

Database Management (Web Application)

Name: Rhythm Paudel

Email: rhythmpaudel12@gmail.com

Github: github.com/rhythm-paudel

LinkedIn: linkedin.com/in/rhythmpaudel

Abstract

A web application for a company LS Corporation for managing database is being developed. The application will be able to view a overview of projects in the company. Furthermore, in the application the user will also be able perform basic CRUD operations related to project and users from the web application. Before the development of the application according to the record data, normalization was performed. Additionally, futher discussion regarding the project development will also be discussed in the report.s

Table of Contents

1. Introduction	1
2. Textual Analysis	2
2.1. User and Project	2
2.2. Project and Task.....	2
2.3. User and Task	2
2.4. Task and Subtask.....	2
2.5. User, Comment, and Task.....	3
2.5.1. User and Comment	3
2.5.2. Task and Comment.....	3
2.6. Task and Resources.....	4
2.7. Project and Milestones.....	4
2.8. User and subtask	4
3. Entity Relationship Diagram	5
4. Normalization.....	7
4.1. UNF:.....	7
4.2. 1NF	7
4.3. 2NF (Checking Partial Dependencies).....	9
4.4. 3NF (Checking for Transitive Dependencies)	12
4.5. Final Tables	15
5. Integration and Assumption	16
5.1. Integration	16
5.1.1. Eliminated tables	17
5.1.2. Added tables.....	17
5.2. Assumptions.....	18
6. Final Entity Relationship Diagram (ERD)	19
7. Data Dictionary	20

8.	Scripts.....	31
8.1.	Roles of user	31
8.2.	UsersDetails	31
8.3.	ProjectDetails	32
8.4.	UserProject	32
8.5.	Milestones	33
8.6.	Tasks	33
8.7.	UserProjectTaskDetails	34
8.8.	Comments	35
8.9.	Resources	35
8.10.	ResourcesTask.....	36
8.11.	Subtasks.....	37
8.12.	UserSubtask.....	38
9.	Data insertion	39
9.1.	RolesOfUser.....	39
9.2.	UsersDetails	39
9.3.	ProjectDetails	40
9.4.	UserProject	40
9.5.	Milestones	41
9.6.	Tasks	41
9.7.	UserProjectTaskDetails	42
9.8.	Comments	42
9.9.	Resources	43
9.10.	ResourcesTask.....	43
9.11.	Subtask	44
9.12.	UserSubtask.....	44

9.13.	Completed date addition	45
10.	Select Statement.....	46
10.1.	RolesOfUser.....	46
10.2.	UsersDetails	46
10.3.	ProjectDetails	47
10.4.	UserProject.....	48
10.5.	Milestones	49
10.6.	Tasks	50
10.7.	UserProjectTaskDetails	51
10.8.	Comments	52
10.9.	Resources	53
10.10.	ResourcesTask.....	54
10.11.	Subtask	55
10.12.	UserSubtask.....	56
11.	Forms	57
11.1.	Homepage.....	57
11.2.	Complex form and queries	58
11.2.1.	SQL Queries	58
11.2.2.	Complex forms	59
11.3.	Simple form	62
12.	User Manual.....	65
13.	Testing.....	76
13.1.	Users basic webform testing	76
13.2.	Projects basic web form testing	82
13.3.	Milestones basic web form testing	89
13.4.	Subtasks basic webform testing.....	95

13.5.	Tasks basic webform testing	101
13.6.	Top performer complex webform testing	107
13.7.	User and Project complex webform testing.....	113
13.8.	Project and milestones.....	116
13.9.	Data validation in ProjectDetails	119
13.10.	Invalid foreign key testing.....	123
14.	Further Discussion	126
15.	Conclusion	127
16.	References.....	128

Table of Figures

Figure 1: Textual analysis of user and project.....	2
Figure 2: Textual analysis of project and task.....	2
Figure 3: Textual analysis of user and task.....	2
Figure 4: Textual analysis of task and subtask	3
Figure 5: User, Comment and Task relationship.	3
Figure 6: Textual analysis of user and comment.....	3
Figure 7: Textual analysis of task and comment.....	3
Figure 8: Textual analysis of task and resource.....	4
Figure 9: Textual analysis of project and milestone	4
Figure 10: Textual analysis of user and subtask	4
Figure 11: Initial ERD	5
Figure 12: Final ERD	19
Figure 13: RolesOfUser data insertion	39
Figure 14: UsersDetails data insertion.....	39
Figure 15: ProjectDetails data insertion.....	40
Figure 16: UserProject data insertion	40
Figure 17: Milestones data insertion	41
Figure 18: Tasks data insertion	41
Figure 19: UserProjectTaskDetails data insertion	42
Figure 20: Comments data insertion.....	42
Figure 21: Resources data insertion	43
Figure 22: ResourcesTask data insertion	43
Figure 23: Subtask data insertion	44
Figure 24: UserSubtask data insertion	44
Figure 25: Completed date insertion.....	45
Figure 26: RolesOfUser select statement.....	46
Figure 27: UsersDetails select statement	47
Figure 28: ProjectDetails select statement	47
Figure 29: UserProject select statement.....	48
Figure 30: Milestones select statement	49

Figure 31: Tasks select statement	50
Figure 32: UserProjectTaskDetails select statement	51
Figure 33: Comments select statement	52
Figure 34: Resources select statement	53
Figure 35: ResourcesTask select statement.....	54
Figure 36: Subtask select statement	55
Figure 37: UserSubtask select statement.....	56
Figure 38: Homepage	57
Figure 39: Homepage (2).....	57
Figure 40: Top Performer Web.....	59
Figure 41: Top Performer Web (2)	60
Figure 42: User Project Web.....	60
Figure 43: User Project Web(2)	60
Figure 44: Project Milestone Web.....	61
Figure 45: Project Milestone Web (2)	61
Figure 46: User Basic Webform.....	62
Figure 47: Project Basic Webform	62
Figure 48: Milestone Basic Webform	63
Figure 49: Tasks Basic Webform	63
Figure 50: Subtasks Basic Webform.....	64
Figure 51: Navigation bar	65
Figure 52: User Manual (Dashboard)	65
Figure 53: Navigating to users page (User Manual)	66
Figure 54: Add new user button	66
Figure 55: Adding a user (User Manual).....	66
Figure 56: Editing User (User Manual)	67
Figure 57: Add new project button (User Manual)	68
Figure 58: Adding new project (User Manual)	68
Figure 59: Editing User (1) (User Manual).....	69
Figure 60: Editing User (2) (User Manual).....	69
Figure 61: Adding Milestone (1) (User Manual)	70

Figure 62: Adding Milestone (2) (User Manual)	70
Figure 63: Editing Milestone (1) (User Manual)	71
Figure 64: Editing Milestone (2) (User Manual)	71
Figure 65: Add Milestone (1) (User Manual)	72
Figure 66: Add milestone (2) (User Manual)	72
Figure 67: Editing Subtask (1) (User Manual)	73
Figure 68: Editing Subtask (2) (User Manual)	73
Figure 69: Add new task (1) (User Manual)	73
Figure 70: Add new task (2) (User Manual)	74
Figure 71: Edit Task (1) (User Manual)	74
Figure 72: Edit Task (2) (User Manual)	75
Figure 73: Adding a user (1)	76
Figure 74: Adding a user (2)	77
Figure 75: Adding a user (3)	77
Figure 76: Editing a user (1)	78
Figure 77: Editing a user (2)	79
Figure 78: Editing a user (3)	79
Figure 79: Deleting a user (1)	80
Figure 80: Deleting a user (2)	81
Figure 81: Adding a project (1)	83
Figure 82: Adding a project (2)	83
Figure 83: Adding a project (3)	84
Figure 84: Editing a project (1)	85
Figure 85: Editing a project (2)	86
Figure 86: Editing a project (3)	86
Figure 87: Deleting a project (1)	87
Figure 88: Deleting a project (2)	88
Figure 89: Adding a milestone (1)	90
Figure 90: Adding a milestone (2)	90
Figure 91: Adding a milestone (3)	90
Figure 92: Editing a milestone (1)	91

Figure 93: Editing a milestone (2).....	92
Figure 94: Editing a milestone (3).....	92
Figure 95: Deleting a milestone (1).....	93
Figure 96: Deleting a record (2).....	94
Figure 97: Adding a subtask (1).....	96
Figure 98: Adding a subtask (2).....	96
Figure 99: Adding a subtask (3).....	97
Figure 100: Editing a subtask (1).....	98
Figure 101: Editing a subtask (2).....	99
Figure 102: Editing a subtask (3).....	99
Figure 103: Deleting a subtask (1).....	100
Figure 104: Deleting a subtask (2).....	100
Figure 105: Add a task (1)	102
Figure 106: Add a task (2)	102
Figure 107: Add a task (3)	103
Figure 108: Editing a task (1).....	104
Figure 109: Editing a task (2).....	105
Figure 110: Editing a task (3)	105
Figure 111: Deleting a task (1).....	106
Figure 112: Deleting a task (2).....	106
Figure 113: Top Performer (1).....	107
Figure 114: Top performer (2)	108
Figure 115: Top performer (3)	108
Figure 116: Top performer (4)	108
Figure 117: Top Performer (3).....	110
Figure 118: Top Performer (4).....	110
Figure 119: Top Performer (5).....	111
Figure 120: Top performer (6).....	112
Figure 121: User and project (1).....	113
Figure 122: User and project (2).....	114
Figure 123: User and project (3).....	114

Figure 124: User and project (4).....	115
Figure 125: Project and milestones (1).....	116
Figure 126: Project and milestones (2).....	117
Figure 127: Project and milestones (3).....	117
Figure 128: Project and milestones (4).....	118
Figure 129: Incorrect data type entry (1)	120
Figure 130: Incorrect data type entry (2)	120
Figure 131: Data validation in ProjectDetails.....	122
Figure 132: Milestones Details Improper foreign key (1)	124
Figure 133: Milestones Details Improper foreign key (2)	124
Figure 134: Project name drop down.....	125

Table of tables

Table 1: RolesOfUser Data Dictionary	20
Table 2: UsersDetails data dictionary	21
Table 3: ProjectDetails data dictionary	22
Table 4: UserProject data dictionary	23
Table 5: Milestones data dictionary	24
Table 6: Tasks data dictionary	25
Table 7: UserProjectTaskDetails data dictionary	26
Table 8: Comments data dictionary	27
Table 9: Subtasks data dictionary	28
Table 10: UserSubtask data dictionary	29
Table 11: Resources data dictionary	29
Table 12: ResourcesTask data dictionary	30
Table 13: Adding a user (test)	76
Table 14: Editing a user (test)	78
Table 15: Deleting a user (test)	80
Table 16: Adding a project (Test)	82
Table 17: Editing a project (test)	85
Table 18: Deleting a project (Test)	87
Table 19: Adding a milestone (Test)	89
Table 20: Editing a milestone (Test)	91
Table 21: Deleting a milestone (Test)	93
Table 22: Adding a subtask (Test)	95
Table 23: Editing a subtask (Test)	98
Table 24: Deleting a subtask (Test)	100
Table 25: Adding a task (Test)	101
Table 26: Editing a task (Testing)	104
Table 27: Deleting a task (Testing)	106
Table 28: Top performer (Test Failure)	107
Table 29: Top performer (Testing)	109
Table 30: User and Project (Testing)	113

Table 31: Project and milestones (Test)	116
Table 32: Data Validation in ProjectDetails (Test Failure)	119
Table 33: Data Validation in ProjectDetails (Testing)	121
Table 34: Invalid foreign key testing (failure)	123
Table 35: Milestones Details foreign key handling (Testing)	125

1. Introduction

LS Corporation is a technology firm that manages multiple projects simultaneously. With many users working on one or more projects, which are further broken down into several tasks, the company has had to face issues with proper management of the tracking system for the projects and users involved in them. To solve this issue, a system was built from scratch, from designing the database to implementing it with the ASPX-based webform for performing CRUD operations.

A rough Entity-Relationship diagram was made, which was then followed by the normalization of a data record of a user involved in several projects, tasks, and milestones. A detailed breakdown of the development of the application will be discussed throughout the report, including the assumptions made while designing the database.

2. Textual Analysis

2.1. User and Project

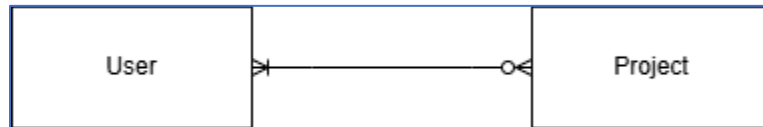


Figure 1: Textual analysis of user and project

Description: A user can be a part of zero or more projects at the same time. A single project needs to have at least one or more users. At least one active user is needed for a project.

2.2. Project and Task

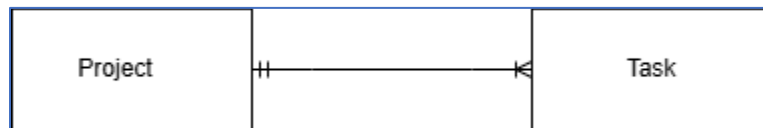


Figure 2: Textual analysis of project and task

Description: A project can have one or more tasks. However, a single task can only be assigned to a single project. Hence, a task cannot exist without a project and vice versa.

2.3. User and Task

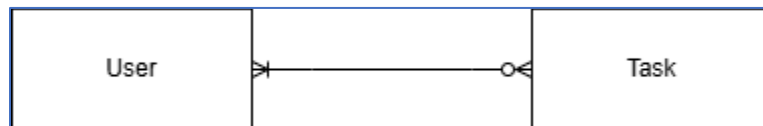


Figure 3: Textual analysis of user and task

Description: A user can be assigned to no, one, or more tasks. A task, however, can have multiple users, but at least one user needs to be involved with the task. Hence, a task without a user cannot exist.

2.4. Task and Subtask

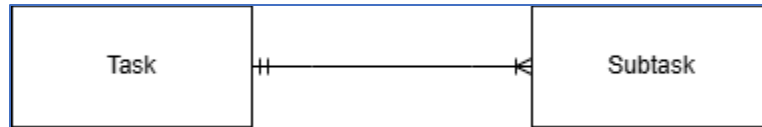


Figure 4: Textual analysis of task and subtask

Description: A single task needs to have at least one or more subtasks. A single subtask is only associated with one task. For a subtask to exist, there needs to be a task.

2.5. User, Comment, and Task



Figure 5: User, Comment and Task relationship.

2.5.1. User and Comment

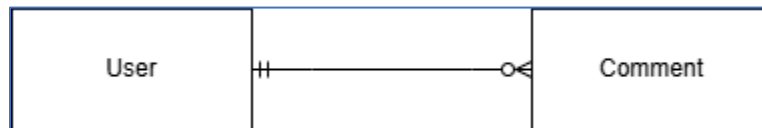


Figure 6: Textual analysis of user and comment

Description: A user can have zero or more comments; however, if a comment exists, it should be posted by a user.

2.5.2. Task and Comment

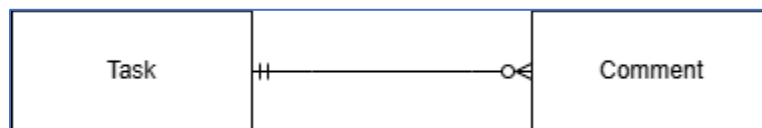


Figure 7: Textual analysis of task and comment

Description: A task can have multiple comments, but if there is a comment in a task, it can be linked to a certain task only.

2.6. Task and Resources

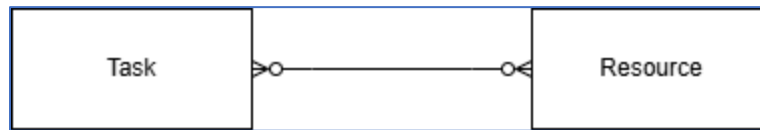


Figure 8: Textual analysis of task and resource

Description: A task can have zero or multiple resources according to the need. Similarly, a resource can also be assigned to multiple tasks.

2.7. Project and Milestones

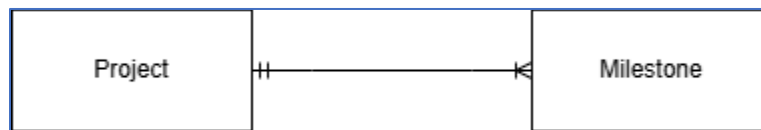


Figure 9: Textual analysis of project and milestone

Description: A project needs to have at least one milestone. A single milestone is only assigned to a single project.

2.8. User and subtask

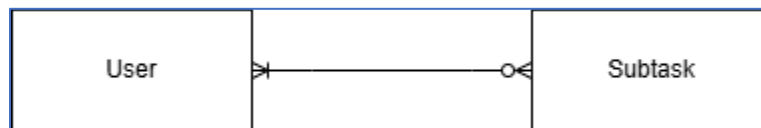


Figure 10: Textual analysis of user and subtask

Description: A user can be assigned to no or many subtasks. A subtask at least needs to have a single user.

