Final Report on
IIT Kanban Board

Submitted to

SPL II Evaluation Committee 2022-2023
Bachelor of Science in Software Engineering
Institute of Information Technology
Noakhali Science and Technology University

Submitted By

Md.Raju Mia
ID:MUH2025016M
raju2515@student.nstu.edu.bd

Israt Jahan Jhumu
ID:BKH2025020F
jhumu2515@student.nstu.edu.bd

Md.Zahid Hasan
ID:ASH2025012M
zahid2515@student.nstu.edu.bd

Md.Saiful Islam
ID:MUH2025033M
saiful2515@student.nstu.edu.bd

Supervised By

Dipok Chandra Das
Assistant Professor
Institute of Information Technology
dipok.iit@nstu.edu.bd

Submission Date: 17th September, 2023

Table of Contents

List of Figures ..... Error! Bookmark not defined.
1. Project Description ..... Error! Bookmark not defined.
1.1 Introduction 1
1.1.1 Motivation 1
1.1.2 Aims and Objectives 1
1.2 Target Users 1
1.3 Requirements 2
1.4 Models, Tools and Resources 2
1.4.1 Model 2
1.4.2 Tools and Resources 3
1.5 Project Members 4
2. User Guide **5**
2.1 User end (Students, Facult Members) 5
3. Source Code Documentation ..... Error! Bookmark not defined.
3.1 Project Structure 5
3.1.1 signin.php ..... **Error! Bookmark not defined.**
3.1.2 home.php 6
3.1.3 AddProjectMebmerOption.php 7
3.1.4 TaskBoard.php 9
3.1.5 TaskCreateForm.php 10
3.1.6 JoinYourProject.php 11
3.2 Lines of Code 12
4. SRS & Development Mapping **13**
5. Challenges and Future Work **13**
Appendix A: SRS Document **27**
Functional Requirement 27
Non-Functional Requirement 31
Usability Requirement 34
Security Requirement 34
Performance Requirement 38
Technical Requirement 39
Use Case Descriptions 41
Appendix B: Project Proposal **84**
Introduction 84
Motivation 84
Objectives 85
Target Customers 85
Model 85
Tools and Resources 85
Challenges 86
Appendix C: Database Schema **87**

List of Figures

Figure 01 Evalutionary model 3
Figure 02 Front page 15
Figure 03 Sign up 15
Figure 04 Place Informatin 16
Figure 05 Email verification system with code 16
Figure 06 Verificatin Successful mesage 17
Figure 07 Sign in into the system 17
Figure 08 Home page of Iit kanban Board 18
Figure 09 Create new project form 18
Figure 10 Place all informatinr 19
Figure 11 Invite member and Successful message 19
Figure 12 Add member and place email 20
Figure 13 Invitation Successful message 20
Figure 14 Place email for add member 21
Figure 15 Different Project show into the home page 21
Figure 16 Task details under list 22
Figure 17 Place information into the task details 22
Figure 18 Task create message 23
Figure 19 Place information with priority 23
Figure 20 Another task created 24
Figure 21 Task details with share documents,comments and delete task 24
Figure 22 Upload file into the task 25
Figure 23 Select file from the drive 25
Figure 24 Download file 26
Figure 25 One user involves various types of project 26
Figure 26 Use Case_Diagram 40
Figure 27 Database Schema 87
Figure 28 ER Diagram 88

List of Tables

Table 01 List of requirements 2
Table 02 List of tools 3
Table 03 List of members 4
Table 04 List of lines of code 12
Table 05 List of functions 13
Table 06 List of use cases 13
Table 07 Enter the System 41
Table 08 Invite Member 42
Table 09 Create Project Folder 44
Table 10 Delete Project Folder 46
Table 11 Crate List 48
Table 12 To Do 50
Table 13 Add Task 51
Table 14 Delete Task 53
Table 15 Assign Member 55
Table 16 Set Priority 57
Table 17 Set Due Date 59
Table 18 Add Desciption 60
Table 19 Crate Board 62
Table 20 Add Status 64
Table 21 Add Comments 66
Table 22 Share Documents 67
Table 23 Download Documents 69
Table 24 Messaging 71
Table 25 Get Notifications 72
Table 26 Notify Update Progress 74
Table 27 Control Privacy 75
Table 28 Control Updating 76
Table 29 Manage Date/Time 78
Table 30 Manage Notification 79
Table 31 Generate Gantt Chart 80
Table 32 Identity Update 82
Table 33 Generate Report 83

1. Project Description

1.1 Introduction

The introduction of software requirement specification provides an in-depth document that describes what the software does, how will it be performed and includes information about all the functional and non-functional requirements.

Welcome to the groundbreaking IIT Kanban Board Software Project, where innovation meets efficiency in project management and collaboration. In this digital era, we recognize the need to optimize the way teachers and students handle their project work within the IIT community. Our purpose is clear to the existing manual processes and elevate productivity to new heights.

1.1.1 Motivation

In IIT we have done each of our SPL through analog process in previous time. There is no well furnished technological platform to understand the improvement of our own work. But in recent time there are numerous developed software for overlooking the improvement of our work, communicating with each others and sharing documents etc. But these resources are not free of cost. So we want to develop a project that aims for enriching our technological knowledge and leads IIT to forward.

1.1.2 Aims and Objectives

The initial aim and objectives of our system were to provide these features,

1. **Digital Project Management:** Create a digital platform that allows teachers and students to efficiently plan, manage, and monitor their academic and research projects.
2. **Improved Collaboration:** Implement features for inviting project members via Gmail, messaging, and document sharing to enhance communication and information exchange.Fine Calculation: The system calculates the fine due for non-return or lost and damaged books. The members will be notified about the fines by the system.
3. **Efficient Task Management:** Enable users to create project folders, task lists, and individual tasks, making it easier to organize and manage project-related activities.
4. **Progress Tracking:** Enable users to track project progress, set priorities, and meet project milestones effectively.

1.2 Target Users

- All Teachers and Students – IIT, NSTU

1.3 Requirements

Table 01 List of requirements

No.	Requirement	Completed
FR-1	User registration/signup.	YES
FR-2	Project invitation	YES
FR-3	Create a Project Folder**.**	YES
FR-4	Delete a Project Folder.	YES
FR-5	Create List.	YES
FR-6	To Do.	YES
FR-7	Add Task.	YES
FR-8	Delete Task.	YES
FR-9	Assign Member.	YES
FR-10	Share Docs.	YES
FR-11	Commenting on tasks/documents	YES
FR-12	Get Notifications.	Not Done Yet
FR-13	Reporting and analytics.	YES
UR-1	Simple and Intuitive navigation.	YES
UR-2	Clear and consistent visual design.	YES
UR-3	Accessible for all.	YES
UR-4	Error Prevention and Recovery.	YES
TR-1	Web Platform	YES
TR-2	Database	YES
TR-3	User Authentication	YES
TR-4	Responsive Design	YES

1.4 Models, Tools and Resources

1.4.1 Model

SPL I has had a significant influence on our project. As we iteratively developed our project using the methodology we had chosen, we encountered a variety of issues. We want to implement something new this time, and after learning from the project from last year, we have decided to

use the agile model to do it. Break tasks into smaller iterations using the agile model. The project risk is reduced and the overall project delivery time requirements are reduced due to the project's breakdown into smaller components.

