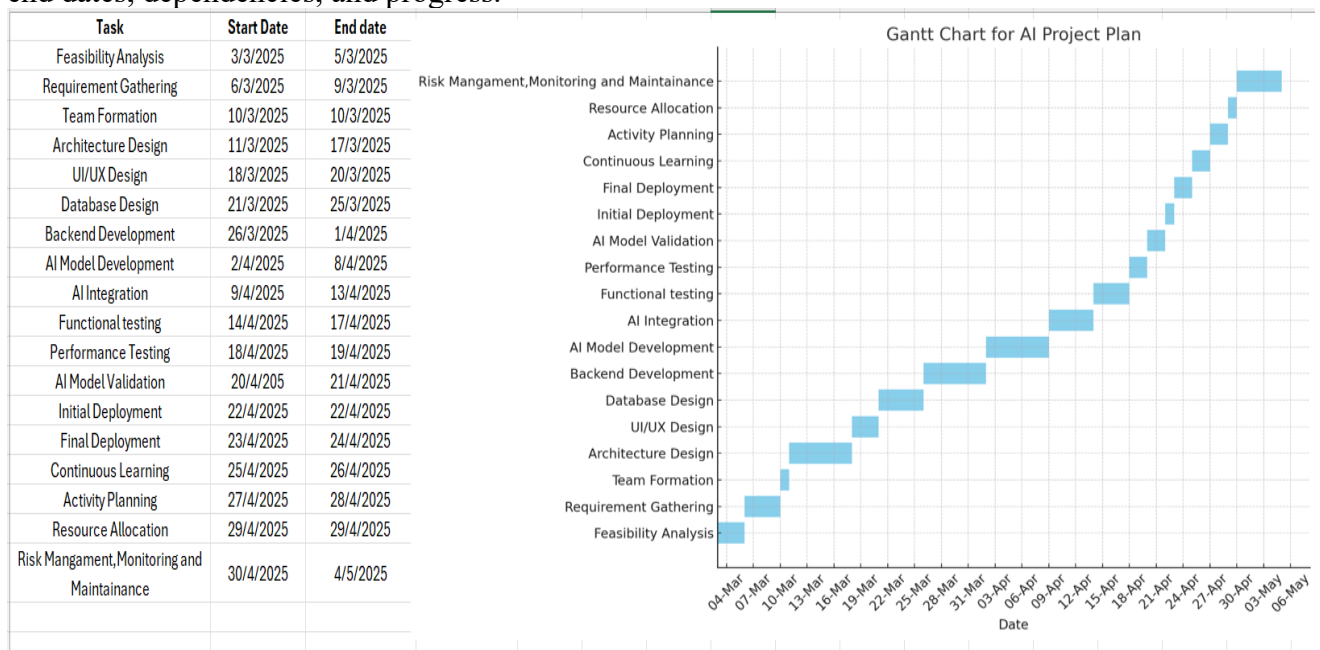
### ➤ *Project WBS Diagram:*

The WBS (Work Breakdown Structure) of project AI-Based Powered Project Assistant is organized into five phases where under phases three subphases are included and under subphases tasks are mentioned.



### ➤ *Project Gantt Chart*

The Gantt Chart visually represents the project schedule, outlining tasks, their durations, start and end dates, dependencies, and progress.



### ➤ *Project Risk Analysis*

Risk Analysis helps the team anticipate uncertainties, evaluate the likelihood and impact of each risk, and develop appropriate mitigation strategies to minimize negative outcomes.

ID	Risk Description	Impact	Risk Response	Risk level	Risk Owner	Notes
R01	Unauthorized access to sensitive project data	High	Implement secure authentication, role-based access control, and regular security monitoring.	High	Security Department	Regular audits, password policies, and security protocol updates are essential to prevent data breaches.
R02	AI model assigns tasks incorrectly due to poor data quality or incomplete profiles	High	Implement robust data validation, continuously update skill profiles, and retrain models with clean data.	High	AI/ML Engineer	Continuous data quality checks and profile updates are required to maintain AI accuracy in task assignments.
R03	NLP voice assistant misinterprets voice commands or natural language inputs	Low	Use state-of-the-art NLP models, include voice-to-text confirmation, allow manual override.	Average	NLP Developer/UX Design Team	Regular model tuning and user feedback collection can reduce misinterpretation of commands.
R04	AI fails to detect project risks accurately	High	Integrate ensemble learning methods and include manual override; allow human verification.	High	Risk Module Developer /Project Management Office	Accuracy must be monitored via test data, with human verification and override features always enabled.
R05	Project timeline delays due to iterative model retraining	High	Plan buffer time in sprints for AI experiments; parallelize non-AI tasks to maintain momentum.	Average	Project Management Office	AI updates should be scheduled separately from core feature deliveries to prevent delays.
R06	Poor user adoption due to complex UI or unclear features	Medium	Conduct frequent usability tests; gather user feedback early and iterate on UI/UX design.	Average	UX Designer Team/ Quality Assurance Department	Intuitive design and early onboarding help minimize training needs and improve user satisfaction.

R07	Inefficient resource allocation suggestions from AI in early versions	Medium	Introduce human-in-the-loop systems initially and fine-tune AI using actual usage data.	Average	AI/ML Engineer	Initial manual oversight of AI recommendations is necessary until the system learns effectively.
R08	Conflicts between manually assigned tasks and AI-suggested assignments	Low	Implement a conflict resolution system and provide visual comparison for manual review.	Low	Project Management office	Visual indicators and admin override are needed to resolve conflicting assignments transparently.
R09	Poor internet connectivity affecting cloud-based AI services	Medium	Include offline fallback options for core project management features; cache essential data locally.	Average	Backend Team	Essential functions should be accessible offline with auto sync once the connection is restored.
R10	Legal or ethical issues in using AI-based recommendations for human tasking	Low	Ensure explainability in AI decisions; include disclaimers and allow manager overrides.	Low	Project Management Office	Ensure all AI decisions are traceable and legally compliant; allow human-in-the-loop decisions.
R11	System downtime or crash during high-load usage	High	Perform load testing, introduce fault tolerance and automatic backup procedures	Average	DevOps Lead(Developer and IT Department Team)	Stress testing and scalable infrastructure are required to ensure uptime during critical usage.
R12	AI model fails to adapt well to evolving project data	Medium	Implement continuous learning and periodic model retraining with recent data	Average	AI/ML Engineer	Model retraining schedules should be automated and monitored to ensure system stays up-to-date.