3. Entity Relationship Diagram

An entity relationship diagram (ERD) is a special type of diagram that shows the relationship between tables in a database. Several types of symbols are used for representation to make the diagram as easy to understand as possible. (Belic & Stryker, 2024). An ERD defines the types of data in each entity (table in terms of database) along with their cardinality with other tables. This helps in understanding the nature of schema.

From the given description of LS Corporation, an ERD, in this case, an initial ERD, was prepared for rough visualization of the tables to be created in the database.

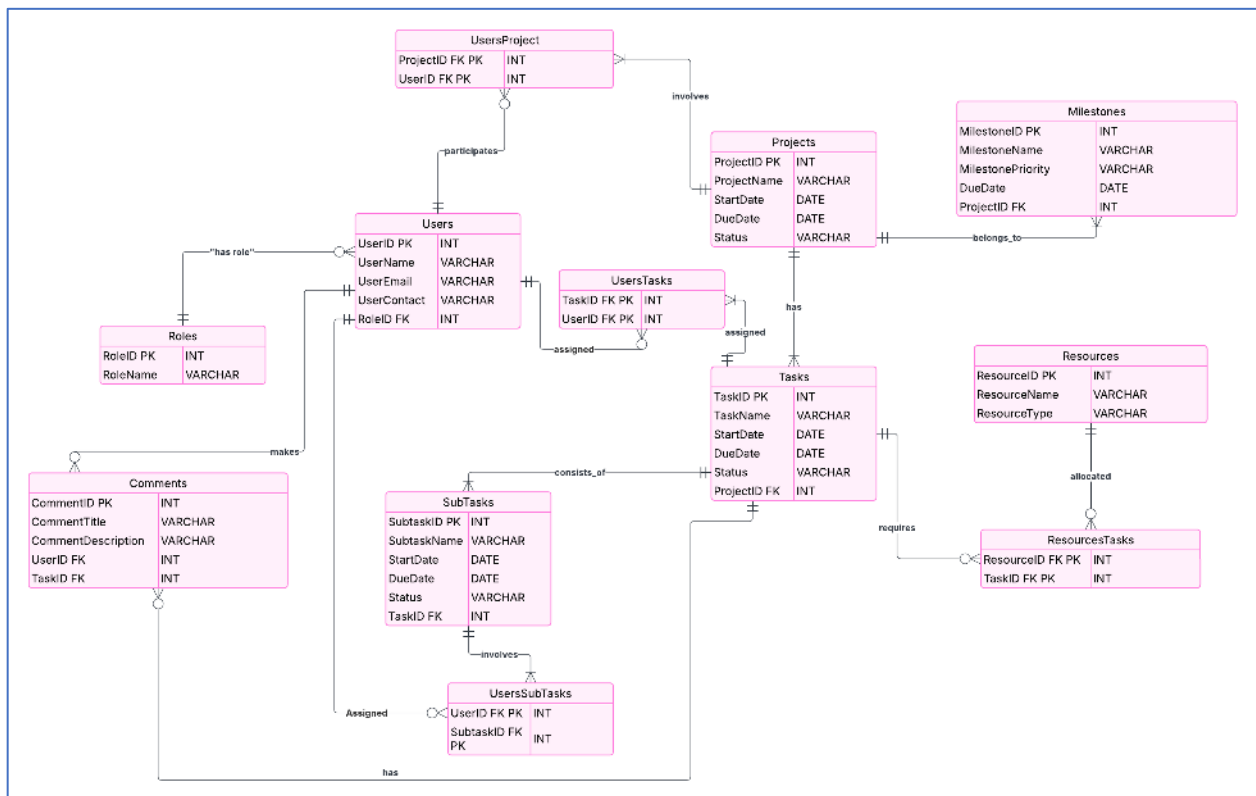


Figure 11: Initial ERD

This ERD shows the relationship between various entities with cardinality. However, this cannot be considered as final ERD, as many missing relationships are there between tables, or even some tables may be added, edited or deleted during the normalization process. The ERD was prepared entirely with the

description of data of LS Corporation. However, certain assumptions were made during the design process.

4. Normalization

Normalization is a step-by-step process that helps to maintain the data in a database in an organized way. Certain rules are followed that ensure the data are protected and minimize the redundancy (Chen, et al., 2024). Normalization is a crucial step required when designing a database. There are certain steps wherein each step, certain action is needed to perform which helps in organizing the database in an efficient way. The several steps involved in normalization are UNF, 1NF, 2NF, 3NF, 4NF, and 5NF, where NF stands for (Normal/ized form).

All the steps are not required for every database normalization. Looking at the complexity of the database, it can be decided up to which normal form is required by the system. In this case, looking at the nature and complexity of the data, 3NF would be enough for LS Corporation.

The normalization of the database is shown below:

4.1. UNF:

In this step, repeating groups are identified with curly braces.

User (UserID, UserName, UserEmail, UserContact,{ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus, {TaskID, TaskName, StartDate, DueDate, Status}})

4.2. 1NF

In this step of normalization, repeating groups are separated, where child groups are passed with the parent group's primary key.

User-1 (UserID, UserName, UserEmail, UserContact)

Project-1 (ProjectID,UserID*,ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

NOTE: Here, UserID and project ID are combined to become a composite primary key because there could be multiple instances of project and if only projectID is made as primary key there the primary key would be repeated.

Tasks-1 (TaskID, TaskName, StartDate, DueDate, Status, ProjectID*
UserID*)

NOTE: Here, UserID, project ID, and userID are combined to become a composite primary key because there could be multiple instances of tasks, and if only projectID is made a primary key, the primary key would be repeated.

Tables:

User-1 (UserID, UserName, UserEmail, UserContact)

Project-1 (ProjectID, UserID*, ProjectName, ProjectStartDate,
ProjectDueDate, ProjectStatus)

Tasks-1 (TaskID, TaskName, StartDate, DueDate, Status, ProjectID*
UserID*)

4.3. 2NF (Checking Partial Dependencies)

FOR User-1

This table is already in 2NF because there is no composite key to check the dependency.

Table:

User-2 (UserID, UserName, UserEmail, UserContact)

FOR Project-1 (ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus, UserID*)

ProjectID → ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus

UserID → X

ProjectID, UserID → X

The table with only UserID is removed as there are no dependent variables. However, ProjectID and UserID are composite keys and cannot be removed

Tables:

Project-Details-2 (ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

User-Project-2 (ProjectID*, UserID*)

FOR Tasks-1 (TaskID, TaskName, StartDate, DueDate, Status, ProjectID, UserID*)

TaskID → TaskName, StartDate, DueDate, Status

ProjectID → X

UserID → X

TaskID, ProjectID → X

TaskID, UserID → X

ProjectID, UserID → X

TaskID, ProjectID, UserID → X

Only composite keys without any dependencies are taken, and other keys without any dependencies are removed.

Tables:

Tasks-2 (TaskID, TaskName, StartDate, DueDate, Status)

User-Project-Task-2 (TaskID*,ProjectID*,UserID*)

Tables after 2NF:

User-2 (UserID, UserName, UserEmail, UserContact)

Project-Details-2 (ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

User-Project-2 (ProjectID*,UserID*)

Tasks-2 (TaskID, TaskName, StartDate, DueDate, Status)

User-Project-Task-2 (TaskID*,ProjectID*,UserID*)

4.4. 3NF (Checking for Transitive Dependencies)

FOR User-2 (UserID, UserName, UserEmail, UserContact)

UserID → UserName → X

UserID → UserEmail → X

UserID → UserContact → X

Assumption: It is assumed that UserEmail and usercontact can be changed.

Here, no transitive dependencies are found.

User-3 (UserID, UserName, UserEmail, UserContact)

FOR Project-Details-2 (ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

ProjectID → ProjectName → X

ProjectID → ProjectStartDate → X

ProjectID → ProjectDueDate → X

ProjectID → ProjectStatus → X

Assumption: Here, it is assumed that ProjectName can be repeated.

Hence, no transitive dependencies were found.

Project-Details-3 (ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

FOR User-Project-2 (ProjectID*, UserID*)

Since there are no non-key attributes, this is already in 3NF form.

User-Project-3 (ProjectID*, UserID*)

FOR Tasks-2 (TaskID, TaskName, StartDate, DueDate, Status)

TaskID → TaskName → X

TaskID → StartDate → X

TaskID → DueDate → X

TaskID → Status → X

Assumption: Here, it is assumed that TaskName can be repeated.

Hence, no transitive dependencies were found.

Tasks-3 (TaskID, TaskName, StartDate, DueDate, Status)

FOR UserProjectTask-2 (TaskID*, ProjectID*, UserID*)

Since there are no non-key attributes, this is already in 3NF form.

User-Project-Task -3 (TaskID*, ProjectID*, UserID*)

4.5. Final Tables

User (UserID, UserName, UserEmail, UserContact)

Project-Details(ProjectID, ProjectName, ProjectStartDate, ProjectDueDate, ProjectStatus)

User-Project(ProjectID*, UserID*)

Tasks(TaskID, TaskName, StartDate, DueDate, Status)

User-Project-Task (TaskID*, ProjectID*, UserID*)

5. Integration and Assumption

5.1. Integration

The following tables were formed after the normalization process from the data record of the user enrolled in the project.

- 1) User
- 2) User-Project
- 3) Project-Details
- 4) Tasks
- 5) User-Project-Task

Normalization is only accountable for the tables created with it. Other tables created with the company description are yet to be integrated. Certain changes also need to be made after the normalization.

In the case of the same tables in initial ERD and normalization formed, the tables formed from the normalization are prioritized as it created with certain protocol. Some table names were also changed because of reserved keywords. Tables formed after the integration of initial ERD with normalized table are as follows:

- 1) User
- 2) User-Project
- 3) Project-Details
- 4) Tasks
- 5) User-Project-Task
- 6) Roles
- 7) Comments
- 8) Subtasks
- 9) UserSubtask
- 10) Resources
- 11) ResourcesTask
- 12) Milestones

5.1.1. Eliminated tables

During the integration of the table from the initial ERD and normalization, some tables from the initial ERD were eliminated; however, other tables were added that were not formed in normalization.

The eliminated table is:

- 1) UserTask: After the formation of UserProjectTask, UserTask becomes redundant. Since every task is related to a project and every project is associated with at least one user, the task is automatically linked with the user and the project. Hence, the table UserTask was removed.

5.1.2. Added tables

The tables that were not provided in the record of the user and were later added to the normalized database.

- 1) Comments
- 2) Milestones
- 3) ResourcesTask (Bridge table of Task and resources)
- 4) Resources
- 5) Subtasks
- 6) UserSubtask (Bridge table of User and subtask)

5.2. Assumptions

Several assumptions were made during the designing of the database.

Assumptions made are listed below:

- 1) Each user is enrolled in zero or multiple projects. The user does not have to be enrolled in at least one project.
- 2) Similarly, each user can be enrolled in at least one task or no task. If a user is not active in any project, it automatically means they are not assigned to any tasks as well.
- 3) Likewise, a user can be enrolled in one or more subtasks if they are enrolled in any tasks; however, if not, the user is not enrolled in any subtasks.
- 4) For each user, a single role is defined. Roles are not changed according to the project.
- 5) A task can have zero or multiple comments. Likewise, a user can also post single or multiple comments on a task if the user wishes to.
- 6) For any task, multiple resources can be used as required by the task. If the wanted resource is available in stock, the resource can be used for the respective task.
- 7) In case of the project size, if the project is too small to have several tasks and subtasks, the project shall have itself as a task and subtask.
- 8) Some fields, like the completed date, were thought to be missing and were added later, which will also help in determining some important factors, like to know if the project, task, and subtask were completed before the due date or after.
- 9) User email address is more important than contact address. The contact address can be repeated. An email address is required from a business perspective for frequent communication and notifications.

6. Final Entity Relationship Diagram (ERD)

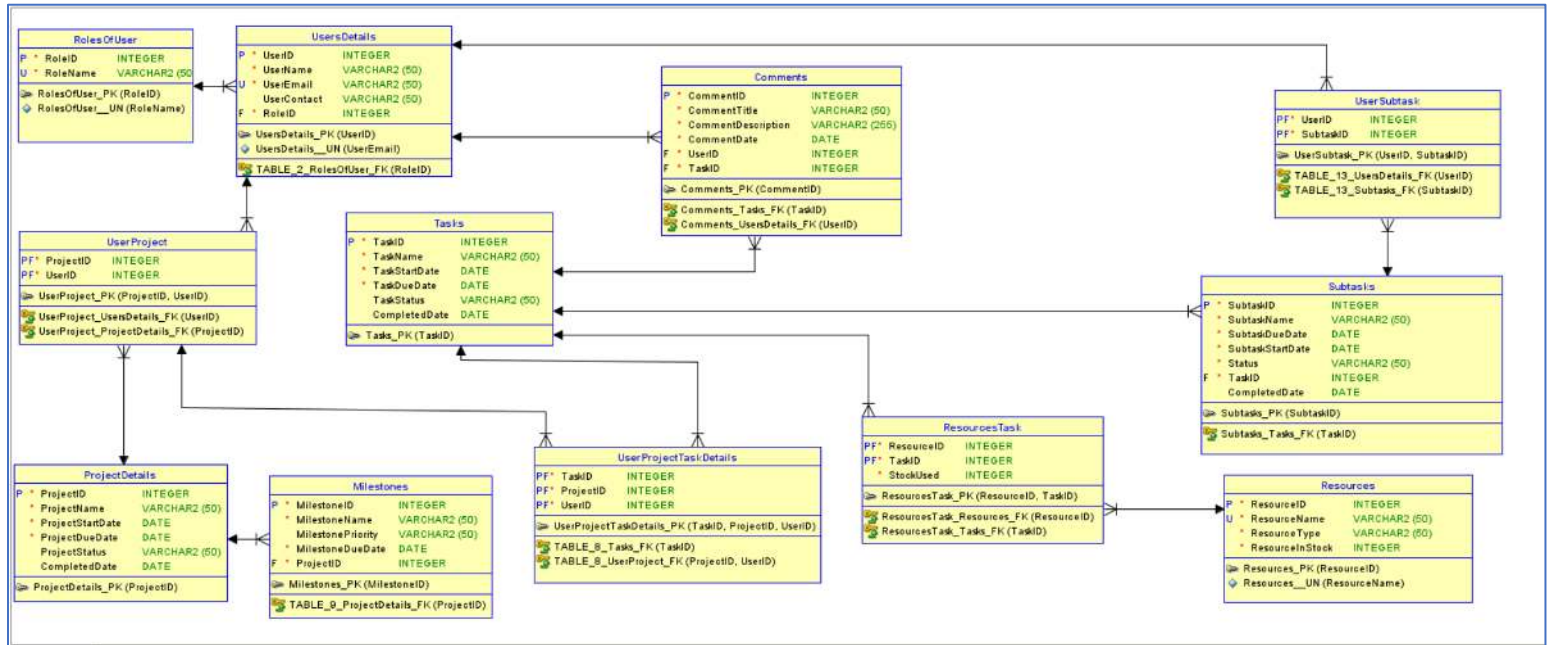


Figure 12: Final ERD

7. Data Dictionary

A dictionary is the description of data and attributes that defines the structure of a database or information system. A data dictionary helps to identify the nature of each table in the database and how and what types of data can be stored in it (UC MERCED LIBRARY, 2025).