Figure-01: Evolutionary Model

1.4.2 Tools and Resources

Table 02 List of tools

Category	Name or Description
Text Editor	Visual Studio Code
IDE	PhpStorm 2022 2.1
Server	Xampp
RDBMS	MySQL
Language	HTML, CSS, JavaScript, PHP, SQL

Learning Resource	1. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages 2nd Edition by Elisabeth Robson 2. Head First JavaScript Programming: A Brain-Friendly Guide 1st Edition by Eric Freeman 3. PHP & MySQL: Server-side Web Development by Jon Duckett 4. https://www.w3schools.com/ 5. https://getbootstrap.com/ 6. https://www.udemy.com/
Booking resources	1. Software Engineering 9th Edition by Lan Sommerville 2. Requirements Engineering Fundamentals by Klaus Pohl 3. 3. Database System Concepts 6th Edition by Abraham Silberschatz
Online resources	1. https://github.com 2. https://stackoverflow.com/ 3. https://getbootstrap.com/

1.5 Project Members

Table 03 List of members

Team Members	Supervisor
Md.Raju Mia (MUH2025016M) Israt Jahan Jhumu (BKH2025020F Md.Zahid Hasan (ASH2025012M) Md.Saiful Islam (MUH2025033M)	Dipok Chandra Das Assistant Professor Institute of Information Technology

2. User Guide

2.1 User end (Students, Faculty Members)

3. Source Code Documentation

3.1 Project Structure

3.1.1 signinaction.php

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

filter_var()

Description: Validates an email address format.
Params: $email (string)
Return Value: boolean (true if valid, false if not)

Database Connection (MySQL):
Description: Establishes a connection to a MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

$conn->connect_error

Description: Checks if there is a connection error.
Params: None
Return Value: boolean (true if there's an error, false if not)

SQL Query Execution:

Description: Executes SQL queries to check user existence and verify the password.
Params: None
Return Value: MySQLi result object

password_verify()

Description: Compares a plain text password with a hashed password.
Params: $password2 (string), $stored_password (string)
Return Value: boolean (true if they match, false if not)

Session Variable Setting:

Description: Sets session variables for user data upon successful login.
Params: $_SESSION['signinemail'] (string), $_SESSION['Sign_In_Email_Id'] (int), $_SESSION['Sign_In_Email_User_Name'] (string)

Return Value: None

header("Location: ...")
Description: Redirects the user to another page.
Params: Location URL (string)
Return Value: None

$conn->close()

Description: Closes the MySQL database connection.
Params: None
Return Value: None

3.1.2 home.php

Code Description:

This PHP script generates a user dashboard page with project listings and various navigation options.

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

HTML and CSS:

Description: Defines the structure and styling of the dashboard using HTML and CSS.
Params: None
Return Value: None

JavaScript (Typewriter Effect):

Description: Creates a typewriter effect for displaying a welcome message.
Params: None
Return Value: None

JavaScript (Dropdown Menu Handling):

Description: Handles the behavior of dropdown menus in the navigation section.
Params: None
Return Value: None

JavaScript (Confirm Delete):

Description: Displays a confirmation dialog for deleting a project.
Params: projectId (int), supervisorId (int), projectName (string)
Return Value: None

Conditional Check (isset($_SESSION['signinemail'])):

Description: Checks if the user is signed in by verifying the existence of the 'signinemail' session variable.
Params: None
Return Value: None

Database Connection:

Description: Establishes a connection to the MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

SQL Query Execution:

Description: Executes SQL queries to retrieve project information.
Params: None
Return Value: MySQLi result object

Project Listing:

Description: Displays a list of user projects with options to invite members and delete projects.
Params: None
Return Value: None

header("Location: signin.php") (Redirect):

Description: Redirects the user to the sign-in page if not authenticated.
Params: Location URL (string)
Return Value: None

3.1.3 AddProjectMemberAction.php

Code Description:

This PHP script handles the invitation of members to join a project and sends email notifications.

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

if ($_SERVER["REQUEST_METHOD"] == "POST") {

Description: Checks if the HTTP request method is POST.
Params: None
Return Value: None

Database Connection (MySQL):

Description: Establishes a connection to the MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

Retrieving Session Variables:

Description: Retrieves session variables like 'signinemail', 'project_Id', 'user_email', and 'project_name'.
Params: None
Return Value: None

Email Invitation Validation:

Description: Checks if the invited email is the same as the supervisor's email and whether the email already has membership in the project.
Params: None
Return Value: None

SQL Query Execution (INSERT):

Description: Inserts a new project member into the database with a "NO" join status.
Params: None
Return Value: None

Sending Email Invitation:

Description: Sends an email invitation to the specified member.
Params: $email (string), $subject (string), $message (string), $headers (string)
Return Value: None

Session Variable Setting (Success Messages):

Description: Sets session variables for success messages.
Params: $_SESSION['success'] (string)
Return Value: None
header("Location: ...") (Redirect):

Description: Redirects the user back to the 'AddProjectMember.php' page with success or error messages.
Params: Location URL (string)
Return Value: None

$conn->close();

Description: Closes the MySQL database connection.
Params: None
Return Value: None

3.1.4 TaskBoard.php

Code Description:

This is an HTML and PHP script for displaying task details for a project.

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

HTML Header and Styles:

Description: Defines HTML structure, styles, and headers for the web page.
Params: None
Return Value: None

Database Connection (MySQL):

Description: Establishes a connection to the MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

Retrieving Session Variables:

Description: Retrieves session variables like 'signinemail', 'project_Id', 'user_email', and 'project_name'.
Params: None
Return Value: None

Displaying Task Details:

Description: Fetches and displays task details for the project.
Params: None
Return Value: None

HTML Buttons and Links:

Description: Provides buttons and links for creating tasks, updating task statuses, and performing actions like comments and file uploads.
Params: None
Return Value: None

JavaScript Function (closeAlert()):

Description: Defines a JavaScript function to close alert messages.
Params: None
Return Value: None

header("Location: ...") (Redirect):

Description: Redirects the user to the 'signin.php' page if they are not logged in.

Params: Location URL (string)
Return Value: None

3.1.5 TaskCreateForm.php

Code Description:

This is an HTML and PHP script for creating a task within a project.

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

HTML Header and Styles:

Description: Defines HTML structure, styles, and headers for the web page.
Params: None
Return Value: None

Database Connection (MySQL):

Description: Establishes a connection to the MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

Retrieving Session Variables:

Description: Retrieves session variables like 'signinemail', 'project_name', 'project_Id', and 'Task_Id'.
Params: None
Return Value: None

Task Creation Form:

Description: Displays a form for creating a task, including fields for task title, description, assigned member, due date, and priority.
Params: None
Return Value: None

Form Submission (TaskCreateFormAction.php):

Description: Handles form submission and sends data to "TaskCreateFormAction.php" for processing.
Params: POST data (task_title, task_description, Assign_Member, due_date, priority)
Return Value: None

JavaScript Function (closeAlert()):

Description: Defines a JavaScript function to close alert messages.
Params: None
Return Value: None

header("Location: ...") (Redirect):

Description: Redirects the user to the 'signin.php' page if they are not logged in.
Params: Location URL (string)
Return Value: None

.

3.1.6 JoinYourProject.php

Code Description:

This is an HTML and PHP script for joining projects and displaying project lists.

session_start()

Description: Initializes or resumes a session.
Params: None
Return Value: None

HTML Header and Styles:

Description: Defines HTML structure, styles, and headers for the web page.
Params: None
Return Value: None

Database Connection (MySQL):

Description: Establishes a connection to the MySQL database.
Params: $servername (string), $username (string), $password (string), $dbname (string)
Return Value: MySQLi connection object

Handling Session Variables:
Description: Retrieves session variables like 'error', 'signinemail', and 'Sign_In_Email_Id'.
Params: None
Return Value: None

Displaying Error Alert:
Description: Displays an error alert if the 'error' session variable is set.
Params: None
Return Value: None

Join Project Form:

Description: Displays a form for users to input a Join ID and join a project.
Params: None
Return Value: None

JavaScript Function (confirmDelete()):

Description: Confirms project deletion and redirects to "DeleteProjectJoin.php" if confirmed.
Params: projectId (integer), supervisorId (string), projectName (string)
Return Value: None

JavaScript Function (closeAlert()):

Description: Defines a JavaScript function to close alert messages.
Params: None
Return Value: None

Project List Display:

Description: Displays a list of projects the user is either supervising or a member of.
Params: None
Return Value: None

3.2 Lines of Code

Code Level

LOC : Line of code in the whole project. The Total number of code lines in whole project.
NCLOC : Non-comment line of code. The line which are not commented and this code accomplish our objective.
CLOC : Comment line of code. The line of code which are not working to calculate output.
ALOC : The average number of lines of code in each file.
DC : It means the number of comment line proportion to average code.

Table 04 List of lines of code

LOC	10112
NCLOC	9302
CLOC	810
Average LOC	259
Density of Comments	4%

4. SRS & Development Mapping

Table 06 List of use cases

Use Case No	Use Case	Implementation
UC 01	Enter the System	Index.html
UC 02	Invite Members	JoinYourProject.php
UC 03	Create a Project	ProjectCreateFormM.php
UC 04	Delete the Project	DeleteProject.php
UC 05	Create List	TaskCreateForm.php
UC 06	To Do	TaskBoard.php
UC 07	Craete Task	TaskBoard.php
UC 08	Delete Task	TaskBoard.php
UC 09	Assign Member	Tsk crete form
UC 10	Set Priority	Taskboard.php
UC 11	Set Due Date	TaskCreateForm.php
UC 12	Add Description	TaskCreateForm.php
UC 13	Crate Board	TaskCreateForm.php
UC 14	Add Status	TaskCreateForm.php
UC 15	Share Documents	upload.php
UC 16	Download Documents	download.php
UC 17	Get Notification	
UC 18	Notify Update Progress	
UC 19	Messaging	comments.php
UC 20	Control Privacy	Logout.php
UC 21	Control Updating	Status_Update.php
UC 22	Manage Date/Time	TaskBoard.php
UC 23	Manage Notification	Notification.php
UC 24	Generate Gantt Chart	Gantt-Chart.php
UC 25	Report Generating	Report.php

5. Challenges and Future Work

The main challenges we faced during developing this system are,

1. The software must be designed with a focus on usability and a positive user experience. Complex or unintuitive interfaces could hinder adoption and efficiency.
2. Gathering feedback from project members and continuously improving the software based on their input will be crucial for its long-term success and relevance.