A data dictionary helps in understanding the types of data that can go into the table, which assists in maintaining consistency throughout the table. The data dictionaries for all the tables created are as follows:

1) RolesOfUser

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
RoleID	Number	(38,0)	Primary Key			To identify the role	1230
RoleName	Varchar2	50	Not Null, Unique			To identify the role by name	'Manager'

Table 1: RolesOfUser Data Dictionary

2) UsersDetails

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
UserID	Integer	38	Primary Key			To identify the user by ID	35210
UserName	Varchar2	50	Not Null			To identify the user by name	'Rhythm'
UserEmail	Varchar2	50	Not Null, Unique			Stores user email address	'rhythm@example.com'
UserContact	Varchar2	50				Stores user contact number	'+1 123 432 132'
RoleID	Integer	38	Foreign Key, Not Null	RolesOfUser	RoleID	For identifying the role of the user	'Manager'

Table 2: UsersDetails data dictionary

3) ProjectDetails

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ProjectID	Integer	38	Primary Key			To identify the project by ID	1230
ProjectName	Varchar2	50	Not Null			To identify the project by name	'E-Commerce Platform'
ProjectStartDate	Date		Not Null			To know the start date of the project	'01-JAN-25'
ProjectDueDate	Date		Not null			To know the due date of project	'01-JAN-25'
ProjectStatus	Varchar2	50				To know the current project status.	'Completed'
CompletedDate	Date					To know the completed date of the project	'01-JAN-25'

Table 3: ProjectDetails data dictionary

4) UserProject

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ProjectID	Integer	38	Primary Key, Foreign Key	ProjectDetails	ProjectID	To identify associated projects through ID	1230
UserID	Integer	38	Primary Key, Foreign Key	UsersDetails	UserID	To identify associated users through ID	1240

Table 4: UserProject data dictionary

5) Milestones

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
MilestoneID	Integer	38	Primary Key			To identify the milestone ID	1230
MilestoneName	Varchar2	50	Not Null			To identify the milestone	'Prototype Completion'
MilestonePriority	Varchar2	50				To know the priority of the milestone	'High'
MilestoneDueDate	Date		Not Null			To know the due date of the milestone	'01-JAN-25'
ProjectID	Integer	38	Foreign Key, Not Null	ProjectDetails	ProjectID	To know which project the milestones belong to	1403

Table 5: Milestones data dictionary

6) Tasks

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
TaskID	Integer	38	Primary Key			To identify the task by its ID	1320043
TaskName	Varchar2	50	Not Null			To identify the task by its name	'API Documentation'
TaskStartDate	Date		Not Null			To know the start date of the task	'01-JAN-25'
TaskDueDate	Date		Not Null			To know the due date of the task	'01-JAN-25'
TaskStatus	Varchar2	50				To know the current status of the task	'Pending'
CompletedDate	Date					To know the completion date of the task	'01-JAN-25'

Table 6: Tasks data dictionary

7) UserProjectTaskDetails

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
TaskID	Integer	38	Primary Key, Foreign Key	Tasks	TaskID	To identify the task by its ID	1320043
ProjectID	Integer	38	Primary Key, Foreign Key	ProjectDetails	ProjectID	To identify the projectID by its ID	213312
UserID	Integer	38	Primary Key, Foreign Key	UsersDetails	UserID	To identify the user by their ID	12334

Table 7: UserProjectTaskDetails data dictionary

8) Comments

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
CommentID	Integer	38	Primary Key			To uniquely identify the comment	1320043
CommentTitle	Varchar2	50	Not Null			To know about what the comment is about.	'Error and bug'
CommentDescription	Varchar2	255	Not Null			A brief description of comments	'Need to fix it asap.'
CommentDate	Date		Not Null			To know when the comment was posted	'01-JAN-25'
UserID	Number	38	Not Null	UsersDetails	UserID	To know who posted the comment	3121
TaskID	Number	38	Not Null	Tasks	TaskID	To know which task is comment for	1231

Table 8: Comments data dictionary

9) Subtasks

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
SubtaskID	Integer	38	Primary Key			To identify the task by its ID	1320043
SubtaskName	Varchar2	50	Not Null			To identify the sub-task by its name	'API Documentation'
SubtaskStartDate	Date		Not Null			To know the start date of the sub-task	'01-JAN-25'
SubtaskDueDate	Date		Not Null			To know the due date of the sub-task	'01-JAN-25'
Status	Varchar2	50	Not Null			To know the current status of the sub-task	'Pending'
CompletedDate	Date					To know the completion date of the sub-task	'01-JAN-25'
TaskID	Integer	38	Foreign Key, Not Null	Tasks	TaskID	To know the associated task with subtask	12331

Table 9: Subtasks data dictionary

10) UserSubtask

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
UserID	Integer	38	Primary Key, Foreign Key	UsersDetails	UserID	To identify the user by its ID	1320043
SubtaskID	Intger	38	Primary Key, Foreign Key	Subtasks	SubtaskID	To identify the subtask by its ID	42123

Table 10: UserSubtask data dictionary

11) Resources

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ResourceID	Integer	38	Primary Key			To uniquely identify the resource by its ID	1320043
ResourceName	Varchar 2	50	Unique, Not null			To store the resource's name	'Laptop'
ResourceType	Varchar2	50	Not null			To store the category of the resource	'Hardware'
ResourceInStock	Integer	38	Not null			To know the in stock quantity	40

Table 11: Resources data dictionary

12) ResourcesTask

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ResourceID	Integer	38	Primary Key			To identify the resource by its ID	1320043
TaskID	Integer	38	Primary Key			To identify the task by its ID	1324054
StockUsed	Integer	38	Not Null			To know how many certain are used in certain task	24

Table 12: ResourcesTask data dictionary

8. Scripts

This section will cover the creation of the table according to the data dictionary defined in the previous section.

8.1. Roles of user

```
CREATE TABLE RolesOfUser
(
    RoleID    INTEGER NOT NULL ,
    RoleName  VARCHAR2 (50) NOT NULL
)
;

ALTER TABLE RolesOfUser
ADD CONSTRAINT RolesOfUser_PK PRIMARY KEY ( RoleID ) ;

ALTER TABLE RolesOfUser
ADD CONSTRAINT RolesOfUser__UN UNIQUE ( RoleName ) ;
```

8.2. UsersDetails

```
CREATE TABLE UsersDetails
(
    UserID      INTEGER NOT NULL ,
    UserName    VARCHAR2 (50) NOT NULL ,
    UserEmail    VARCHAR2 (50) NOT NULL ,
    UserContact VARCHAR2 (50) ,
    RoleID      INTEGER NOT NULL
)
;

ALTER TABLE UsersDetails
ADD CONSTRAINT UsersDetails_PK PRIMARY KEY ( UserID ) ;

ALTER TABLE UsersDetails
ADD CONSTRAINT UsersDetails__UN UNIQUE ( UserEmail ) ;

ALTER TABLE UsersDetails
ADD CONSTRAINT TABLE_2_RolesOfUser_FK FOREIGN KEY
(
    RoleID
)
REFERENCES RolesOfUser
(
    RoleID
)
;
```

8.3. ProjectDetails

```
CREATE TABLE ProjectDetails
(
    ProjectID          INTEGER NOT NULL ,
    ProjectName        VARCHAR2 (50) NOT NULL ,
    ProjectStartDate    DATE NOT NULL ,
    ProjectDueDate      DATE NOT NULL ,
    ProjectStatus       VARCHAR2 (50) ,
    CompletedDate       DATE
)
;

ALTER TABLE ProjectDetails
ADD CONSTRAINT ProjectDetails_PK PRIMARY KEY ( ProjectID ) ;
```

8.4. UserProject

```
CREATE TABLE UserProject
(
    ProjectID INTEGER NOT NULL ,
    UserID    INTEGER NOT NULL
)
;

ALTER TABLE UserProject
ADD CONSTRAINT UserProject_PK PRIMARY KEY ( ProjectID, UserID ) ;

ALTER TABLE UserProject
ADD CONSTRAINT UserProject_ProjectDetails_FK FOREIGN KEY
(
    ProjectID
)
REFERENCES ProjectDetails
(
    ProjectID
)
;

ALTER TABLE UserProject
ADD CONSTRAINT UserProject_UsersDetails_FK FOREIGN KEY
(
    UserID
)
REFERENCES UsersDetails
(
    UserID
)
;
```

8.5. Milestones

```
CREATE TABLE Milestones
(
    MilestoneID          INTEGER NOT NULL ,
    MilestoneName        VARCHAR2 (50) NOT NULL ,
    MilestonePriority     VARCHAR2 (50) ,
    MilestoneDueDate     DATE NOT NULL ,
    ProjectID            INTEGER NOT NULL
)
;

ALTER TABLE Milestones
    ADD CONSTRAINT Milestones_PK PRIMARY KEY ( MilestoneID ) ;

ALTER TABLE Milestones
    ADD CONSTRAINT TABLE_9_ProjectDetails_FK FOREIGN KEY
    (
        ProjectID
    )
    REFERENCES ProjectDetails
    (
        ProjectID
    )
;
;
```

8.6. Tasks

```
CREATE TABLE Tasks
(
    TaskID              INTEGER NOT NULL ,
    TaskName            VARCHAR2 (50) NOT NULL ,
    TaskStartDate       DATE NOT NULL ,
    TaskDueDate         DATE NOT NULL ,
    TaskStatus          VARCHAR2 (50) ,
    CompletedDate       DATE
)
;

ALTER TABLE Tasks
    ADD CONSTRAINT Tasks_PK PRIMARY KEY ( TaskID ) ;
```

8.7. UserProjectTaskDetails

```
CREATE TABLE UserProjectTaskDetails
(
    TaskID      INTEGER NOT NULL ,
    ProjectID   INTEGER NOT NULL ,
    UserID      INTEGER NOT NULL
)
;

ALTER TABLE UserProjectTaskDetails
ADD CONSTRAINT UserProjectTaskDetails_PK PRIMARY KEY ( TaskID,
ProjectID, UserID ) ;

ALTER TABLE UserProjectTaskDetails
ADD CONSTRAINT TABLE_8_Tasks_FK FOREIGN KEY
(
    TaskID
)
REFERENCES Tasks
(
    TaskID
)
;

ALTER TABLE UserProjectTaskDetails
ADD CONSTRAINT TABLE_8_UserProject_FK FOREIGN KEY
(
    ProjectID,
    UserID
)
REFERENCES UserProject
(
    ProjectID,
    UserID
)
;
```

8.8. Comments

```
CREATE TABLE Comments
(
    CommentID          INTEGER NOT NULL ,
    CommentTitle       VARCHAR2 (50) NOT NULL ,
    CommentDescription  VARCHAR2 (255) NOT NULL ,
    CommentDate        DATE NOT NULL ,
    UserID             INTEGER NOT NULL ,
    TaskID             INTEGER NOT NULL
)
;

COMMENT ON COLUMN Comments.CommentTitle IS ' '
;

ALTER TABLE Comments
    ADD CONSTRAINT Comments_PK PRIMARY KEY ( CommentID ) ;

ALTER TABLE Comments
    ADD CONSTRAINT Comments_Tasks_FK FOREIGN KEY
    (
        TaskID
    )
    REFERENCES Tasks
    (
        TaskID
    )
;

ALTER TABLE Comments
    ADD CONSTRAINT Comments_UsersDetails_FK FOREIGN KEY
    (
        UserID
    )
    REFERENCES UsersDetails
    (
        UserID
    )
;
```

8.9. Resources

```
CREATE TABLE Resources
(
    ResourceID          INTEGER NOT NULL ,
    ResourceName       VARCHAR2 (50) NOT NULL ,
    ResourceType       VARCHAR2 (50) NOT NULL ,
    ResourceInStock    INTEGER NOT NULL
)
;

ALTER TABLE Resources
    ADD CONSTRAINT Resources_PK PRIMARY KEY ( ResourceID ) ;
```

```
ALTER TABLE Resources
  ADD CONSTRAINT Resources__UN UNIQUE ( ResourceName ) ;
```

8.10. ResourcesTask

```
CREATE TABLE ResourcesTask
(
  ResourceID INTEGER NOT NULL ,
  TaskID     INTEGER NOT NULL ,
  StockUsed  INTEGER NOT NULL
)
;

ALTER TABLE ResourcesTask
  ADD CONSTRAINT ResourcesTask_PK PRIMARY KEY ( ResourceID,
  TaskID ) ;

ALTER TABLE ResourcesTask
  ADD CONSTRAINT ResourcesTask_Resources_FK FOREIGN KEY
  (
    ResourceID
  )
  REFERENCES Resources
  (
    ResourceID
  )
;

ALTER TABLE ResourcesTask
  ADD CONSTRAINT ResourcesTask_Tasks_FK FOREIGN KEY
  (
    TaskID
  )
  REFERENCES Tasks
  (
    TaskID
  )
;
```

8.11. Subtasks

```
CREATE TABLE Subtasks
(
  SubtaskID          INTEGER NOT NULL ,
  SubtaskName        VARCHAR2 (50) NOT NULL ,
  SubtaskDueDate     DATE NOT NULL ,
  SubtaskStartDate   DATE NOT NULL ,
  Status             VARCHAR2 (50) NOT NULL ,
  TaskID             INTEGER NOT NULL ,
  CompletedDate      DATE
)
;

ALTER TABLE Subtasks
  ADD CONSTRAINT Subtasks_PK PRIMARY KEY ( SubtaskID ) ;

ALTER TABLE Subtasks
  ADD CONSTRAINT Subtasks_Tasks_FK FOREIGN KEY
  (
    TaskID
  )
  REFERENCES Tasks
  (
    TaskID
  )
;
;
```

8.12. UserSubtask

```
CREATE TABLE UserSubtask
(
  UserID    INTEGER NOT NULL ,
  SubtaskID INTEGER NOT NULL
);

ALTER TABLE UserSubtask
  ADD CONSTRAINT UserSubtask_PK PRIMARY KEY ( UserID, SubtaskID
);

ALTER TABLE UserSubtask
  ADD CONSTRAINT TABLE_13_Subtasks_FK FOREIGN KEY
  (
    SubtaskID
  )
  REFERENCES Subtasks
  (
    SubtaskID
  )
;

ALTER TABLE UserSubtask
  ADD CONSTRAINT TABLE_13_UsersDetails_FK FOREIGN KEY
  (
    UserID
  )
  REFERENCES UsersDetails
  (
    UserID
  )
;
```

9. Data insertion

9.1. RolesOfUser

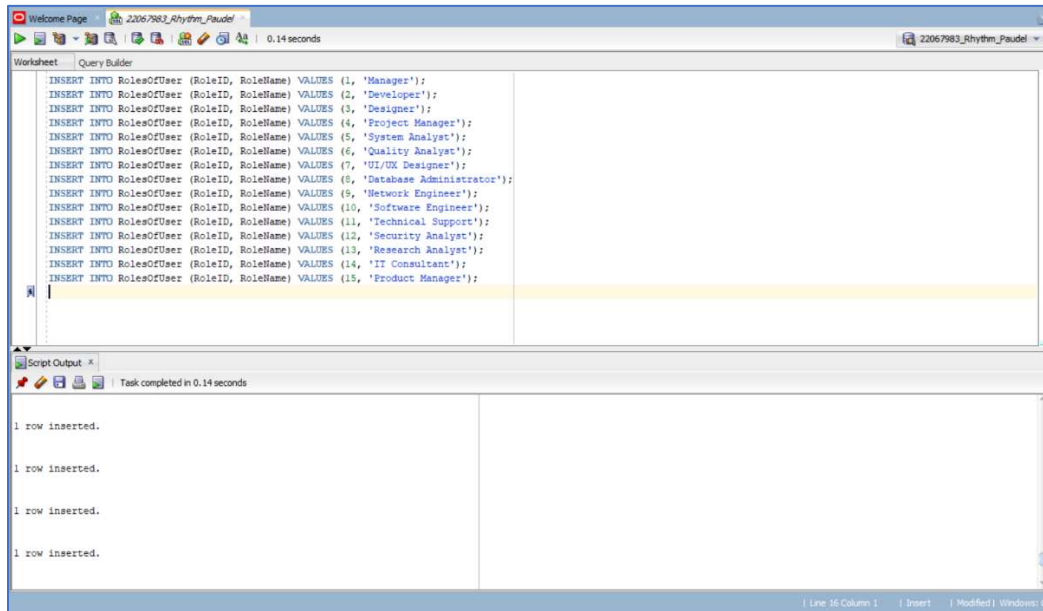


Figure 13: RolesOfUser data insertion

9.2. UsersDetails

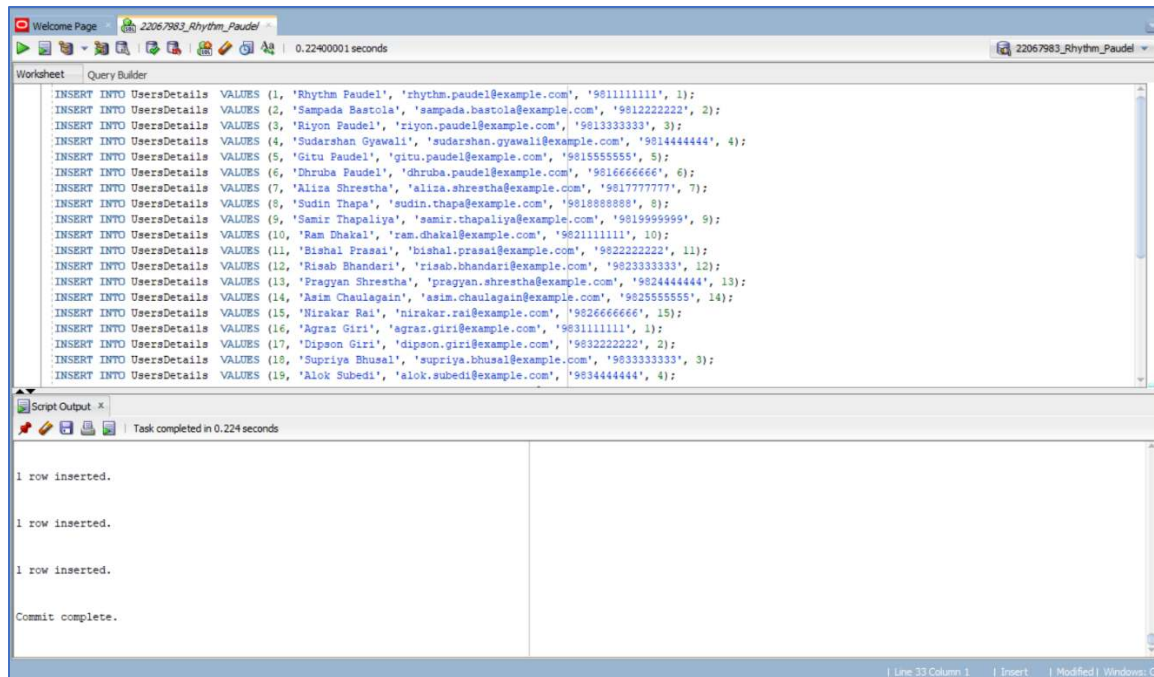


Figure 14: UsersDetails data insertion

9.3. ProjectDetails

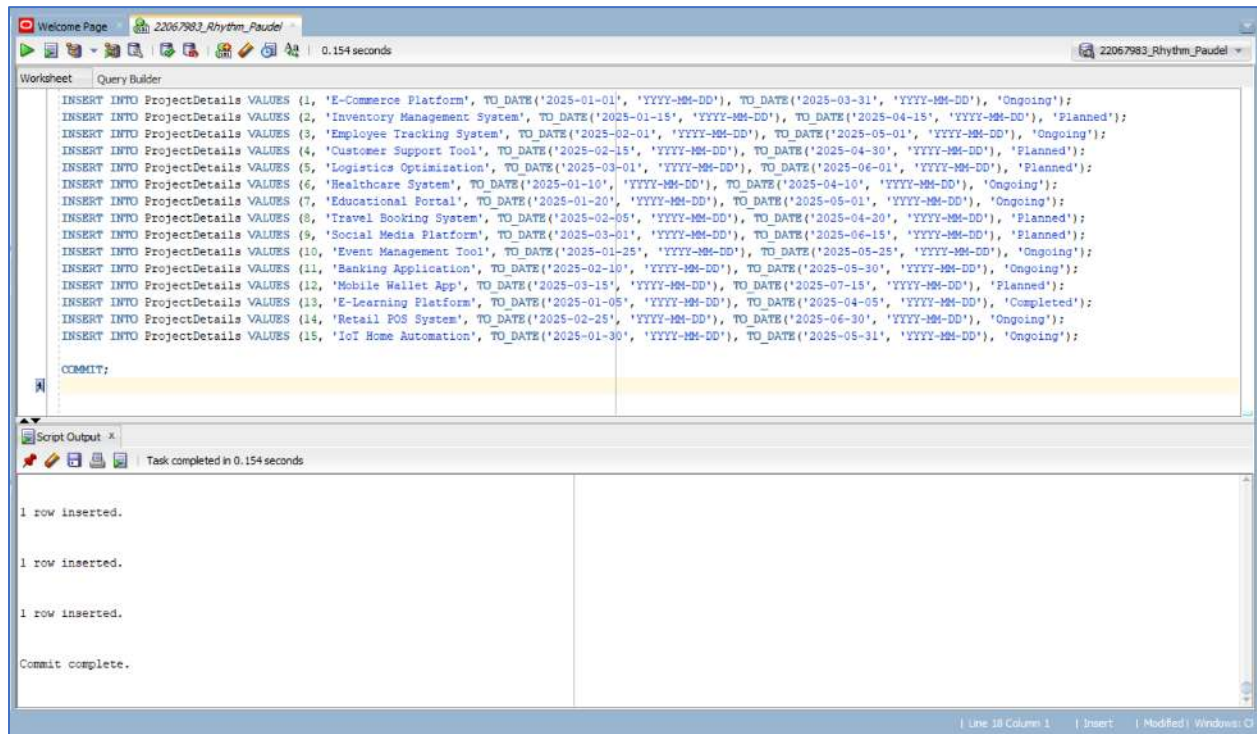


Figure 15: ProjectDetails data insertion

9.4. UserProject

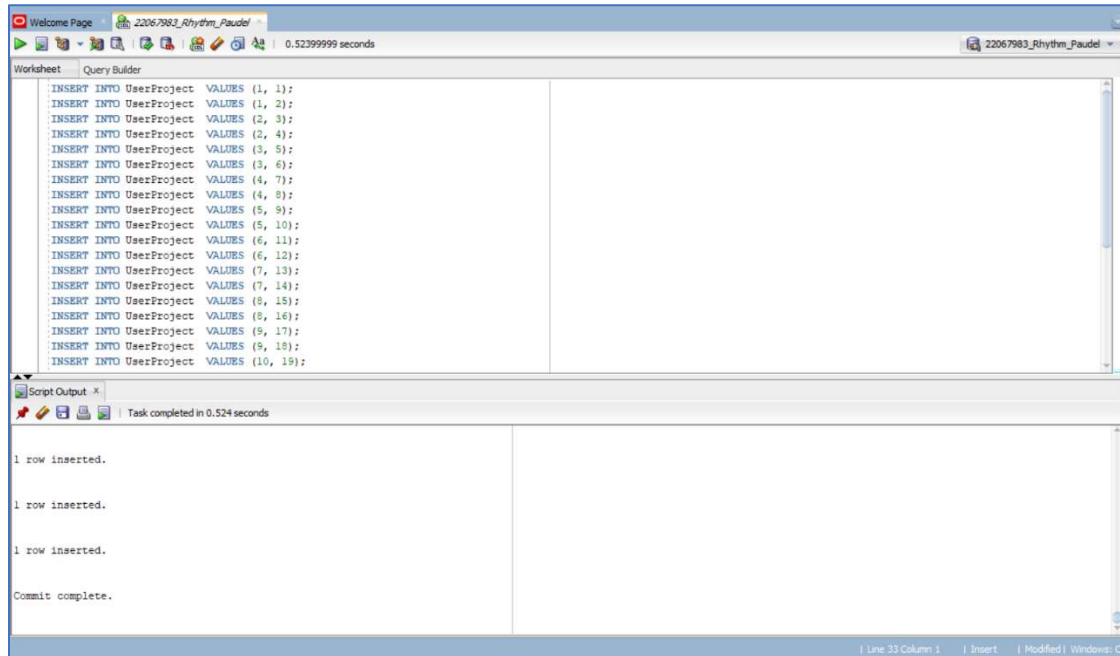


Figure 16: UserProject data insertion

9.5. Milestones

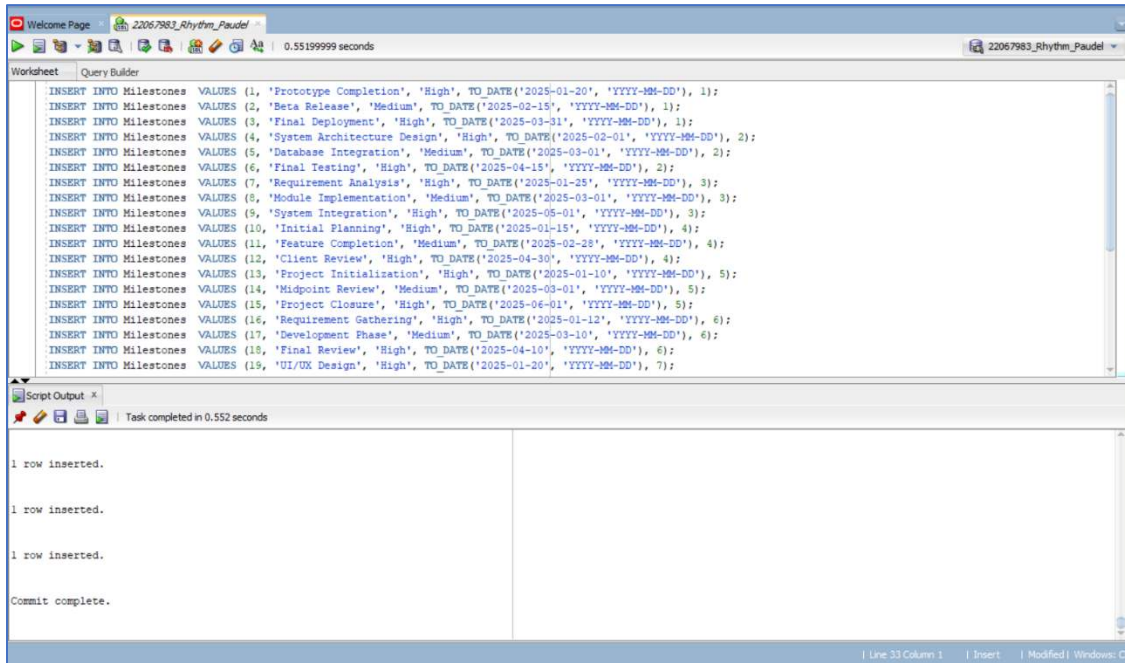


Figure 17: Milestones data insertion

9.6. Tasks

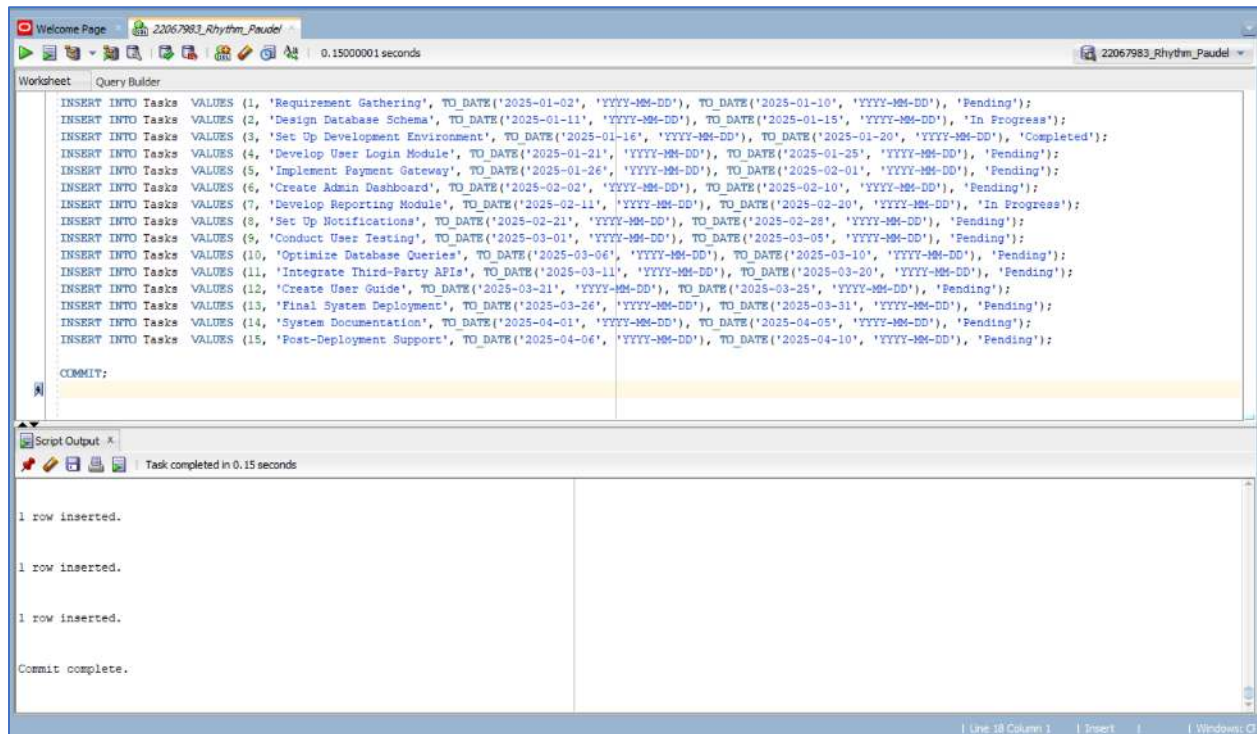


Figure 18: Tasks data insertion

9.7. UserProjectTaskDetails

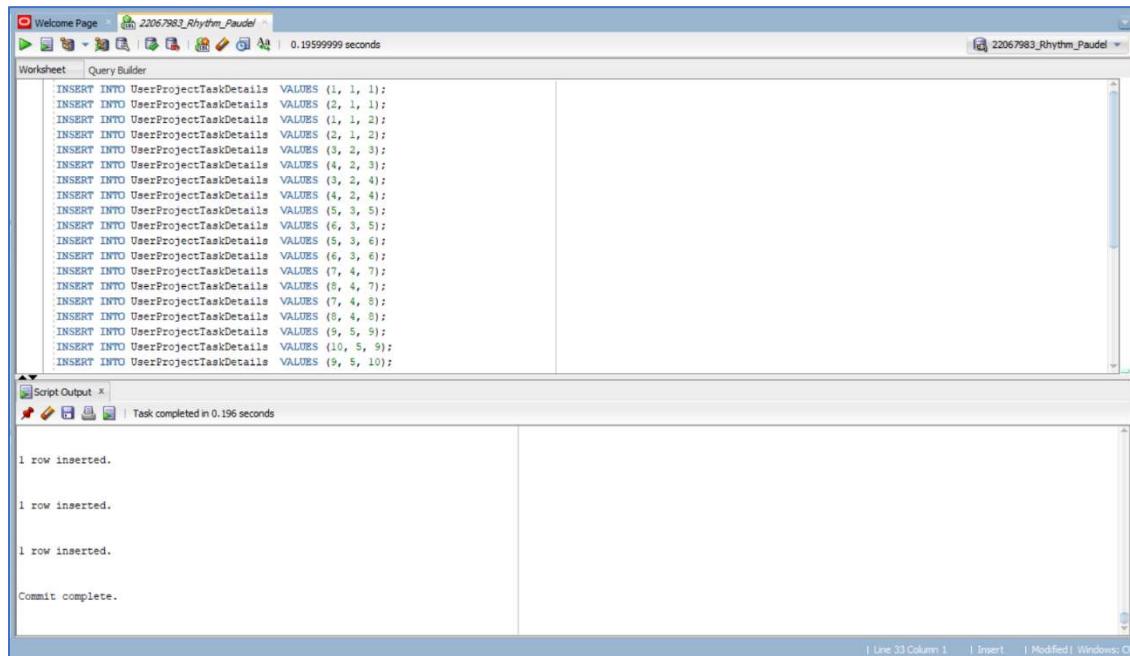


Figure 19: UserProjectTaskDetails data insertion

9.8. Comments

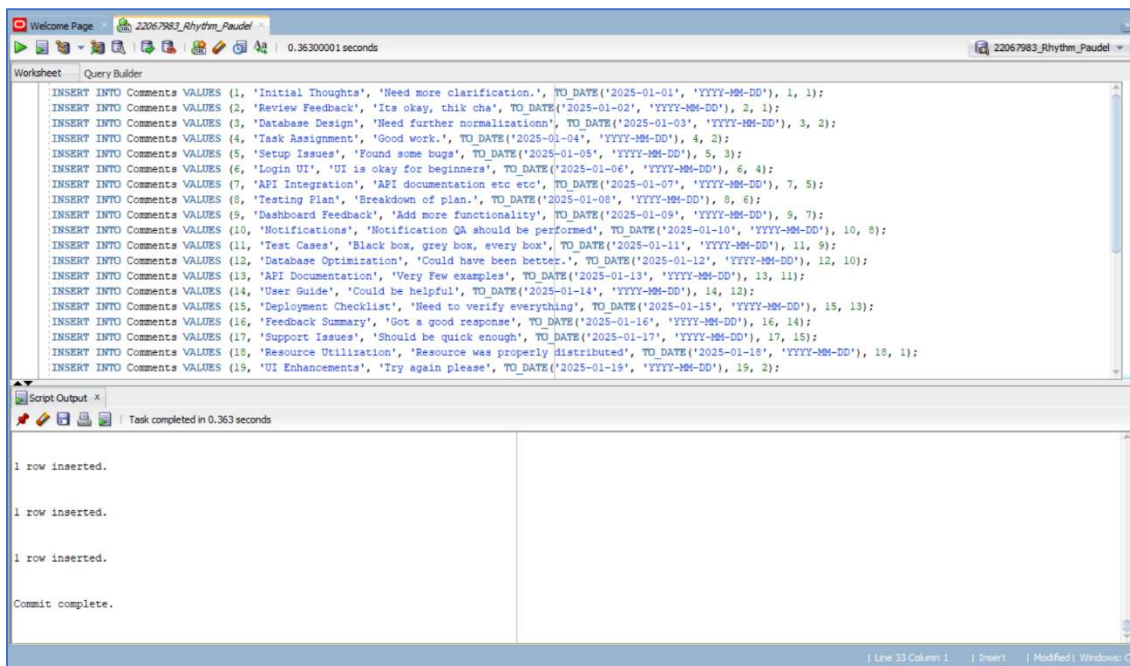


Figure 20: Comments data insertion

9.9. Resources

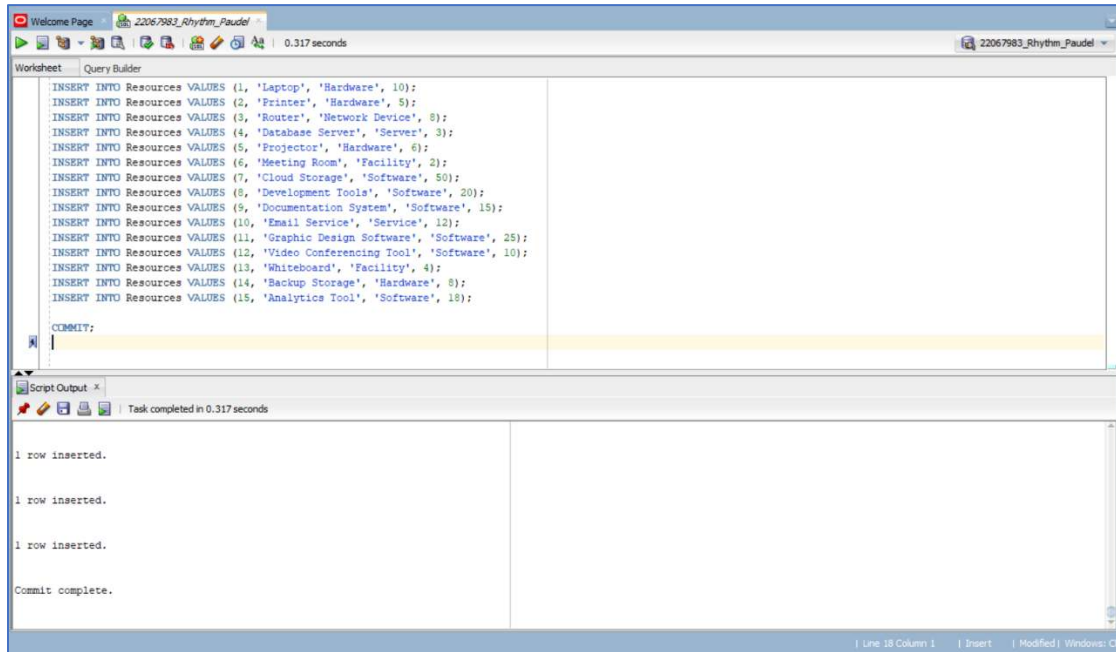


Figure 21: Resources data insertion

9.10. ResourcesTask

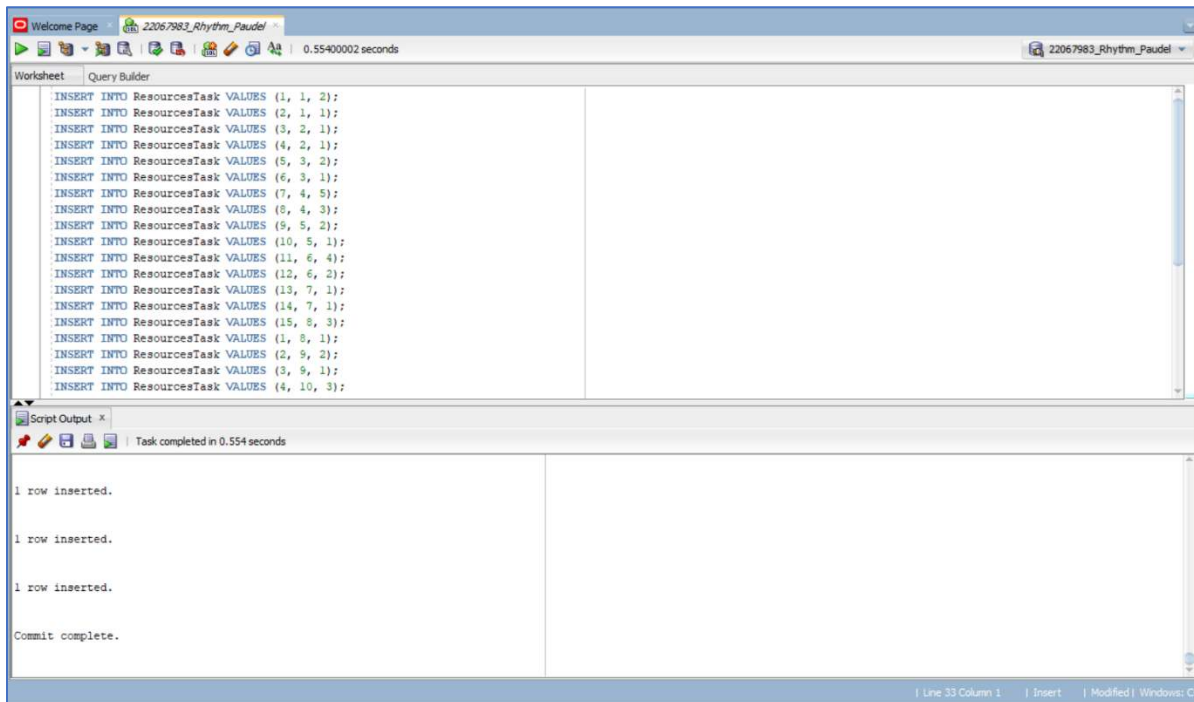


Figure 22: ResourcesTask data insertion

9.11. Subtask

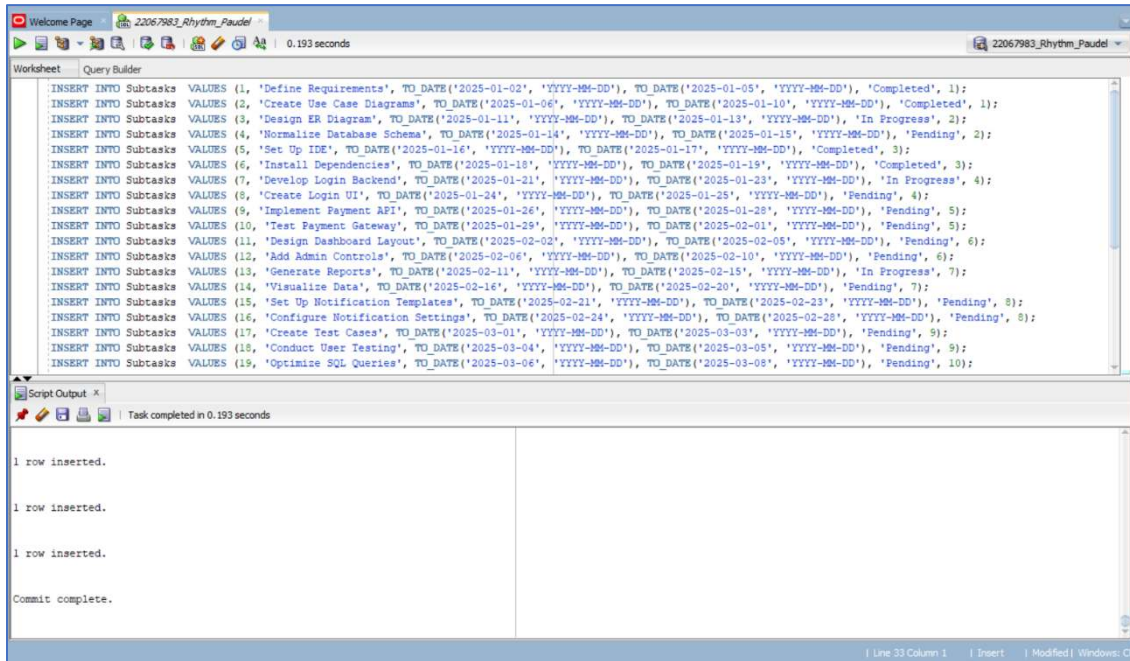


Figure 23: Subtask data insertion

9.12. UserSubtask

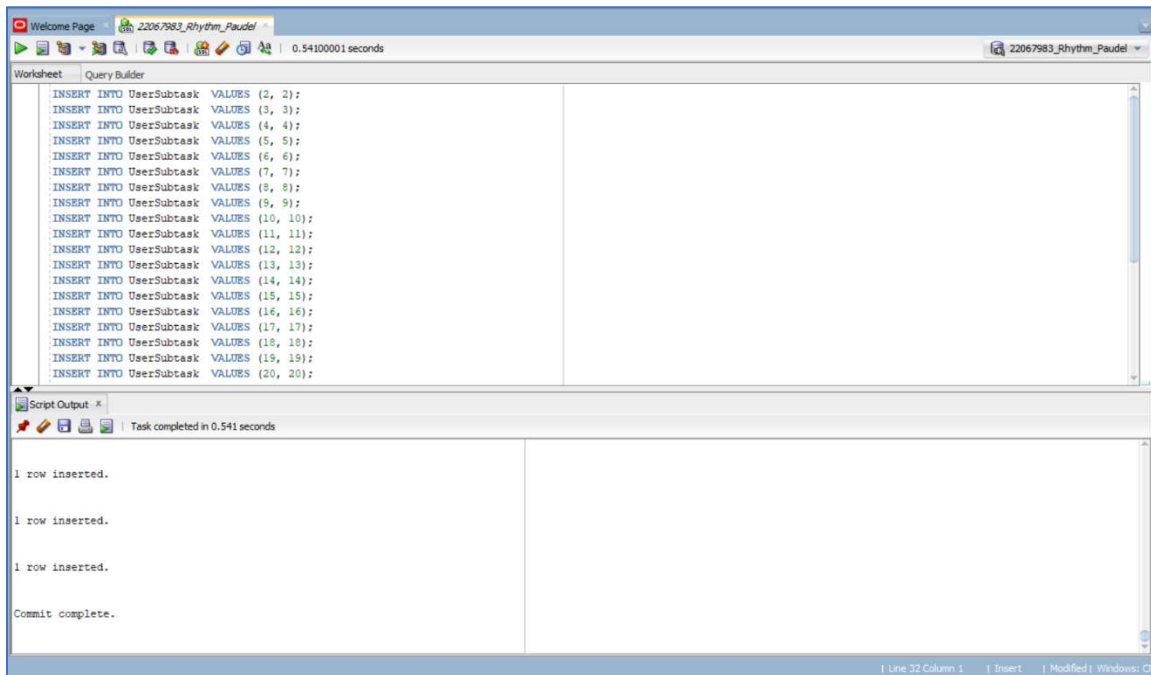


Figure 24: UserSubtask data insertion

9.13. Completed date addition

Since completedDate field was added later, the values were not inserted during the creation of projectdetails, tasks and subtasks tables.

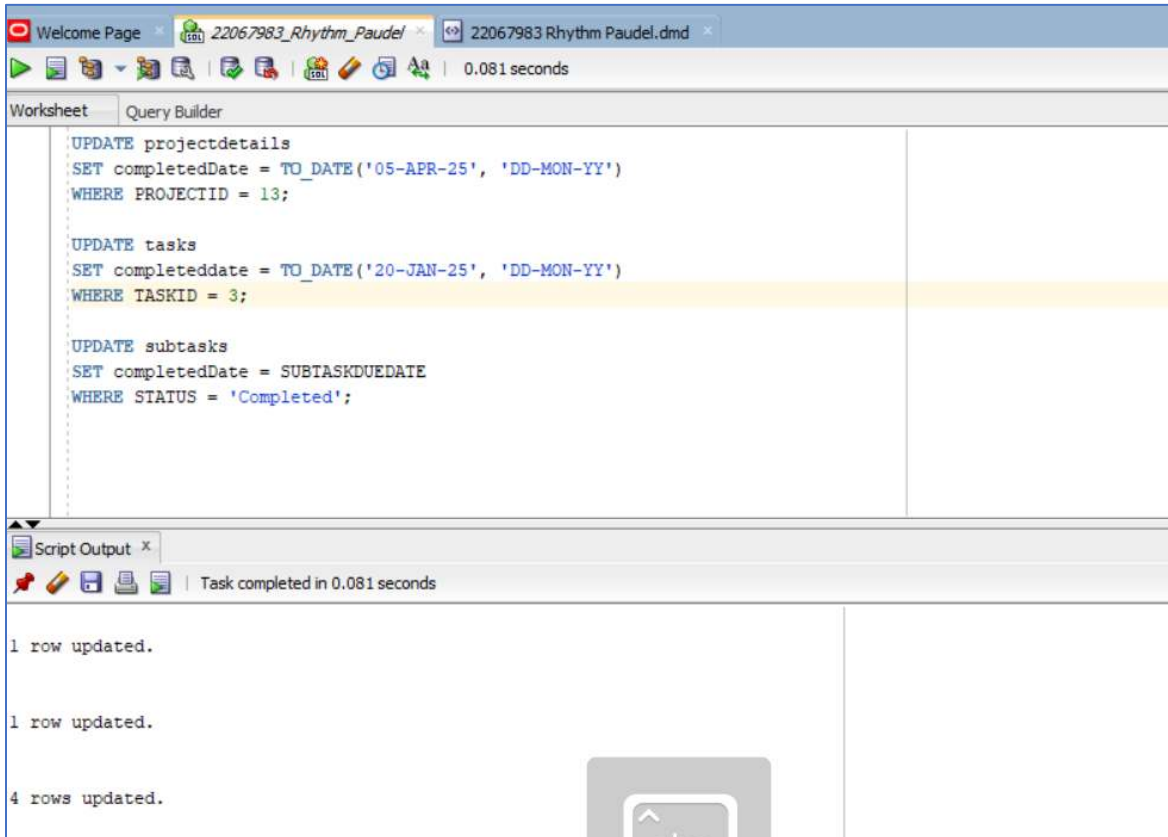


Figure 25: Completed date insertion