3. As the number of users and projects grows, the system must be scalable to handle increased data and user load without performance degradation.

The future work we are proposing,

1. Consider integrating the Kanban board software with other existing academic systems or tools used within the IIT system for a more seamless experience.
2. Enhance reporting capabilities to provide detailed insights into project performance, allowing for data-driven decision-making.
3. Develop a mobile application version of the software to enable project management and collaboration on smartphones and tablets.

Some Figure of IIT Kanban Board

Front Page

This is the opening page of IIT Kanban Board.User(Teachers/students) can Sign Up and Sign In into the system.A user must have an account to entire the system.It is the one of the mandatory part in our system.

Figure-02

Sign Up

If a user want to entire into our system,he/she have to create account in our system with place user's Email address,User_Name and Password and finally he/she click into the Create Account.It is the first condition to entire the system.

Figure-03

Place Information

This is the Page of Place information about Sign-In of IIT-Kanban-Board. In this segment User can easily Sign In their Account to given their Email-Address and Random Password

Figure-04

Email verification system with verification code.

This is the page of Email Verification System with Verification code of IIT-Kanban-Board. In this segment User can easily Verify their Account with Verification code that should be provide their Email Address

Figure-05

Verification Successful message

This page refers to Verification Successful message about the User that should be already verified his/her Account Successfully

Figure-06

Sign in into the system

This page refers to User to Sign in into his/her Account that should be previously created. In this segment User can easily enter their Account with Email and passcode.

Figure-07

Home page of IIT Kanban Board

This is the Home-Page of IIT-Kanban-Board. When User has Email Address and also password that should be already placed and verified their account then this page shown. It should be defined User project, Create-Project, Add-member and also other things.

Figure-08

Create New Project Form as A Supervisor

This is the another page of Create Project Form as a Supervisor . In this segment User can Create Project as a Supervisor with Project Name and Project Description

Figure-09

Create New project as A Member

This page Refers to Create New Project as a Member. In this segment User can Create Project as a Member with Project Name ,Project Description And also a Supervisor Email Address

Figure-10

Invite member and Successful Message

This page refers to Invite Member remaining in as a supervisor. In this segment Supervisor can easily invite a member and when she/he invite a member to see invitation successful message.

Figure-11

Add member and place email

This page refers to Add Member and Place Email. In this segment User can easily add in a project that should be created by supervisor with an Email Address

Figure-12

Invitation Successful message

This page refers to Invitation Successful message when executing to add- in a member in the Project .

Figure-13

Place Email for add member

This page refers to Place Email for add member . In this segment User can easily add a member with their Email Address in a existing project

Figure-14

Different project show into the Home page

This page refers to Existing Project in a User already created or joined. In this segment User can easily to see their existing Project that should be already assigned.

Figure-15

Task Details Under List

This Page refers to Create task that should be assigned to a Project member including task Title, Assign member , Due date , and also Set priority.

Figure-16

Place Information into the task Details

This is the page of place information of IIT Kanban Board about Crated task details information. In this segment Task name, Assign member Email Address, Due Date and also a Member Priority

Figure-17

Task Create Message

This page refers to when a User create a task then confirmation message shown Task Created

Figure-18

Place information with priority

This page refers to Place information with Priority about a Member that should be already assigned with task name, email address, due date and set priority.

Figure-19

Another Task created

This page refers to multiple task with an Specific supervisor. In this segment Supervisor create another task with Task name, Assign member, Due Date and also set Priority.

Figure-20

Task Details with Share Documents, Comments, Delete and Task

This page refers to Task Details with Share Documents, Comments, Delete . In this segment Supervisor can easily Share Documents, Comments ,Delete this specific Task.

Figure-21

Uploaded File into the task

This page refers to Upload a file remaining into a specific Task. In this segment Supervisor,User can easily upload a specific file with pecific task.

Figure-22

Select file from the Drive

This page refers to Select file from the Drive with a Specific task. In this segment Supervisor/User member can easily Select a file from the drive and attach it.

Figure-23

Also Download the file

This page refers to Download The File with a specific Task that should be already created. In this segment Supervisor/User can easily Download Uploaded Files with specific format.

Figure-24

One user involves with various types of project

This page refers to A specific users involves with various types of Project that should be already created. In this segment User can join any project with Join Id and also verification code about entire project.

Figure-25

Appendix A: SRS Document

Functional Requirement

Enter the System

FR-1	User registration/signup.		
Description	Users should be able to sign-up.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Invite Members

FR-2	**Project invitation**		
Description	Users should be able to invite others to projects.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Create a Project Folder

FR-3	**Create a Project Folder**		
Description	Users should be able to create new project folders		
Stakeholders	Project Member, Supervisor	**Priority**	High

Delete a Project Folder

FR-4	**Delete a Project Folder**		
Description	Users should be able to delete new project folders		
Stakeholders	Project Member, Supervisor	**Priority**	High

Create List

FR-5	**Create List**		
Description	Users should be able to create List.		
Stakeholders	Project Member, Supervisor	**Priority**	High

To Do

FR-6	**To Do**		
Description	A feature for Adding Task or Create Task.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Add Task

FR-7	**Add Task**

Description	Users should be able to add Task		
Stakeholders	Project Member, Supervisor	**Priority**	High

Delete Task

FR-8	**Delete Task**		
Description	Users should be able to delete Task.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Assign Member

FR-9	**Assign Member**		
Description	Users should be able to assign member for each.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Share Docs

FR-10	**Share Docs**
Description	Users should be able to upload project documents and share it.

Stakeholders	Project Member, Supervisor	**Priority**	Medium

Commenting

FR-11	Commenting on tasks/documents.		
Description	Users should be able to comment on tasks/documents		
Stakeholders	Project Member, Supervisor	**Priority**	Medium

Get Notifications

FR-12	**Get Notifications**		
Description	Users should receive notifications for updates		
Stakeholders	Project Member, Supervisor	**Priority**	High

Reporting and analytics

FR-13	Reporting and analytics		
Description	Managers/Admins should generate project reports		
Stakeholders	Project Member, Supervisor	**Priority**	High

Non-Functional Requirement

These requirements specify the quality characteristics that the app needs to possess such as performance, security, and usability.

Performance

NFR-1	Performance		
Description	The application should load quickly and be responsive to its user interactions.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Scalability

NFR-2	Scalability		
Description	The application should be able to handle a high number of concurrent users.		
Stakeholders	Project Member, Supervisor /Administrative	**Priority**	High

Security

NFR-3	Security		
Description	The application should protect user data and prevent unauthorized access.		
Stakeholders	Project Member, Supervisor /Administrative	**Priority**	High

Usability

NFR-4	Usability		
Description	The application should be easy to navigate and use for all users, including those with disabilities.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Accessibility

NFR-5	Accessibility		
Description	The application should meet accessibility standards and guidelines, to make it available to as many people as possible.		
Stakeholders	Project Member, Supervisor	**Priority**	Medium

Compatibility

NFR-6	Compatibility		
Description	The application should work on different devices and browsers		
Stakeholders	Project Member, Supervisor	**Priority**	Medium

Maintainability

NFR-7	Maintainability

Description	The code should be well-organized and easy to maintain.		
Stakeholders	Development Team/Administrator	**Priority**	Medium

Backup and Recovery

NFR-8	Backup and Recovery		
Description	Regular backups should be performed, and data recovery ensured		
Stakeholders	Project Member, Supervisor, Administrator	**Priority**	Low

Usability requirements
These requirements specify how easy the app should be to use and navigate, and how accessible it should be to users with different abilities.