10. Select Statement

10.1. RolesOfUser

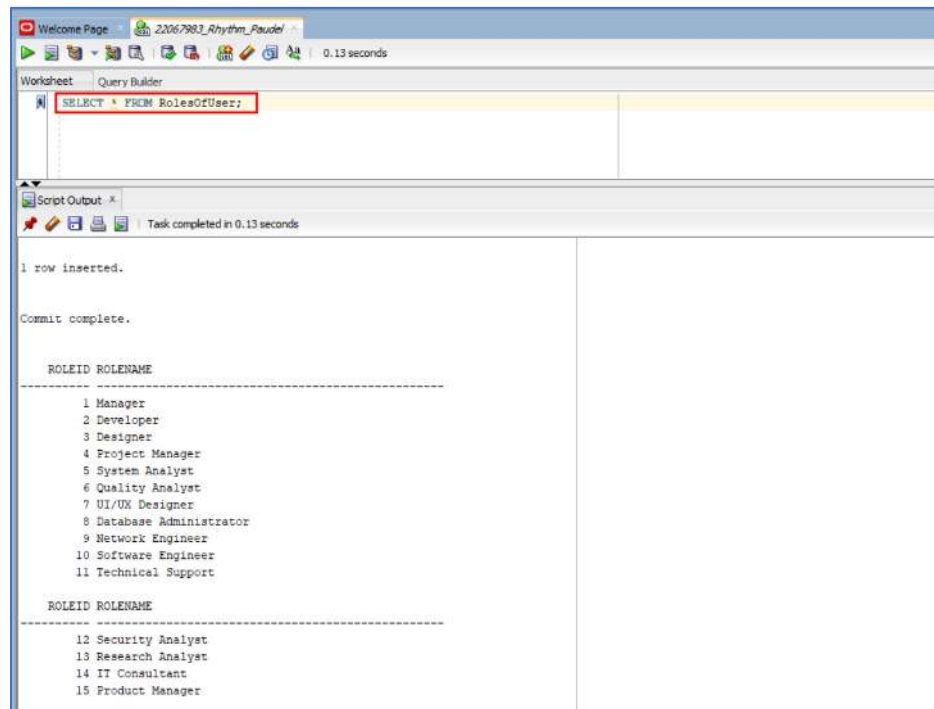


Figure 26: RolesOfUser select statement

10.2. UsersDetails

The screenshot shows a database query builder interface. The query entered is `SELECT * FROM UsersDetails;`. The output table displays 30 rows of user data.

USERID	USERNAME	USEREMAIL	USERCONTACT	ROLEID
12	Risab Bhandari	risab.bhandari@example.com	9823333333	12
13	Pragyan Shrestha	pragyan.shrestha@example.com	9824444444	13
14	Asim Chaulagain	asim.chaulagain@example.com	9825555555	14
15	Nirakar Rai	nirakar.raii@example.com	9826666666	15
16	Agraz Giri	agraz.giri@example.com	9831111111	1
17	Dipson Giri	dipson.giri@example.com	9832222222	2
18	Supriya Bhusal	supriya.bhusal@example.com	9833333333	3
19	Alok Subedi	alok.subedi@example.com	9834444444	4
20	Lekhnath Katuwal	lekhnath.katuwal@example.com	9835555555	5
21	Abhishek Khadka	abhishek.khadka@example.com	9836666666	6
22	Sulav Budhathoki	sulav.budhathoki@example.com	9837777777	7
23	Binod Chaudhary	binod.chaudhary@example.com	9838888888	8
24	Sisan Baniya	sisan.baniya@example.com	9839999999	9
25	Adarsha Lemichhane	adarsha.lemichhane@example.com	9841111111	10
26	Rajesh Hamal	rajesh.hamal@example.com	9842222222	11
27	KP Oli	kp.olii@example.com	9843333333	12
28	Lekhnath Adhikari	lekhnath.adhikari@example.com	9844444444	13
29	Sushil Koirala	sushil.koirala@example.com	9845555555	14
30	Sajit Shakya	sajit.shakya@example.com	9846666666	15

30 rows selected.

Figure 27: UsersDetails select statement

10.3. ProjectDetails

The screenshot shows a database query builder interface. The query entered is `select * from PROJECTDETAILS;`. The output table displays 16 rows of project data.

PROJECTID	PROJECTNAME	PROJECTSTARTDATE	PROJECTDUEDATE	PROJECTSTATUS	COMPLETEDDATE
1	1 E-Commerce Platform	01-JAN-25	31-MAR-25	Ongoing	(null)
2	2 Inventory Management System	15-JAN-25	15-APR-25	Planned	(null)
3	3 Employee Tracking System	01-FEB-25	01-MAY-25	Ongoing	(null)
4	4 Customer Support Tool	15-FEB-25	30-APR-25	Planned	(null)
5	5 Logistics Optimization	01-MAR-25	01-JUN-25	Planned	(null)
6	6 Healthcare System	10-JAN-25	10-APR-25	Ongoing	(null)
7	7 Educational Portal	20-JAN-25	01-MAY-25	Ongoing	(null)
8	8 Travel Booking System	05-FEB-25	20-APR-25	Planned	(null)
9	9 Social Media Platform	01-MAR-25	15-JUN-25	Planned	(null)
10	10 Event Management Tool	25-JAN-25	25-MAY-25	Ongoing	(null)
11	11 Banking Application	10-FEB-25	30-MAY-25	Ongoing	(null)
12	12 Mobile Wallet App	15-MAR-25	15-JUL-25	Planned	(null)
13	13 E-Learning Platform	05-JAN-25	05-APR-25	Completed	05-APR-25
14	14 Retail POS System	25-FEB-25	30-JUN-25	Ongoing	(null)
15	15 IoT Home Automation	30-JAN-25	31-MAY-25	Ongoing	(null)
16	16 Newly-Assigned-WebAdmin	01-JAN-25	31-MAR-25	Ongoing	(null)

Figure 28: ProjectDetails select statement

10.4. UserProject

The screenshot shows a database query tool interface. At the top, there's a 'Welcome Page' tab and a user profile '22067983_Rhythm_Paudel'. Below the toolbar, the 'Query Builder' tab is active, displaying the SQL statement: `SELECT * FROM UserProject;`. The 'Script Output' tab below shows the execution results, indicating 'Task completed in 0.035 seconds'. The results are presented in two tables, each with columns 'PROJECTID' and 'USERID'.

PROJECTID	USERID
6	12
7	13
7	14
8	15
8	16
9	17
9	18
10	19
10	20
11	21
11	22

PROJECTID	USERID
12	23
12	24
13	25
13	26
14	27
14	28
15	29
15	30

30 rows selected.

Figure 29: UserProject select statement

10.5. Milestones

Welcome Page | 22067983_Rhythm_Paudel | 0.059 seconds

Worksheet | Query Builder

SELECT * FROM milestones;

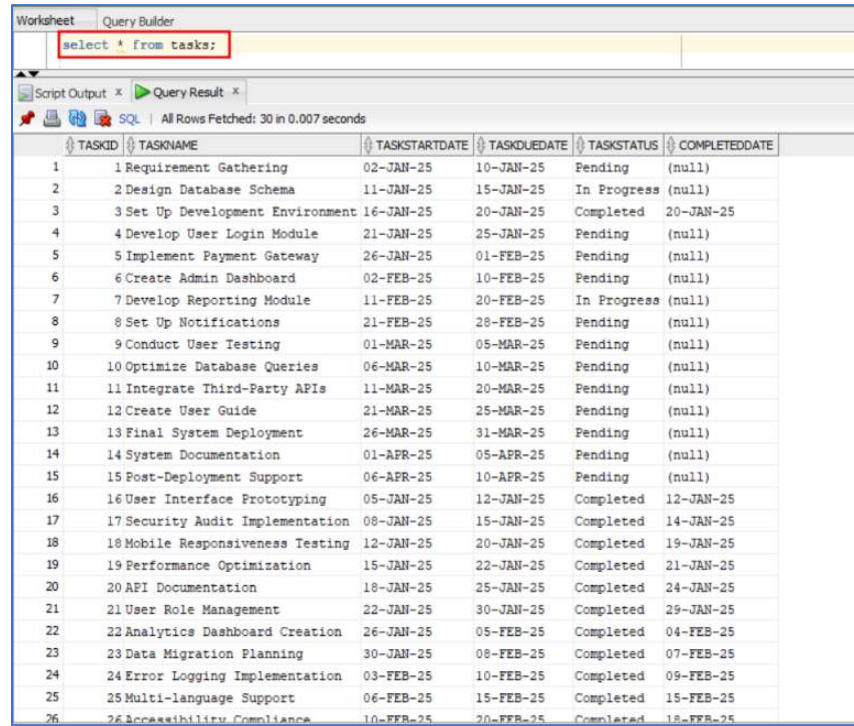
Script Output: X | Task completed in 0.059 seconds

MILESTONEID	MILESTONENAME	MILESTONEPRIORITY	MILESTONE	PROJECTID
12	Client Review	High	30-APR-25	4
13	Project Initialisation	High	10-JAN-25	5
14	Midpoint Review	Medium	01-MAR-25	5
15	Project Closure	High	01-JUN-25	5
16	Requirement Gathering	High	12-JAN-25	6
17	Development Phase	Medium	10-MAR-25	6
18	Final Review	High	10-APR-25	6
19	UI/UX Design	High	20-JAN-25	7
20	Backend Implementation	Medium	20-FEB-25	7
21	User Testing	High	01-MAY-25	7
22	Feature Planning	High	25-JAN-25	8
MILESTONEID	MILESTONENAME	MILESTONEPRIORITY	MILESTONE	PROJECTID
23	Feature Implementation	Medium	01-MAR-25	8
24	Beta Launch	High	20-APR-25	8
25	Requirements Finalization	High	01-FEB-25	9
26	Development	Medium	01-APR-25	9
27	Launch	High	15-JUN-25	9
28	Planning Phase	High	15-JAN-25	10
29	Development Completion	Medium	15-MAR-25	10
30	Final Deployment	High	25-MAY-25	10

30 rows selected.

Figure 30: Milestones select statement

10.6. Tasks

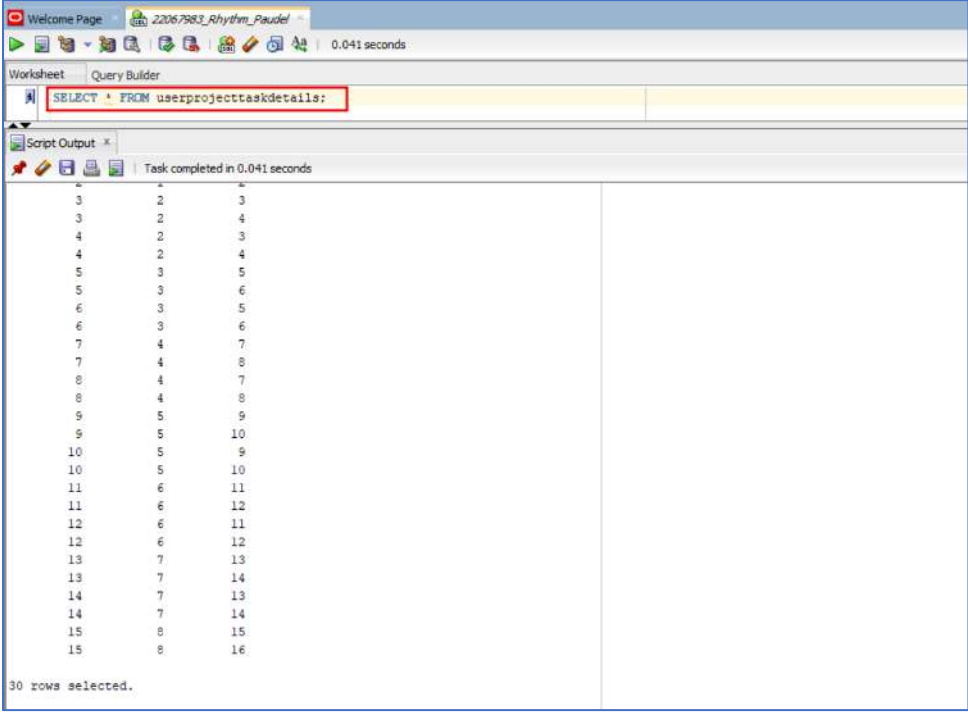


The screenshot shows a database query tool interface. At the top, there's a 'Query Builder' tab with a SQL query: `select * from tasks;`. Below the query, there's a 'Script Output' tab and a 'Query Result' tab. The 'Query Result' tab displays a table with 6 columns: TASKID, TASKNAME, TASKSTARTDATE, TASKDUEDATE, TASKSTATUS, and COMPLETEDDATE. The table contains 26 rows of task data, including details like task names (e.g., 'Requirement Gathering', 'Design Database Schema'), start and due dates, and status (e.g., 'Pending', 'In Progress', 'Completed').

TASKID	TASKNAME	TASKSTARTDATE	TASKDUEDATE	TASKSTATUS	COMPLETEDDATE
1	1 Requirement Gathering	02-JAN-25	10-JAN-25	Pending	(null)
2	2 Design Database Schema	11-JAN-25	15-JAN-25	In Progress	(null)
3	3 Set Up Development Environment	16-JAN-25	20-JAN-25	Completed	20-JAN-25
4	4 Develop User Login Module	21-JAN-25	25-JAN-25	Pending	(null)
5	5 Implement Payment Gateway	26-JAN-25	01-FEB-25	Pending	(null)
6	6 Create Admin Dashboard	02-FEB-25	10-FEB-25	Pending	(null)
7	7 Develop Reporting Module	11-FEB-25	20-FEB-25	In Progress	(null)
8	8 Set Up Notifications	21-FEB-25	28-FEB-25	Pending	(null)
9	9 Conduct User Testing	01-MAR-25	05-MAR-25	Pending	(null)
10	10 Optimize Database Queries	06-MAR-25	10-MAR-25	Pending	(null)
11	11 Integrate Third-Party APIs	11-MAR-25	20-MAR-25	Pending	(null)
12	12 Create User Guide	21-MAR-25	25-MAR-25	Pending	(null)
13	13 Final System Deployment	26-MAR-25	31-MAR-25	Pending	(null)
14	14 System Documentation	01-APR-25	05-APR-25	Pending	(null)
15	15 Post-Deployment Support	06-APR-25	10-APR-25	Pending	(null)
16	16 User Interface Prototyping	05-JAN-25	12-JAN-25	Completed	12-JAN-25
17	17 Security Audit Implementation	08-JAN-25	15-JAN-25	Completed	14-JAN-25
18	18 Mobile Responsiveness Testing	12-JAN-25	20-JAN-25	Completed	19-JAN-25
19	19 Performance Optimization	15-JAN-25	22-JAN-25	Completed	21-JAN-25
20	20 API Documentation	18-JAN-25	25-JAN-25	Completed	24-JAN-25
21	21 User Role Management	22-JAN-25	30-JAN-25	Completed	29-JAN-25
22	22 Analytics Dashboard Creation	26-JAN-25	05-FEB-25	Completed	04-FEB-25
23	23 Data Migration Planning	30-JAN-25	08-FEB-25	Completed	07-FEB-25
24	24 Error Logging Implementation	03-FEB-25	10-FEB-25	Completed	09-FEB-25
25	25 Multi-language Support	06-FEB-25	15-FEB-25	Completed	15-FEB-25
26	26 Accessibility Compliance	10-FEB-25	20-FEB-25	Completed	18-FEB-25

Figure 31: Tasks select statement

10.7. UserProjectTaskDetails

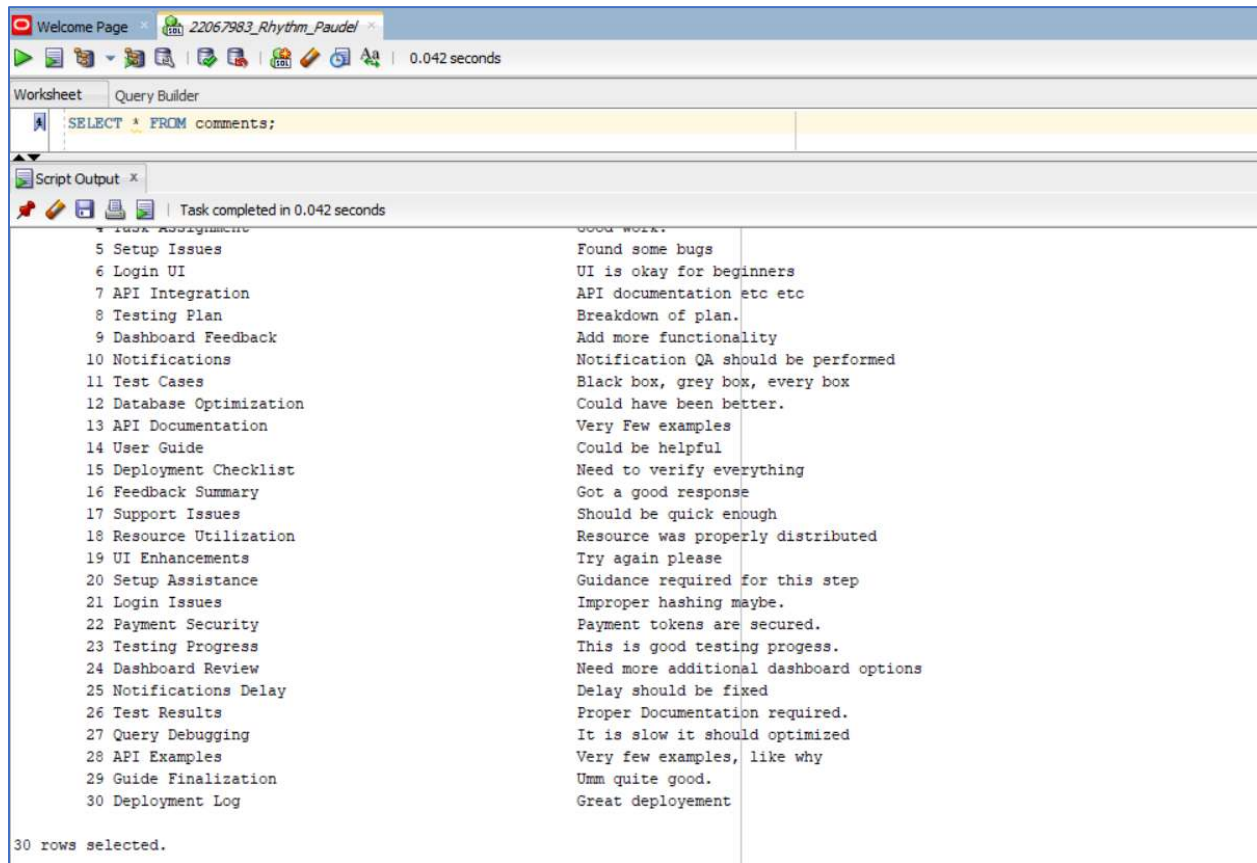


The screenshot shows a software interface with a 'Query Builder' tab. The SQL query entered is `SELECT * FROM userprojecttaskdetails;`. Below the query, the 'Script Output' tab displays the results of the query, which is a table with 30 rows. The table has three columns, each containing numerical values. The status bar at the bottom indicates '30 rows selected.'

3	2	3
3	2	4
4	2	3
4	2	4
5	3	5
5	3	6
6	3	5
6	3	6
7	4	7
7	4	8
8	4	7
8	4	8
9	5	9
9	5	10
10	5	9
10	5	10
11	6	11
11	6	12
12	6	11
12	6	12
13	7	13
13	7	14
14	7	13
14	7	14
15	8	15
15	8	16

Figure 32: UserProjectTaskDetails select statement

10.8. Comments



The screenshot shows a software interface with a top toolbar, a 'Worksheet' tab, and a 'Query Builder' section. The query entered is 'SELECT * FROM comments;'. Below this, a 'Script Output' section shows 'Task completed in 0.042 seconds'. The main area displays a table with 30 rows of data. The first column lists task names, and the second column contains corresponding comments. At the bottom left, it says '30 rows selected.'