Simple and Intuitive navigation

UR-1	Simple and Intuitive navigation		
Description	The application should have a simple, consistent and easy-to-use navigation structure.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Clear and consistent visual design

UR-2	Clear and consistent visual design		
Description	The application should have a clear and consistent visual design, which helps users understand the app's structure and organization.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Accessible for all

UR-3	Accessible for all		
Description	The application should be designed to meet accessibility guidelines and standards, to make it available to as many people as possible.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Error Prevention and Recovery

UR-5	Error Prevention and Recovery		
Description	The application should be designed to prevent errors, and provide clear and concise feedback when errors occur, helping users recover from those errors.		
Stakeholders	Administrator	**Priority**	High

Security requirement
These requirements specify the measures that need to be taken to protect the app and its users' data, such as data encryption, user authentication, and access controls.

User Authentication

SR-1	User Authentication		
Description	Users should be required to provide a valid username and password in order to access their account.		
Stakeholders	System	**Priority**	High

Access control

SR-3	Access control		
Description	The application should restrict access to sensitive data and functionality based on the user's role and authorization level.		
Stakeholders	Administrator	**Priority**	High

Secure Transmission

SR-4	Secure Transmission		
Description	Data transmission between the user's device and the server should be secured using HTTPS.		
Stakeholders	Administrator	**Priority**	High

Regular Software updates

SR-6	Regular Software updates

Description	The application should be regularly updated with the latest security patches and fixes.		
Stakeholders	Administrator	**Priority**	High

Vulnerability scans

SR-7	Vulnerability scans		
Description	The application should be regularly scanned for vulnerabilities and any issues found should be addressed promptly.		
Stakeholders	Administrator	**Priority**	High

Performance requirements
These requirements specify the desired level of responsiveness and speed of the app, as well as the number of concurrent users it should be able to handle.

Load time

PR-1	Load time		
Description	The application should have a fast load time, ideally under a few seconds.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Responsiveness

PR-2	Responsiveness

Description	The application should be responsive to user interactions and provide instant feedback for actions taken.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Server Response Time

PR-3	Server response time		
Description	The server should provide a response time under a few seconds when handling requests from the application.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Concurrent users

PR-4	Concurrent time		
Description	The application should be able to handle a high number of concurrent users without any significant performance degradation.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Scalability

PR-5	Scalability
Description	The application should be able to scale to accommodate increased traffic or user numbers.

Stakeholders	Administrator	**Priority**	High

Optimization

PR-6	Optimization		
Description	The application should be optimized for performance, including techniques like caching, magnification and code splitting.		
Stakeholders	Administrator	**Priority**	High

Performance requirements are important because they determine how well the app will perform under different conditions and how well it can handle high traffic. It's essential that the app is responsive, fast and can handle a high number of concurrent users, as slow loading times and unresponsive interactions can discourage users from using the app. It's also important that the app is designed with scalability in mind, so it can be easily adapted to accommodate increased traffic or user numbers in the future.

Technical requirements

These requirements specify the technical constraints and limitations that the app needs to meet, such as the need to work offline, or to work on different browsers and devices.

Web Platform

TR-1	Web Platform		
Description	The application will be a web-based platform that can be accessed through a browser on desktop and mobile devices.		
Stakeholders	Project Member, Supervisor	**Priority**	High

Database

TR-2	Database		
Description	A database will be used to store information about restaurants, menu items, prices, and user ratings.		
Stakeholders	Project Member, Supervisor	**Priority**	High

User Authentication

TR-3	User Authentication		
Description	The application will have user authentication features to allow users to log in and provide ratings		
Stakeholders	Project Member, Supervisor, System	**Priority**	High

Responsive Design

TR-4	Responsive Design		
Description	The application will have a responsive design to ensure a good user experience on different devices.		
Stakeholders	Project Member, Supervisor, Administrator	**Priority**	High

Use Case Diagram

Figure-26

Use Case Descriptions

Table 07 Enter the System

Use Case	Enter the System	
Goal	Access the IIT Kanban Board Software Project.	
Pre-conditions	The project member has valid login credentials.	
Post conditions	Project member successfully enters the system.	
Success End Condition	The project member successfully enters the system.	
Failed End Condition	Project member is unable to log in or access the system.	
Primary Actors: Secondary Actors:	Supervisor and Project Member N/A	
Trigger	The project member wants to use the system.	
Main Success Flows	Step	Action
	1	The project member opens the IIT Kanban Board Software Project application or website.
	2	The system presents the login screen.
	3	The project member enters their login credentials (username and password).
	4	The project member submits the login or sign-up information.
	5	If the login information valid, then enter into the system.
	6	The system displays the main dashboard of the IIT Kanban Board Software Project, showing relevant project information.
Alternative Flows	Step	Action

	4a	If the login credentials are invalid.
		4a1. The system displays an error message indicating the incorrect credentials.
		4a2. The project member can re-enter the login
		4a3. Information or request password reset if needed.
		4a4. If the project member re-enters the login
		4a5. Information, the system re-verifies the credentials.
		4a6. If the project member requests a password reset, the system follows the password reset process.
Quality Requirements	1. Performance: The system should provide quick and responsive login verification. 2. Reliability: The login process should be reliable, ensuring the project member's access to the system 3. Usability: The login screen should be user-friendly, allowing project members to easily enter their credentials. 4. Security: The system should securely handle and validate login credentials to prevent unauthorized access. 5. Availability: The system should be available and accessible for project members to log in at any time.	

Table 08 Invite Member

Use Case	Invite Members
Goal	Access the IIT Kanban Board Software Project.
Pre-conditions	The project member is logged into the system.
Post conditions	Project member receives an invitation to join the project
Success End Condition	The invitation is successfully sent to the new member(s).
Failed End Condition	Project member is not successfully invited.

Primary Actors: Secondary Actors:	Supervisor and Project Member N/A	
Trigger	The project member wants to invite new members.	
Main Success Flows	Step	Action
	1	The project member navigates to the "Invite Members" section of the IIT Kanban Board Software Project.
	2	The system displays the invite members interface, providing options to enter email addresses or select from a list of existing contacts.
	3	The project member enters the email address(es) of the new member(s) to be invited.
	4	The project member verifies the entered email address(es) and proceeds.
	5	The project member clicks the "Send Invitation" button.
	6	The system displays a confirmation message indicating that the invitations have been successfully sent.
Alternative Flows	Step	Action
	3a	If the project member selects the option to select from a list of existing contacts instead of entering email addresses:
		3a1. The system displays a list of contacts associated with the project member's account.
	6a	6a1. The system displays an error message indicating the failure to send the invitations
		6a2. The project member can choose to retry sending the invitations or cancel the operation.
Quality Requirements	1. Performance: The system should send invitations promptly and handle multiple invites efficiently.	

	2. Reliability: The invitation sending process should be reliable, ensuring that all invitations are sent without errors. 3. Usability: The invite members interface should be intuitive and user-friendly, allowing project members to easily invite new members. 4. Security: The system should securely handle email addresses and ensure that invitations are sent to the correct recipients. 5. Availability: The invite members feature should be available and accessible for project members whenever needed.

Table 09 Create Project Folder

Use Case	Create Project Folder	
Goal	Create a new project folder and associated task list.	
Pre-conditions	The supervisor is logged into the IIT Kanban Board Software Project.	
Post conditions	Project folder is successfully created	
Success End Condition	The project member successfully enters the system.	
Failed End Condition	Project folder is not created or an error occurs.	
Primary Actors: Secondary Actors:	Supervisor Project Member	
Trigger	The supervisor wants to organize a new project.	
Main Success Flows	Step	Action

	1.	The supervisor selects the "Create Project Folder" option from the main menu of the IIT Kanban Board Software Project.
	2.	The supervisor enters the project folder name, description, and any other required information.
	3.	The supervisor enters the project folder name, description, and any other required information.
	4.	The supervisor clicks the "Create" button to create the project folder.
	5.	The system validates the entered information and creates the project folder.
	6.	The system displays the created project folder on the main dashboard.
	7.	The supervisor selects the created project folder.
	8.	The system displays the project folder interface and provides an option to create a new task list within the folder.
	9.	The supervisor selects the "Create List" option.
	10.	The system prompts the supervisor to enter the name and other details of the new task list.
	11.	The supervisor enters the task list details and clicks the "Create" button.
	12.	The system validates the entered information and creates the task list within the project folder.
	13.	The system displays the created task list within the project folder.
Alternative Flows	Step	Action
	9a	If the supervisor chooses to create multiple task lists within the project folder:
		9a1. The supervisor repeats steps 9-13 for each additional task list.
Quality Requirements	1.	Performance: The system should create project folders and associated task lists quickly and responsively.

	2. Reliability: The creation process should be reliable, ensuring that project folders and task lists are created without errors. 3. Usability: The interface for creating project folders and task lists should be user-friendly and intuitive for the supervisor. 4. Security: The system should ensure the confidentiality and access control of project folders and task lists. 5. Availability: The creation feature should be available and accessible for the supervisor to create project folders and task lists at any time.

Table 10 Delete Project Folder

Use Case	Delete the Project Folder	
Goal	Delete a project folder and its associated contents.	
Pre-conditions	The project member has valid login credentials.	
Post conditions	The project folder is successfully deleted.	
Success End Condition	The project member successfully enters the system.	
Failed End Condition	Project folder is not deleted or an error occurs.	
Primary Actors: Secondary Actors:	Supervisor Project Member	
Trigger	The primary actor (Supervisor or Project Member) wants to delete a project folder.	
Main Success Flows	Step	Action

	1.	The primary actor (Supervisor or Project Member) selects the project folder to be deleted from the IIT Kanban Board Software Project interface.
	2.	The system displays the project folder interface, showing the folder's contents and options.
	3.	The primary actor (Supervisor or Project Member) locates the "Delete Folder" option within the interface.
	4.	The primary actor (Supervisor or Project Member) clicks on the "Delete Folder" option.
	5.	The system prompts the primary actor (Supervisor or Project Member) to confirm the deletion action.
	6.	The primary actor (Supervisor or Project Member) confirms the deletion by clicking the "Confirm" button.
	7.	The system verifies the primary actor's permissions and proceeds with the deletion process.
	8.	The system deletes the project folder and its associated contents, including tasks, lists, and files. The system displays a confirmation message indicating that the project folder has been successfully deleted.
Alternative Flows	Step	Action
	7a	If the primary actor (Supervisor or Project Member) does not have sufficient permissions to delete the folder:
		4a1. The system displays an error message indicating the lack of permissions.
Quality Requirements	1.	Performance: The system should handle project folder deletions quickly, even with large amounts of associated data.
	2.	Reliability: The deletion process should be reliable, ensuring that the project folder and its contents are permanently removed.
	3.	Usability: The delete folder option should be easily accessible and understandable for both supervisors and project members.

	4. Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the deletion to prevent unauthorized removal of important project data. 5. Availability: The delete folder feature should be available and functional at all times for authorized primary actors (Supervisor or Project Member).

Table 11 Create List

Use Case	Create List	
Goal	Create a new list within a project folder.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	New list is successfully created and added to the project.	
Success End Condition	The project member successfully enters the system.	
Failed End Condition	New list is not created or an error occurs.	
Primary Actors: Secondary Actors:	Supervisor N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to create a new list.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project folder where the list will be created.
	2.	The system displays the project folder interface, showing the existing lists and options.

	3.	The primary actor (Supervisor or Project Member) locates the "Create List" or "Add List" option within the interface.
	4.	The primary actor (Supervisor or Project Member) clicks on the "Create List" or "Add List" option.
	5.	The system presents a form or input field for the primary actor (Supervisor or Project Member) to enter the name of the new list.
	6.	The primary actor (Supervisor or Project Member) enters the desired name for the new list.
	7.	The primary actor (Supervisor or Project Member) submits the form or confirms the list creation.
	8.	The system validates the entered information and creates the new list within the project folder.
	9.	The system displays a confirmation message indicating that the new list has been successfully created.
Alternative Flows	Step	Action
	8a	If the primary actor (Supervisor or Project Member) enters a duplicate or invalid name for the new list:
		8a1. The system displays an error message indicating the issue with the entered name.
Quality Requirements	1.	Performance: The system should create lists quickly, even within project folders with a large number of existing lists.
	2.	Reliability: The list creation process should be reliable, ensuring that the new list is accurately created and visible to authorized users.
	3.	Usability: The create list option should be easily accessible and intuitive for both supervisors and project members.
	4.	Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the list creation to prevent unauthorized additions.
	5.	Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the list creation to prevent unauthorized additions.

Table 12 To Do

Use Case	To Do	
Goal	Create a new "To Do" task within a project.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	To-do task is successfully added or updated	
Success End Condition	A new "To Do" task is successfully created within the project.	
Failed End Condition	To-do task is not added or updated.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to create a new "To Do" task.	
Main Success Flows	Step	Action
	1	The primary actor (Supervisor or Project Member) navigates to the project or list where the task will be created.
	2	The system displays the project or list interface, showing the existing tasks and options.
	3	The primary actor (Supervisor or Project Member) locates the "Create Task" or "Add Task" option within the interface.
	4	The primary actor (Supervisor or Project Member) clicks on the "Create Task" or "Add Task" option.
	5	The system presents a form or input fields for the primary actor (Supervisor or Project Member) to enter task details such as title, description, due date, assignee, and priority.

	6	The primary actor (Supervisor or Project Member) enters the required information for the new "To Do" task.
	7	The primary actor (Supervisor or Project Member) submits the form or confirms the list creation.
	8	The system validates the entered information and creates the new "To Do" task within the project or list.
	9	The system displays a confirmation message indicating that the new "To Do" task has been successfully created.
Alternative Flows	Step	Action
	8a	If the primary actor (Supervisor or Project Member) enters incomplete or invalid information for the task:
		8a1. The system displays an error message indicating the missing or incorrect information.
		8a2.If the primary actor (Supervisor or Project Member) cancels the task creation, the system returns to the project or list interface without creating the task.
Quality Requirements	1. The system should validate and enforce any required fields for the task. 2. The system should provide real-time updates for task changes. 3. The task status should be synchronized across all project members.	

Table 13 Add Task

Use Case	Add task
Goal	Add a new task to a project
Pre-conditions	Project member is logged in and has access to the project
Post conditions	The new task is successfully added to the project
Success End Condition	The new task is added to the project

Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	Project member wants to add a task	
Main Success Flows	step	Action
	1.	Project member selects the project where they want to add the task.
	2.	Project member navigates to the "Tasks" section of the project
	3.	Project member clicks on the "Add Task" button.
	4.	Project member fills in the necessary details for the task (e.g., title, description, due date)
	5.	Project member clicks on the "Save" button to submit the task.
	6.	The system validates the input and creates a new task associated with the project.
Alternative Flow	N/A	
Quality Requirements	1. The system should validate the input to ensure that required fields are filled and the data is in the correct format. 2. The system should provide clear error messages when there are validation errors. 3. The task creation process should be efficient and responsive to provide a smooth user experience.	

	4. The system should handle concurrent task creation requests properly to avoid conflicts. 5. The new task should be added to the project and visible to authorized project members.

Table 14 Delete Task

Use Case	Delete Task	
Goal	Delete an existing task within a project or list.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	Task is successfully deleted.	
Success End Condition	The task is successfully deleted from the project or list.	
Failed End Condition	Task is not deleted.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to delete a task.	
Main Success Flows	Step	Action
	1	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task to be deleted.
	2	The system displays the project or list interface, showing the existing tasks and options.
	3	The primary actor (Supervisor or Project Member) locates the task to be deleted within the interface.

	4	The primary actor (Supervisor or Project Member) selects the task to be deleted.
	5	The system presents a confirmation dialog or prompts for confirmation before deleting the task.
	6	The primary actor (Supervisor or Project Member) confirms the deletion by clicking on the "Delete" or "Confirm" button.
	7	The system displays a confirmation message indicating that the task has been successfully deleted.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) cancels the deletion:
		6a1. The system closes the confirmation dialog or returns to the project or list interface without deleting the task.
		6a2. The task remains in the project or list without any changes.
Quality Requirements		1. Performance: The system should delete tasks quickly, even within projects or lists with a large number of existing tasks. 2. Reliability: The task deletion process should be reliable, ensuring that the task is accurately removed from the project or list. 3. Usability: The task deletion option should be easily accessible and intuitive for both supervisors and project members. 4. Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the task deletion to prevent unauthorized removals. 5. Availability: The task deletion feature should be available and functional at all times for authorized primary actors (Supervisor or Project Member).

Table 15 Assign Member

Use Case	Assign Member	
Goal	Assign a project member to a task within a project.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	Project member is successfully assigned to the task.	
Success End Condition	A project member is successfully assigned to the task.	
Failed End Condition	Project member is not assigned to the task.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to assign a member to a task.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task.
	2.	The system displays the project or list interface, showing the existing tasks and options.
	3.	The primary actor (Supervisor or Project Member) locates the task to which they want to assign a member within the interface.
	4.	The primary actor (Supervisor or Project Member) selects the task to be assigned.
	5.	The system displays the task details, including the assigned member (if any) or an option to assign a member.
	6.	The primary actor (Supervisor or Project Member) clicks on the "Assign Member" option.