Task	Comments
4 Task Assignment	Good work.
5 Setup Issues	Found some bugs
6 Login UI	UI is okay for beginners
7 API Integration	API documentation etc etc
8 Testing Plan	Breakdown of plan.
9 Dashboard Feedback	Add more functionality
10 Notifications	Notification QA should be performed
11 Test Cases	Black box, grey box, every box
12 Database Optimization	Could have been better.
13 API Documentation	Very Few examples
14 User Guide	Could be helpful
15 Deployment Checklist	Need to verify everything
16 Feedback Summary	Got a good response
17 Support Issues	Should be quick enough
18 Resource Utilization	Resource was properly distributed
19 UI Enhancements	Try again please
20 Setup Assistance	Guidance required for this step
21 Login Issues	Improper hashing maybe.
22 Payment Security	Payment tokens are secured.
23 Testing Progress	This is good testing progress.
24 Dashboard Review	Need more additional dashboard options
25 Notifications Delay	Delay should be fixed
26 Test Results	Proper Documentation required.
27 Query Debugging	It is slow it should optimized
28 API Examples	Very few examples, like why
29 Guide Finalization	Umm quite good.
30 Deployment Log	Great deployment

Figure 33: Comments select statement

10.9. Resources

Worksheet | Query Builder

SELECT * FROM resources;

Script Output x | Task completed in 0.032 seconds

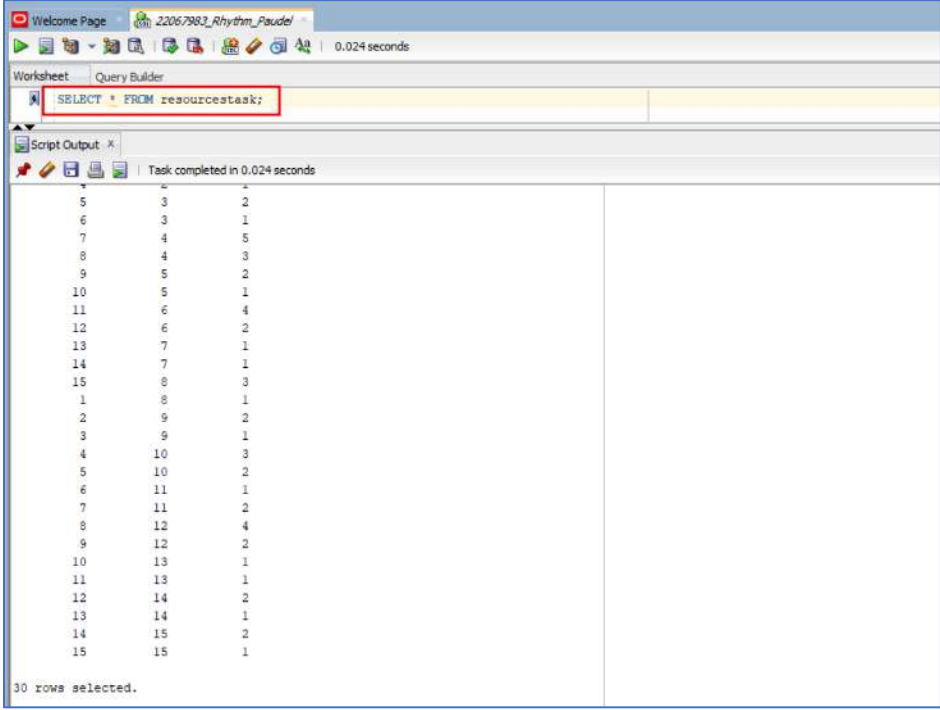
RESOURCEID	RESOURCENAME	RESOURCETYPE	RESOURCEINSTOCK
1	Laptop	Hardware	10
2	Printer	Hardware	5
3	Router	Network Device	8
4	Database Server	Server	3
5	Projector	Hardware	6
6	Meeting Room	Facility	2
7	Cloud Storage	Software	50
8	Development Tools	Software	20
9	Documentation System	Software	15
10	Email Service	Service	12
11	Graphic Design Software	Software	25
12	Video Conferencing Tool	Software	10
13	Whiteboard	Facility	4
14	Backup Storage	Hardware	8
15	Analytics Tool	Software	18

30 rows selected.

15 rows selected.

Figure 34: Resources select statement

10.10. ResourcesTask



The screenshot shows a software interface with a 'Query Builder' tab. The query entered is `SELECT * FROM resourceTask;`, which is highlighted with a red rectangle. Below the query, the 'Script Output' tab displays the results of the query. The output is a table with 30 rows and 3 columns. The first 15 rows are numbered 5 through 15, and the next 15 rows are numbered 1 through 15. The status bar at the bottom indicates '30 rows selected.'

5	3	2
6	3	1
7	4	5
8	4	3
9	5	2
10	5	1
11	6	4
12	6	2
13	7	1
14	7	1
15	8	3
1	8	1
2	9	2
3	9	1
4	10	3
5	10	2
6	11	1
7	11	2
8	12	4
9	12	2
10	13	1
11	13	1
12	14	2
13	14	1
14	15	2
15	15	1

Figure 35: ResourcesTask select statement

10.11. Subtask

Worksheet

Query Builder

select * from subtasks;

Script Output x

Query Result x

All Rows Fetched: 30 in 0.016 seconds

SUBTASKID	SUBTASKNAME	SUBTASKDUE DATE	SUBTASKSTARTDATE	STATUS	TASKID	COMPLETEDDATE
1	1 Define Requirements	02-JAN-25	05-JAN-25	Completed	1	02-JAN-25
2	2 Create Use Case Diagrams	06-JAN-25	10-JAN-25	Completed	1	06-JAN-25
3	3 Design ER Diagram	11-JAN-25	13-JAN-25	In Progress	2	(null)
4	4 Normalize Database Schema	14-JAN-25	15-JAN-25	Pending	2	(null)
5	5 Set Up IDE	16-JAN-25	17-JAN-25	Completed	3	16-JAN-25
6	6 Install Dependencies	18-JAN-25	19-JAN-25	Completed	3	18-JAN-25
7	7 Develop Login Backend	21-JAN-25	23-JAN-25	In Progress	4	(null)
8	8 Create Login UI	24-JAN-25	25-JAN-25	Pending	4	(null)
9	9 Implement Payment API	26-JAN-25	28-JAN-25	Pending	5	(null)
10	10 Test Payment Gateway	29-JAN-25	01-FEB-25	Pending	5	(null)
11	11 Design Dashboard Layout	02-FEB-25	05-FEB-25	Pending	6	(null)
12	12 Add Admin Controls	06-FEB-25	10-FEB-25	Pending	6	(null)
13	13 Generate Reports	11-FEB-25	15-FEB-25	In Progress	7	(null)
14	14 Visualize Data	16-FEB-25	20-FEB-25	Pending	7	(null)
15	15 Set Up Notification Templates	21-FEB-25	23-FEB-25	Pending	8	(null)
16	16 Configure Notification Settings	24-FEB-25	28-FEB-25	Pending	8	(null)
17	17 Create Test Cases	01-MAR-25	03-MAR-25	Pending	9	(null)
18	18 Conduct User Testing	04-MAR-25	05-MAR-25	Pending	9	(null)
19	19 Optimize SQL Queries	06-MAR-25	08-MAR-25	Pending	10	(null)
20	20 Test Query Performance	09-MAR-25	10-MAR-25	Pending	10	(null)
21	21 Add API Endpoints	11-MAR-25	15-MAR-25	Pending	11	(null)
22	22 Integrate External APIs	16-MAR-25	20-MAR-25	Pending	11	(null)
23	23 Write User Documentation	21-MAR-25	23-MAR-25	Pending	12	(null)
24	24 Format User Guide	24-MAR-25	25-MAR-25	Pending	12	(null)
25	25 Deploy Final System	26-MAR-25	30-MAR-25	Pending	13	(null)
26	26 Monitor Deployment	31-MAR-25	31-MAR-25	Pending	13	(null)

Figure 36: Subtask select statement

10.12. UserSubtask

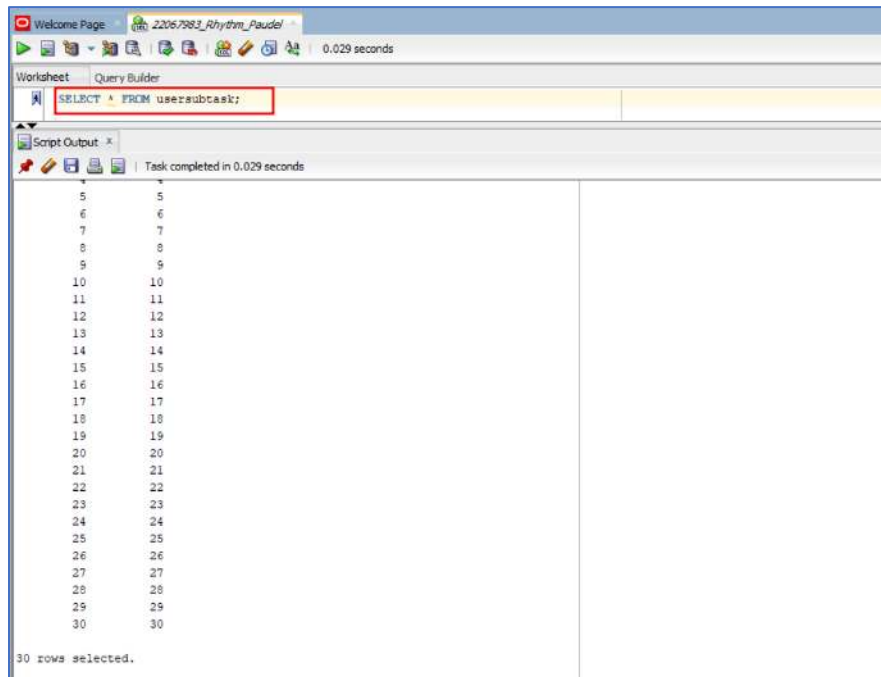


Figure 37: UserSubtask select statement

11. Forms

11.1. Homepage

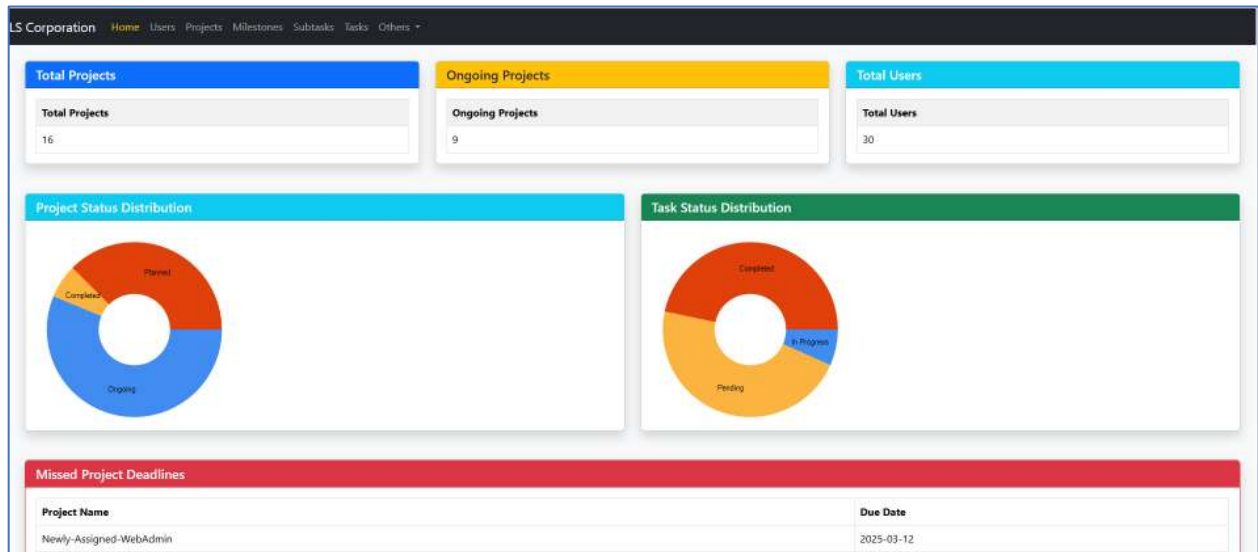


Figure 38: Homepage

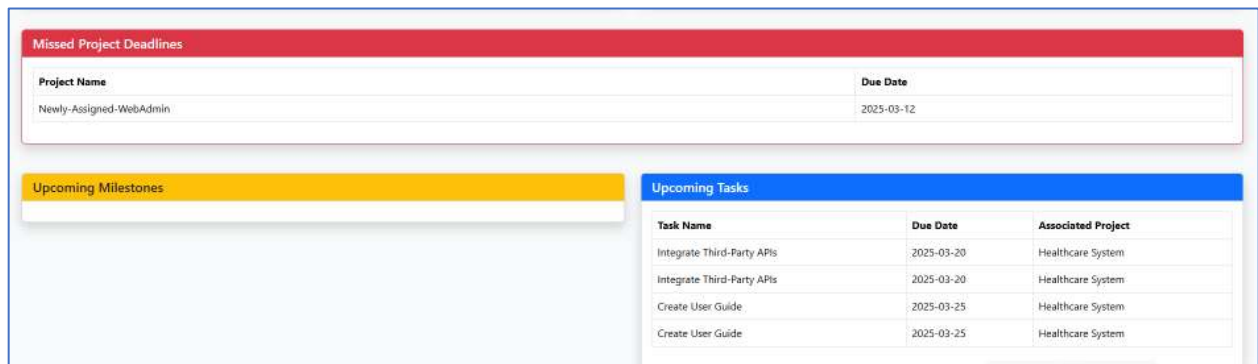


Figure 39: Homepage (2)

11.2. Complex form and queries

11.2.1. SQL Queries

I) Top 3 performers

The query below returns the top 3 performers who have done the most number of tasks in the selected project.

```
SELECT u.userid AS "USERID",
       u.username AS
"Username",
       u.useremail AS "Email",
       u.usercontact AS
"Contact",
       r.rolename AS "Roles",
       t.task_count AS "Task
Count"
FROM usersdetails u
JOIN rolesofuser r ON r.roleid
= u.roleid
JOIN (
    SELECT upt.USERID,
COUNT(upt.TaskID) as task_count
FROM
userprojecttaskdetails upt
WHERE taskid IN (
    SELECT taskid FROM
tasks WHERE LOWER(taskstatus)='completed'
)
AND upt.projectid =
:projectid
GROUP BY upt.USERID
ORDER BY COUNT(upt.TaskID)
DESC
) t ON u.userid = t.userid
WHERE ROWNUM <= 3
```

II) User project

The query given below return the project details the selected user is working on

Displays the detail of the user

```
select u.userid "USERID", u.username "Username", u.useremail
"Email", u.usercontact "Contact", r.rolename from usersdetails u join
rolesofuser r on r.roleid=u.roleid where u.userid=:usersdetails
```

Displays the details of the projects the user is working on

```
select p.projectid "ProjectID", p.projectname "Project  
Name", p.projectstartdate "Start date", p.projectduedate  
"Due Date", p.projectstatus "Status"  
from projectdetails p  
join userproject up on  
p.projectid = up.projectid where up.userid=:usersdetails
```

III) Project Milestones

The given query returns the milestones of the selected project.

Displays the details of the selected project

```
select p.projectid "ProjectID", p.projectname "Project  
Name",  
p.projectstartdate "Start date", p.projectduedate "Due  
Date", p.projectstatus "Status"  
from projectdetails p  
where  
p.projectid=:project
```

Displays the milestones of the selected project

```
select p.projectid "ProjectID", p.projectname "Project  
Name",  
p.projectstartdate "Start date", p.projectduedate "Due  
Date", p.projectstatus "Status"  
from projectdetails p  
where  
p.projectid=:project
```

11.2.2. Complex forms

I) Top Performer

User ID	Username	Email	Contact	Roles	Task Count
3	Riyon Paudel	riyon.paudel@example.com	9813333333	Designer	1
4	Sudarshan Gyawali	sudarshan.gyawali@example.com	9814444444	Project Manager	1

Figure 40: Top Performer Web

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Top Performer

Choose a project:

Newly-Assigned-WebAdmin

User ID	Username	Email	Contact	Roles	Task Count
1	Rhythm Paudel	rhythm.paudel@example.com	9811111111	Manager	5
2	Sampada Bastola	sampada.bastola@example.com	9812222222	Developer	3
4	Sudarshan Gyawali	sudarshan.gyawali@example.com	9814444444	Project Manager	3

Figure 41: Top Performer Web (2)

II) User Project

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

User Projects

Choose a user:

Rhythm Paudel

User ID	Username	Email	Contact	ROLENAME
1	Rhythm Paudel	rhythm.paudel@example.com	9811111111	Manager

Project ID	Project Name	Start date	Due Date	Status
1	E-Commerce Platform	2025-01-01	2025-03-31	Ongoing
16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12	Ongoing

Figure 42: User Project Web

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

User Projects

Choose a user:

Sudin Thapa

User ID	Username	Email	Contact	ROLENAME
8	Sudin Thapa	sudin.thapa@example.com	9818888888	Database Administrator

Project ID	Project Name	Start date	Due Date	Status
4	Customer Support Tool	2025-02-15	2025-04-30	Planned

Figure 43: User Project Web(2)

III) Project Milestones

LS Corporation

[Home](#) [Users](#) [Projects](#) [Milestones](#) [Subtasks](#) [Tasks](#) [Others](#) ▾

Project Milestones

Choose a project:

E-Commerce Platform

Project ID	Project Name	Start date	Due Date	Status
1	E-Commerce Platform	1/1/2025 12:00:00 AM	2025-03-31	Ongoing

Milestone ID	Milestone name	Priority	Due Date of Milestone
1	Prototype Completion	High	2025-01-20
2	Beta Release	Medium	2025-02-15
3	Final Deployment	High	2025-03-31