	7.	The system presents a list of project members or allows the primary actor to search and select a member to be assigned.
	8.	The primary actor (Supervisor or Project Member) selects the project member to be assigned to the task.
	9.	The primary actor (Supervisor or Project Member) confirms the assignment by clicking on the or "Confirm" button.
	10.	The system updates the task details to reflect the assigned member and notifies the relevant parties.
	11.	The system displays a confirmation message indicating that the member has been successfully assigned to the task.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) decides not to assign a member:
		6a1 The primary actor (Supervisor or Project Member) cancels the assignment process.
		6a2. The system returns to the task details without making any changes to the assignment.
Quality Requirements		1. Performance: The system should assign members to tasks quickly, even within projects or lists with a large number of existing tasks and members. 2. Reliability: The member assignment process should be reliable, ensuring that the assigned member is accurately recorded and visible to authorized users. 3. Usability: The member assignment option should be user-friendly, allowing supervisors and project members to easily assign and view assigned members. 4. Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the member assignment to prevent unauthorized assignments.

	5. Availability: The member assignment feature should be available and functional at all times for authorized primary actors (Supervisor or Project Member).

Table 16 Set Priority

Use Case	Set priority	
Goal	Set the priority level of a task within a project.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	The priority level of the task is successfully updated.	
Success End Condition	The priority of the task is successfully set.	
Failed End Condition	The priority level of the task is not updated.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to set the priority of a task.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task.
	2.	The system displays the project or list interface, showing the existing tasks and options.
	3.	The primary actor (Supervisor or Project Member) locates the task for which they want to set the priority within the interface.
	4.	The primary actor (Supervisor or Project Member) selects the task to be prioritized.

	5.	The system displays the task details, including the current priority or an option to set the priority.
	6.	The primary actor (Supervisor or Project Member) clicks on the "Set Priority" or "Change Priority" option.
	7.	The system presents a list or menu of priority levels (e.g., high, medium, low) or allows the primary actor to enter a numeric value.
	8.	The primary actor (Supervisor or Project Member) selects the desired priority level or enters the numeric value for the task.
	9.	The primary actor (Supervisor or Project Member) confirms the priority setting by clicking on the or "Confirm" button.
	10.	The system updates the task details to reflect the new priority level and notifies the relevant parties.
	11.	The system displays a confirmation message indicating that the priority of the task has been successfully set.
Alternative Flow	N/A	
Quality Requirements	1. Performance: The system should assign members to tasks quickly, even within projects or lists with a large number of existing tasks and members. 2. Reliability: The member assignment process should be reliable, ensuring that the assigned member is accurately recorded and visible to authorized users. 3. Usability: The member assignment option should be user-friendly, allowing supervisors and project members to easily assign and view assigned members. 4. Security: The system should verify the permissions of the primary actor (Supervisor or Project Member) before allowing the member assignment to prevent unauthorized assignments.	

	5. Availability: The member assignment feature should be available and functional at all times for authorized primary actors (Supervisor or Project Member).

Table 17 Set Due date

Use Case	Set Due Date
Goal	Set the due date for a task within a project
Pre-conditions	The primary actor has appropriate permissions.
Post conditions	Project member is successfully assigned to the task.
Success End Condition	The due date of the task is successfully set.
Failed End Condition	Project member is not assigned to the task.
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A
Trigger	The primary actor (Supervisor or Project Member) wants to set the due date of a task.

Main Success Flows	Step	Action
	1.	Supervisor navigates to the project dashboard
	2.	Supervisor selects the project
	3.	Supervisor selects the task to update the due date
	4.	Supervisor updates the due date or change the due date of the task
	5.	Supervisor selects 'confirm' to save the changes
	6.	The due date of the task is updated
Alternative Flow	N/A	
Quality Requirements	1. The system should display the current due date of the task. 2. The system should provide a date picker for selecting the due date. 3. The system should validate the due date selection and update the task accordingly.	

Table 18 Add Description

Use Case	Add Description
Goal	Add a description to provide additional details or context to a task.
Pre-conditions	The primary actor has appropriate permissions.
Post conditions	The description is successfully added to the task.
Success End Condition	The primary actor has appropriate permissions.
Failed End Condition	The description is not added to the task.
Primary Actors:	Supervisor, Project Member

Secondary Actors:	N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to add a description to a task.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task.
	2.	The system displays the project or list interface, showing the existing tasks and options.
	3.	The primary actor (Supervisor or Project Member) locates the task to which they want to add a description within the interface.
	4.	The primary actor (Supervisor or Project Member) selects the task to access the task details.
	5.	The system displays the task details, including any existing description or an option to add a description.
	6.	The primary actor (Supervisor or Project Member) clicks on the "Add Description" or "Edit Description" option.
	7.	The system presents a text field or editor where the primary actor can enter the description.
	8.	The primary actor (Supervisor or Project Member) enters the desired description for the task.
	9.	The primary actor (Supervisor or Project Member) confirms the addition of the description by clicking on the "Confirm" button.
	10.	The system updates the task details to include the new description and notifies the relevant parties.
	11.	The system displays a confirmation message indicating that the description has been successfully added to the task.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) decides not to add a description:

		6a1 The primary actor (Supervisor or Project Member) cancels the description addition process.
		6a2. The system returns to the task details without making any changes to the description.
Quality Requirements	1. The system should provide a text editor for entering the description. 2. The system should allow formatting options for the description. 3. The system should display the current description of the task.	

Table 19 Create Board

Use Case	Create Board	
Goal	Create a new board within the project.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	The new board is successfully created in the project.	
Success End Condition	A new board is successfully created within the project.	
Failed End Condition	The new board is not created in the project.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to create a new board.	
Main Success Flows	Step	Action

	1.	The primary actor (Supervisor or Project Member) navigates to the project where they want to create a new board.
	2.	The primary actor (Supervisor or Project Member) navigates to the project where they want to create a new board.
	3.	The primary actor (Supervisor or Project Member) locates the option to create a new board within the interface.
	4.	The primary actor (Supervisor or Project Member) selects the "Create New Board" option.
	5.	The system presents a form or dialog where the primary actor can enter the details of the new board.
	6.	The primary actor (Supervisor or Project Member) enters the required information, such as the board name, description, or any other relevant details.
	7.	The primary actor (Supervisor or Project Member) confirms the creation of the new board by clicking on the "Confirm" button.
	8.	The system creates the new board within the project, assigns appropriate permissions, and updates the project interface.
	9.	The system displays a confirmation message indicating that the new board has been successfully created.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) decides not to create a new board:
		6a1. The primary actor (Supervisor or Project Member) cancels the board creation process.
		6a2. The primary actor (Supervisor or Project Member) cancels the board creation process.
Quality Requirements	1.	The system should validate that a unique name is provided for the new board.

	2. The system should provide options for categorizing the new board. 3. The system should allow customization of the board's layout and design.

Table 20 Add Status

Use Case	Add Status	
Goal	Add a status to indicate the progress of a task.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	The status is successfully added to the task.	
Success End Condition	The status is successfully added to the task.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to add a status to a task.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task.
	2.	The system displays the project or list interface, showing the existing tasks and options.
	3.	The primary actor (Supervisor or Project Member) locates the task to which they want to add a status within the interface.

	4.	The primary actor (Supervisor or Project Member) selects the task to access the task details.
	5.	The system displays the task details, including any existing status or an option to add a status.
	6.	The primary actor (Supervisor or Project Member) clicks on the "Add Status" or "Edit Status" option.
	7.	The system presents a list of predefined statuses or allows the primary actor to enter a custom status.
	8.	The primary actor (Supervisor or Project Member) selects the desired status from the list or enters a custom status.
	9.	The primary actor (Supervisor or Project Member) confirms the addition of the status by clicking on the "Confirm" button.
	10.	The system updates the task details to include the new status and notifies the relevant parties.
	11.	The system displays a confirmation message indicating that the status has been successfully added to the task.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) decides not to add a status:
		6a1.The primary actor (Supervisor or Project Member) cancels the status addition process.
		6a2.The system returns to the task details without making any changes to the status.
Quality Requirements	1. The system should provide a clear and intuitive interface for selecting the status. 2. The system should allow customization of status options based on project requirements. 3. The system should validate that a status is selected before saving.	

Table 21 Add Comments

Use Case	Add Comments	
Goal	Add comments to provide additional information or updates to a task.	
Pre-conditions	The primary actor has appropriate permissions.	
Post conditions	The comments are successfully added to the task.	
Success End Condition	The status is successfully added to the task.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to add a comment to a task.	
Main Success Flows	Step	Action
	1.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the task.
	2.	The system displays the project or list interface, showing the existing tasks and options.
	3.	The primary actor (Supervisor or Project Member) locates the task to which they want to add a comment within the interface.
	4.	The primary actor (Supervisor or Project Member) selects the task to access the task details.
	5.	The system displays the task details, including any existing comments or an option to add a comment.
	6.	The primary actor (Supervisor or Project Member) clicks on the "Add Comment" option.

	7.	The system presents a text field or dialog where the primary actor can enter the comment text.
	8.	The primary actor (Supervisor or Project Member) enters the comment text providing relevant information or updates.
	9.	The primary actor (Supervisor or Project Member) confirms the addition of the comment by clicking on the "Confirm" button.
	10.	The system updates the task details to include the new comment and notifies the relevant parties.
	11.	The system displays a confirmation message indicating that the comment has been successfully added to the task.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) decides not to add a comment.
		6a1. The primary actor (Supervisor or Project Member) cancels the comment addition process.
Quality Requirements	1. The system should provide a user-friendly interface for entering comments. 2. The system should support formatting options for comments (e.g., text styling, links, etc.). 3. The system should ensure the comments are associated with the correct task.	

Table 22 Share Documents

Use Case	Share Documents
Goal	Share documents with other project members.
Pre-conditions	The primary actor has appropriate permissions.

Post conditions	Documents are successfully shared with the project members.	
Success End Condition	The document is successfully shared with the intended recipients.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to share a document.	
Main Success Flows	Step	Action
	10.	The primary actor (Supervisor or Project Member) navigates to the project or list containing the document.
	11.	The system displays the project or list interface, showing the existing documents and options.
	12.	The primary actor (Supervisor or Project Member) locates the document they want to share within the interface.
	13.	The system presents a "Share" option for the selected document.
	14.	The primary actor (Supervisor or Project Member) clicks on the "Share" option.
	15.	The system displays a dialog or form to enter the email addresses or usernames of the intended recipients.
	16.	The primary actor (Supervisor or Project Member) enters the email addresses or usernames of the recipients.
	17.	The primary actor (Supervisor or Project Member) confirms the sharing action by clicking on the "Confirm" button.
	18.	The system sends email notifications or generates access links to the shared document for the intended recipients.

	19.	The system updates the document sharing status, indicating that it has been shared with the specified recipients.
	20.	The system displays a confirmation message indicating that the document has been successfully shared.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) wants to share the document with all project members:
		6a1. The primary actor selects an option to share the document with all project members.
		6a2. The system automatically includes all project members as recipients.
Quality Requirements	1. The system should provide a secure and reliable document sharing mechanism. 2. The system should allow the Supervisor to select multiple documents to share. 3. The system should provide an efficient way to select project members for document sharing. 4. The system should ensure that only authorized project members have access to the shared documents.	

Table 23 Download Documents

Use Case	Download Documents
Goal	Download a document for offline access or storage.
Pre-conditions	The primary actor has appropriate permissions.
Post conditions	Supervisor or Project Member receives the notifications.
Success End Condition	The document is successfully downloaded to the primary actor's device or storage.
Failed End Condition	N/A

Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) wants to download a document.	
Main Success Flows	Step	Action
	1	The primary actor (Supervisor or Project Member) navigates to the project or list containing the document.
	2	The system displays the project or list interface, showing the existing documents and options.
	3	The primary actor (Supervisor or Project Member) locates the document they want to download within the interface.
	4	The primary actor (Supervisor or Project Member) selects the document to access the document details or options.
	5	The system presents a "Download" option for the selected document.
	6	The primary actor (Supervisor or Project Member) clicks on the "Download" option.
	7	The system initiates the download process and transfers the document to the primary actor's device or storage.
	8	The primary actor (Supervisor or Project Member) receives the downloaded document and can access it offline or store it as needed.
Alternative Flow	N/A	

Quality Requirements	1. The system should provide a secure and reliable document download mechanism. 2. The system should allow the Supervisor or Project Member to select multiple documents for download. 3. The system should ensure that only authorized users have access to download documents. 4. The system should provide an efficient and fast download process.

Table 24 Messaging

Use Case	Messaging	
Goal	Easily collaboration with the project member, upload file and download file.	
Pre-conditions	Must have to be included as a project member, or supervisor.	
Post conditions	Identification update is generated and available for viewing.	
Success End Condition	Identification update is successfully generated and available.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Members Supervisor, Project Members	
Trigger	Press the Message Icon.	
Main Success Flows	Step	Action
	1	Select List.
	2	Press the Message Icon.
	3	Write text and share docs.

	4	For send click on send icon.
Alternative Flow	N/A	
Quality Requirements	1. Delivery within 5 second. 2. Message seen Understanding with good interface in blue color.	

Table 25 Get Notifications

Use Case	Get Notifications	
Goal	Receive timely notifications about project updates or relevant activities.	
Pre-conditions	The primary actor has subscribed to receive notifications and has appropriate permissions.	
Post conditions	The new board is successfully created in the project.	
Success End Condition	The primary actor receives the notification with the relevant information.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The system detects a project update or activity that requires notification.	
Main Success Flows	Step	Action

	1	The primary actor receives the notification with the relevant information.
	2	The system generates a notification containing the relevant information, such as project name, task details, user name, or comment text.
	3	The system checks the notification preferences and permissions of the primary actor (Supervisor or Project Member).
	4	The system sends the notification to the primary actor via their preferred communication channel, such as email, in-app notification, or mobile push notification.
	5	The primary actor (Supervisor or Project Member) receives the notification.
	6	The primary actor reviews the notification and takes appropriate action based on the information provided
Alternative Flows	Step	Action
	3a	If the primary actor (Supervisor or Project Member) has specified custom notification settings:
		3a1. The system checks the primary actor's customized notification preferences and channels.
		3a2. The system sends the notification through the specified channels according to the primary actor's preferences.
Quality Requirements		1. The system should provide customizable notification settings to allow users to choose their preferred notification types. 2. The system should ensure timely delivery of notifications to the Supervisor or Project Member. 3. The system should provide clear and concise notifications with relevant information. 4. The system should support various notification channels (e.g., email, in-app notifications).

Table 26 Notify Update Progress

Use Case	Notify Update Progress	
Goal	Notify progress updates to relevant project members.	
Pre-conditions	The primary actor has appropriate permissions to update progress and has subscribed to receive notifications.	
Post conditions	Supervisor and Project Member receive the progress update.	
Success End Condition	A new board is successfully created within the project.	
Failed End Condition	The new board is not created in the project.	
Primary Actors: Secondary Actors:	Supervisor, Project Member N/A	
Trigger	The primary actor (Supervisor or Project Member) updates the progress of a task or project.	
Main Success Flows	Step	Action
	1	The primary actor (Supervisor or Project Member) updates the progress of a task or project.
	2	The system detects the progress update and generates a notification.
	3	The system checks the notification preferences and permissions of relevant project members.
	4	The system sends notifications to the relevant project members regarding the progress update.
	5	The relevant project members receive the notifications containing the progress update information.
Alternative Flows	Step	Action
	6a	If the primary actor (Supervisor or Project Member) wants to notify specific project members:

		6a1. The primary actor selects the specific project members to notify about the progress update.
		6a2. The system checks the notification preferences and permissions of the selected project members.
Quality Requirements	1. The system should promptly notify the Supervisor and Project Member of the progress update. 2. The progress update notification should include relevant information such as task/project name, completed percentage, and any additional notes. 3. The system should handle concurrent progress updates accurately to avoid conflicts or data loss. 4. The notification should be delivered to the Supervisor and Project Member through their preferred communication channels (e.g., email, in-app notification).	

Description For Administrator

Table 27 Control Privacy

Use Case	Control Privacy
Goal	Manage privacy settings and access permissions for users or project members.
Pre-conditions	The Administrator has appropriate privileges and access to the system.
Post conditions	Privacy settings are updated and enforced in the system.
Success End Condition	Privacy settings and access permissions are successfully modified.
Failed End Condition	Privacy settings are not updated or enforced as intended.
Primary Actors: Secondary Actors:	Administrator N/A

Trigger	The Administrator needs to modify privacy settings or access permissions.	
Main Success Flows	Step	Action
	1.	Administrator accesses the privacy control interface
	2.	Administrator selects the desired privacy settings or options
	3.	The system updates the privacy settings and enforces them
Alternative Flow	N/A	
Quality Requirements	1. The system should provide a user-friendly interface for privacy control. 2. Privacy settings should be granular and customizable. 3. The system should enforce the privacy settings consistently. 4. Privacy settings should be properly documented and communicated 5. The system should log any privacy control actions for audit purposes 6. The privacy control interface should be secure and protected	

Table 28 Control Updating

Use Case	Control Updating
Goal	Control and manage software updates in the system.
Pre-conditions	Administrator has appropriate access and permissions.
Post conditions	Software updates are controlled and managed in the system.

Success End Condition	Software updates are successfully controlled and managed.	
Failed End Condition	Software updates are not controlled or managed as intended.	
Primary Actors: Secondary Actors:	Administrator N/A	
Trigger	Administrator initiates update control actions.	
Main Success Flows	Step	Action
	1.	Administrator accesses the update control interface.
	2.	Administrator selects the desired update control options.
	3.	The system applies the update control actions.
Alternative Flow	N/A	
Quality Requirements	1. The system should provide a user-friendly interface for update control. 2. Update control options should be comprehensive and flexible. 3. The system should enforce update control consistently 4. Update control actions should be properly documented and auditable. 5. The update control interface should be secure and protected.	

Description For System

Table 29 Manage Date/Time

Use Case	Mange Date/Time	
Goal	Manage the date and time settings of the system.	
Pre-conditions	System is operational and capable of managing date/time.	
Post conditions	Date and time settings are successfully managed by the system.	
Success End Condition	Date and time settings are successfully managed by the system.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	System N/A	
Trigger	System initialization or user request to change date/time settings.	
Main Success Flows	Step	Action
	1.	System initializes or user requests to change date/time settings.
	2.	System displays the current date and time settings.
	3.	User provides input to change the date and time settings.
	4.	System verifies the validity of the new settings
	5.	System updates and applies the new date and time settings
Alternative Flow	N/A	

Quality Requirements	1. The system should provide an intuitive and user-friendly interface for managing date/time settings. 2. The system should validate the input for date and time settings to ensure they are within acceptable ranges. 3. The system should handle time zone conversions accurately and seamlessly. 4. Date and time changes should be reflected across the system in a synchronized manner. 5. The system should have the capability to automatically synchronize with external time sources.

Table 30 Manage Notification

Use Case	Mange Notification	
Goal	Manage the notification settings of the system	
Pre-conditions	System is operational and capable of managing notifications.	
Post conditions	Notification settings are successfully managed by the system.	
Success End Condition	Notification settings are successfully managed by the system.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	System N/A	
Trigger	System initialization or user request to manage notifications	
Main Success Flows	Step	Action
	1	System initializes or user requests to manage notifications.

	2	System displays the current notification settings.
	3	User provides input to enable/disable or customize settings.
	4	System updates and applies the new notification settings.
Alternative Flow	N/A	
Quality Requirements	1. The system should provide an intuitive interface for managing notification settings. 2. The system should allow users to enable/disable notifications for specific events or modules. 3. The system should provide options for customizing notification preferences. 4. The system should support multiple notification channels, such as email, push notifications, or SMS. 5. The system should deliver notifications in a timely and reliable manner 6. The system should provide the ability to schedule notifications for specific times or recurring intervals.	

Table 31 Generate Gantt chart

Use Case	Generate Gantt Chart
Goal	Generate a Gantt chart for project scheduling.
Pre-conditions	Project data and scheduling information are available.
Post conditions	Gantt chart is generated and available for viewing.
Success End Condition	Gantt chart is successfully generated and available for viewing.
Failed End Condition	N/A

Primary Actors: Secondary Actors:	System N/A	
Trigger	User request to generate a Gantt chart.	
Main Success Flows	Step	Action
	1	User requests to generate a Gantt chart.
	2	System retrieves project data and scheduling information.
	3	System generates the Gantt chart based on the project data.
	4	System presents the generated Gantt chart to the user.
Alternative Flow	N/A	
Quality Requirements	1. The system should accurately represent the project schedule in the generated Gantt chart. 2. The Gantt chart should provide a clear visualization of project tasks, durations, and dependencies. 3. The system should handle complex project schedules and dependencies. 4. The Gantt chart should be generated in a timely manner. 5. The system should allow users to customize the appearance and format of the Gantt chart.	

	6. The system should provide options for exporting or sharing the generated Gantt chart. 7. The Gantt chart should be easily readable and understandable by project stakeholders. 8. The system should handle updates to project data and reflect the changes in the generated Gantt chart.

Table 32 Identity Update

Use Case	Identify Update	
Goal	Generate an identification update for a project.	
Pre-conditions	Project data and status information are available.	
Post conditions	Identification update is generated and available for viewing.	
Success End Condition	Identification update is successfully generated and available.	
Failed End Condition	N/A	
Primary Actors: Secondary Actors:	System N/A	
Trigger	User request to generate an identification update.	
Main Success Flows	Step	Action
	1	User requests to generate an identification update.
	2	System retrieves project data and status information.
	3	System generates the identification update.

	4	System presents the generated identification update to the user.
Alternative Flow	N/A	
Quality Requirements	3. The system should accurately reflect the project status in the identification update. 4. The identification update should provide a concise summary of the project status 5. The system should handle updates to project data and reflect the changes in the identification update 6. The identification update should be generated in a The system should allow users to customize the content and format of the identification update 7. The identification update should be easily readable and understandable by project stakeholders 8. The system should provide options for exporting or sharing the generated identification update	

Table 33 Generate Report

Use Case	Generate Report
Goal	Generate a comprehensive evaluation report for project members.
Pre-conditions	Project data and evaluation metrics are available.
Post conditions	Report is generated and made available.
Success End Condition	Report is successfully generated and accessible
Failed End Condition	N/A

Primary Actors: Secondary Actors:	System N/A	
Trigger	System detects the need to generate a report.	
Main Success Flows	Step	Action
	1	Enter into IIT Kanban Board Software.
	2	Select the report.
	3	Click on Generate report.
	4	System will show the report containing necessary information like project name, Supervisor and teammate identity, Project Description, Assign task, date and time, Gantt Chart.
	5	System will provide final report
Alternative Flow	N/A	
Quality Requirements	1. Generate the report within 5 sec 2. Generate report in pdf format	

Source: SRS (Full Document)

Appendix B: Project Proposal

Introduction

This is a proposal report for Software Project Lab II. This project will be developed during the 5th semester of BSSE, Institute of Information Technology, Noakhali Science and Technology University from 29 March 2023 by the proposal presentation, and the final project will be delivered on 07 September 2023. Our team will work along with our regular academic courses.

Motivation

In IIT we have done each of our SPL through analog process in previous time. There is no well furnished technological platform to understand the improvement of our own work. But in recent time there are numerous developed software for overlooking the improvement of our work, communicating with each others and sharing documents etc. But these resources are not free of

cost. So we want to develop a project that aims for enriching our technological knowledge and leads IIT to forward.

Objectives

The primary objective of the "IIT Kanban Board Software Project" is to revolutionize project management and collaboration within the IIT community. Through the development of a digital platform, this project aims to provide teachers and students with efficient tools for planning, tracking, and managing their academic and research projects. The core objectives include enhancing collaboration among project members, boosting productivity by automating manual tasks, and facilitating seamless communication. The software's primary focus is on improving task management, document organization, and providing real-time notifications to keep project members informed. Additionally, data privacy and security are of utmost importance to ensure the protection of sensitive academic information. By achieving these objectives, the project aspires to usher in a new era of digital project management at IIT, ultimately leading to improved project outcomes and more effective academic and research endeavors.

Target Customers

The target customers for our system are students and Teachers.

Model

Our project is highly inspired by SPL I. We faced different problems with our selected model to develop our project which was an iterative process. This time, taking experience from the previous year's project, we want to implement something new and we have selected the agile model to develop this project. Agile Model break tasks into smaller iterations. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

Tools and Resources

Category	**Name or Description**
Text Editor	Visual Studio Code
Server	XAMPP
RDBMS	MySQL
Language	HTML, CSS, JavaScript, PHP, SQL
Learning Resource	1. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages 2nd Edition by Elisabeth Robson 2. Head First JavaScript Programming: A Brain-Friendly Guide 1st Edition by Eric Freeman

	3. PHP & MySQL: Server-side Web Development by Jon Duckett

Challenges

The main challenges to face during developing this system are,

1. The software must be designed with a focus on usability and a positive user experience. Complex or unintuitive interfaces could hinder adoption and efficiency.
2. Gathering feedback from project members and continuously improving the software based on their input will be crucial for its long-term success and relevance.
3. As the number of users and projects grows, the system must be scalable to handle increased data and user load without performance degradation.

Source: Project Proposal (Full Document)

Appendix C: Database Schema

Figure-27

ER Diagram

Figure-28