Figure 44: Project Milestone Web

LS Corporation

[Home](#) [Users](#) [Projects](#) [Milestones](#) [Subtasks](#) [Tasks](#) [Others](#) ▾

Project Milestones

Choose a project:

Travel Booking System

Project ID	Project Name	Start date	Due Date	Status
8	Travel Booking System	2/5/2025 12:00:00 AM	2025-04-20	Planned

Milestone ID	Milestone name	Priority	Due Date of Milestone
22	Feature Planning	High	2025-01-25
23	Feature Implementation	Medium	2025-03-01
24	Beta Launch	High	2025-04-20

Figure 45: Project Milestone Web (2)

11.3. Simple form

1) User Webform

LS Corporation Home **Users** Projects Milestones Subtasks Tasks Others ▾

User Details

User ID:

Username:

Email Address:

Contact Number:

User Role:

	ID	Username	Email Address	Contact Number	Role
Edit Delete	1	Rhythm Paudel	rhythm.paudel@example.com	9811111111	Manager ▾
Edit Delete	2	Sampada Bastola	sampada.bastola@example.com	9812222222	Developer ▾
Edit Delete	3	Riyon Paudel	riyon.paudel@example.com	9813333333	Designer ▾

Figure 46: User Basic Webform

2) Project Webform

LS Corporation Home Users **Projects** Milestones Subtasks Tasks Others ▾

Project Details

Project ID:

Project Name:

Start Date:

Due Date:

Status:

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	1	E-Commerce Platform	2025-01-01	2025-03-31		Ongoing
Edit Delete	2	Inventory Management System	2025-01-15	2025-04-15		Planned
Edit Delete	3	Employee Tracking System	2025-02-01	2025-05-01		Ongoing
Edit Delete	4	Customer Support Tool	2025-02-15	2025-04-30		Planned

Figure 47: Project Basic Webform

3) Milestone

LS Corporation Home Users Projects Milestones Subtasks Tasks Others ▾

Milestones Details

Milestone ID:

Milestone Name:

Priority:

Due Date:

Project:

	ID	Name	Priority	Due Date	Project ID	Project Name
Edit Delete	1	Prototype Completion	High	2025-01-20	1	E-Commerce Platform
Edit Delete	2	Beta Release	Medium	2025-02-15	1	E-Commerce Platform
Edit Delete	3	Final Deployment	High	2025-03-31	1	E-Commerce Platform

Figure 48: Milestone Basic Webform

4) Tasks

LS Corporation Home Users Projects Milestones Subtasks Tasks Others ▾

Tasks Details

ID:

Name:

Start Date:

Due Date:

Status:

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	1	Requirement Gathering	2025-01-02	2025-01-10		Pending
Edit Delete	2	Design Database Schema	2025-01-11	2025-01-15		In Progress
Edit Delete	3	Set Up Development Environment	2025-01-16	2025-01-20	2025-01-20	Completed
Edit Delete	4	Develop User Login Module	2025-01-21	2025-01-25		Pending

Figure 49: Tasks Basic Webform

5) Subtasks

LS Corporation

HomeUsersProjectsMilestonesSubtasksTasksOthers

Subtasks Details

ID:

Name:

Start Date:

Due Date:

Status:

Task:

Requirement Gathering

Insert

Cancel

	ID	Name	Start Date	Due Date	Completed Date	Status	TaskID	Task
Edit Delete	1	Define Requirements	2025-01-05	2025-01-02	2025-01-02	Completed	1	Requirement Gathering
Edit Delete	2	Create Use Case Diagrams	2025-01-10	2025-01-06	2025-01-06	Completed	1	Requirement Gathering

Figure 50: Subtasks Basic Webform

12. User Manual

The user will be introduced with the dashboard when opening the web application for the first time. In the dashboard, the user can see a navigation bar, which will lead to various sections of the application.

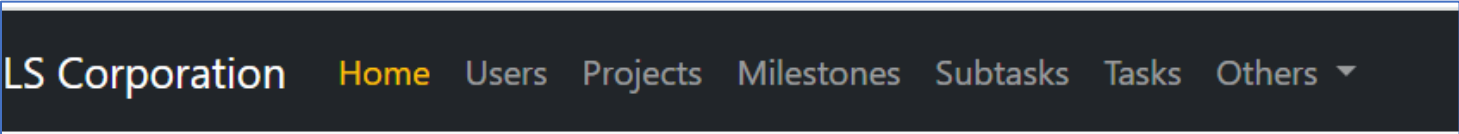


Figure 51: Navigation bar

The home is the default page for the application where user can view a brief overview of the projects, milestones, and tasks.

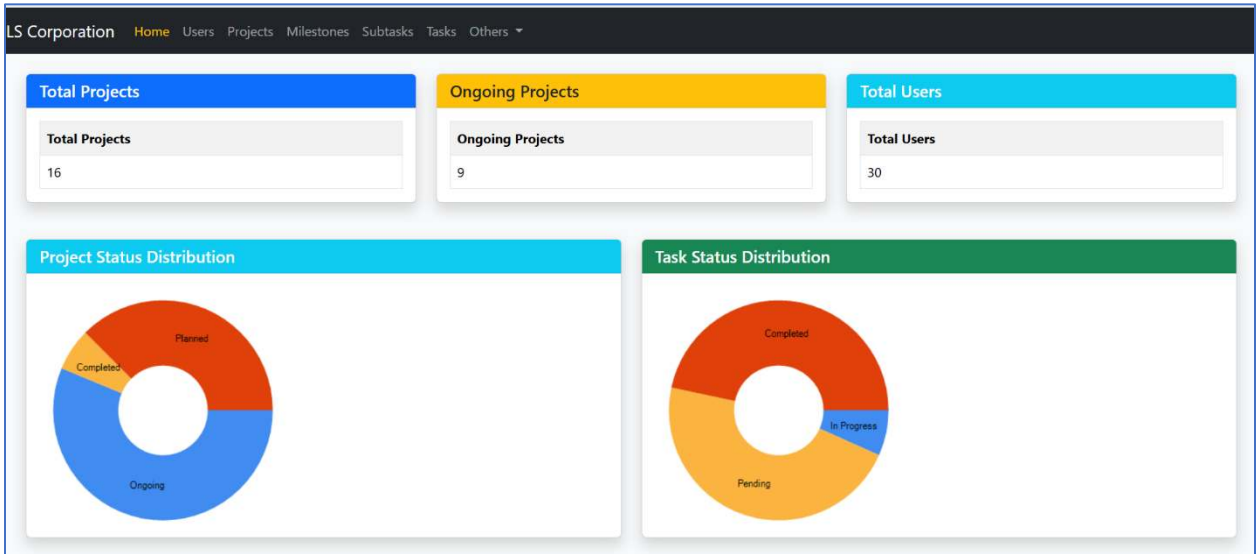


Figure 52: User Manual (Dashboard)

I) Navigating between pages

User can click on any page they want to be redirected to. In the example provided clicking on Users will be redirected to users page where the user will be able to view, edit and add users.

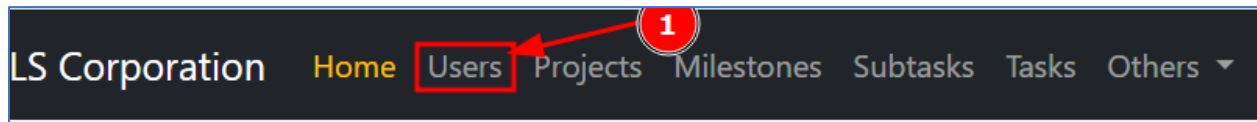


Figure 53: Navigating to users page (User Manual)

II) Actions on Users page

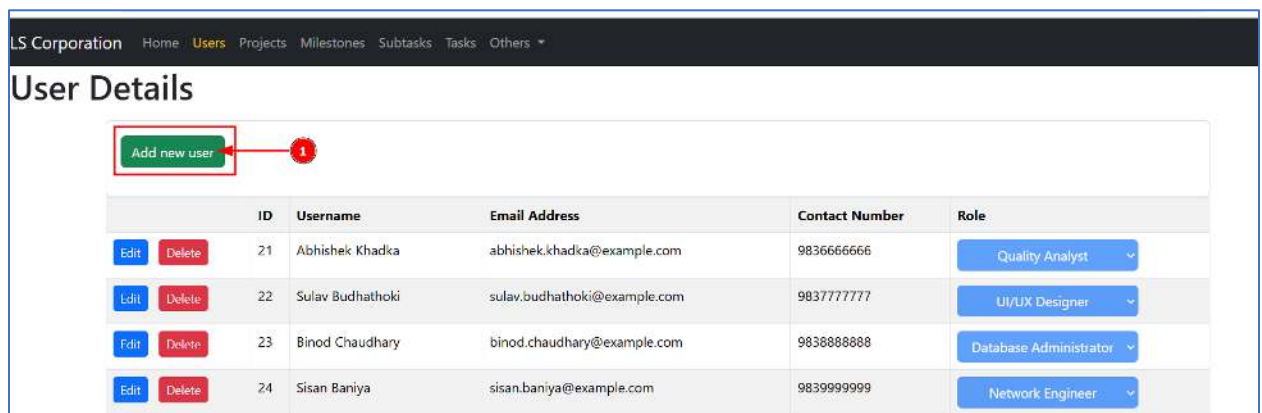


Figure 54: Add new user button

Performing the actions shown below will add a new user.

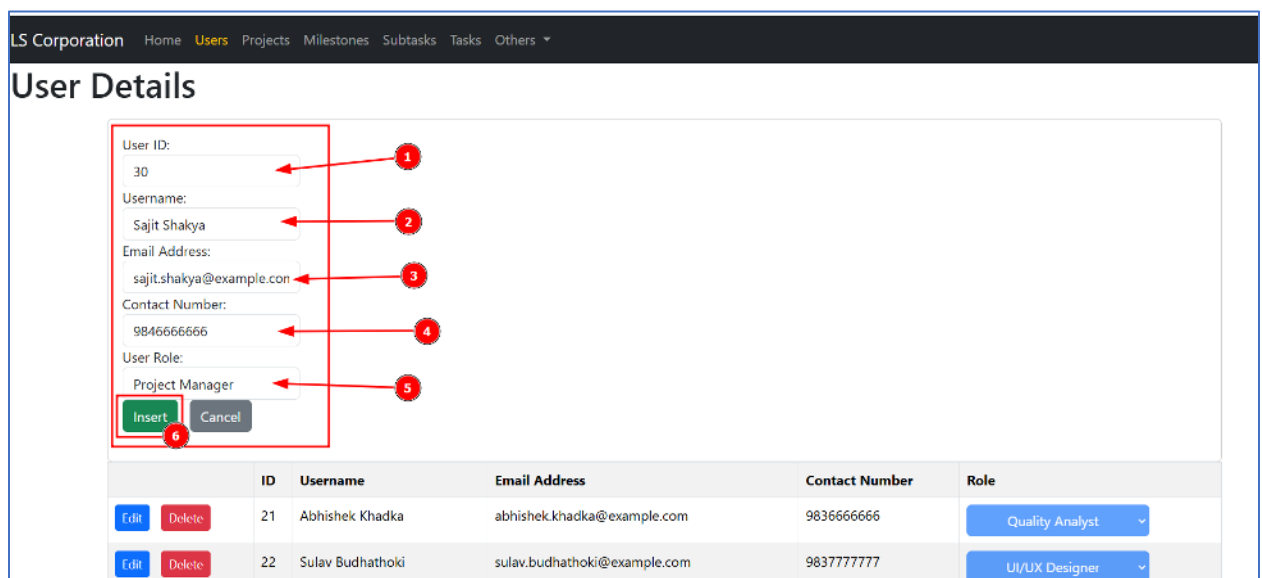


Figure 55: Adding a user (User Manual)

On clicking edit button the user will have the access to edit the selected row and either update or cancel. On clicking the delete button, the selected user will be deleted.

LS Corporation

Home

Users

Projects

Milestones

Subtasks

Tasks

Others

User Details

Add new user

	ID	Username	Email Address	Contact Number	Role
Update Cancel	1	Rhythm Paudel	This field is edited	9811111111	Manager
Edit Delete	2	Sampada Bastola	sampada.bastola@example.com	9812222222	Developer
Edit Delete	3	Riyon Paudel	riyon.paudel@example.com	9813333333	Designer

Figure 56: Editing User (User Manual)

III) Actions on Projects page

To add a new project following actions can be performed as shown in the picture.

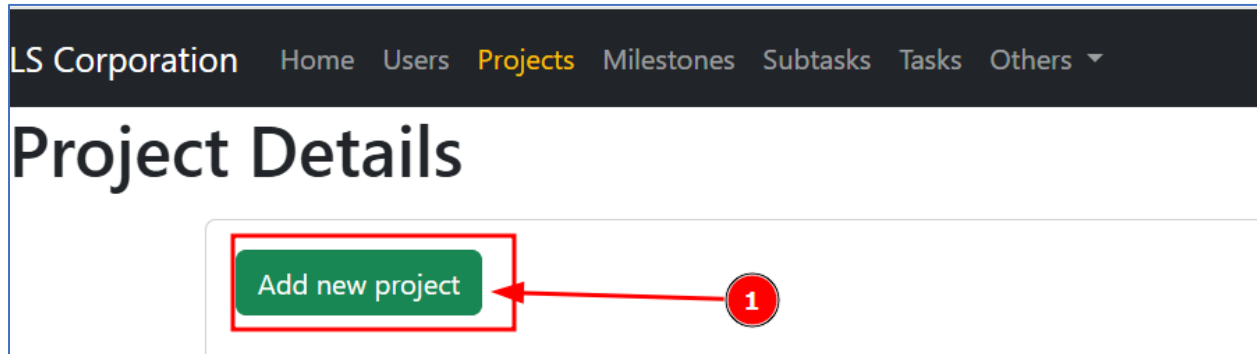


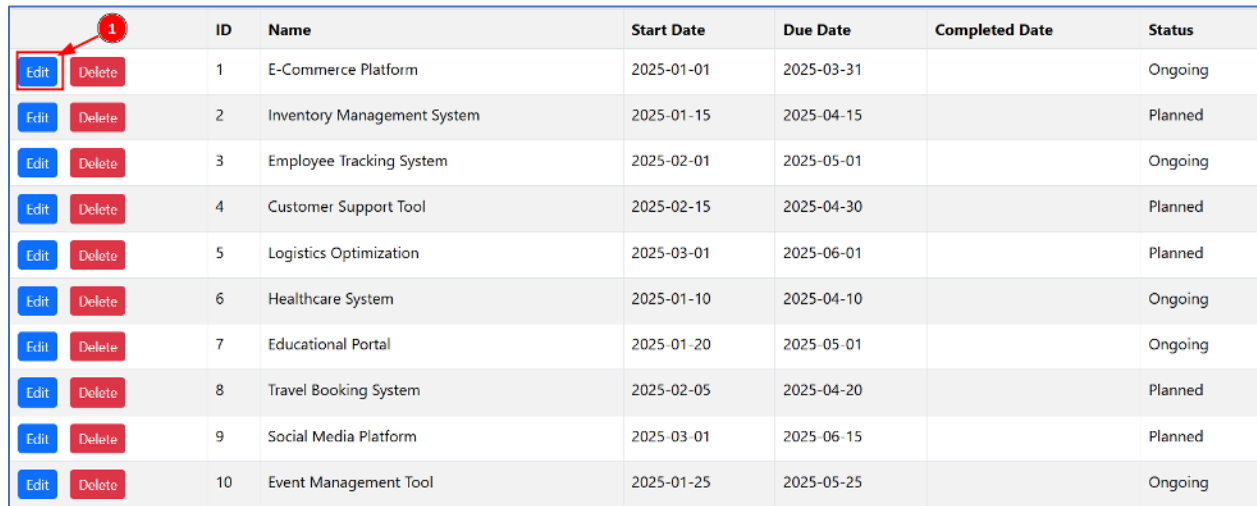
Figure 57: Add new project button (User Manual)

The screenshot shows the 'Project Details' page with a dark navigation bar at the top containing 'LS Corporation', 'Home', 'Users', 'Projects' (highlighted), 'Milestones', 'Subtasks', 'Tasks', and 'Others' with a dropdown arrow. Below the navigation bar is the title 'Project Details'. In the main content area, there is a form to add a new project. The form contains the following fields and buttons:

- Project ID:** A text input field containing '312'. A red arrow points from a red circle containing the number '2' to this field.
- Project Name:** A text input field containing 'Test Proj'. A red arrow points from a red circle containing the number '3' to this field.
- Start Date:** A date input field containing '2025-03-12'. A red arrow points from a red circle containing the number '4' to this field.
- Due Date:** A date input field containing '2025-03-19'. A red arrow points from a red circle containing the number '5' to this field.
- Status:** A dropdown menu showing 'Pending'. A red arrow points from a red circle containing the number '6' to this field.
- Buttons:** There are two buttons at the bottom: a green 'Insert' button and a grey 'Cancel' button. A red arrow points from a red circle containing the number '7' to the 'Insert' button.

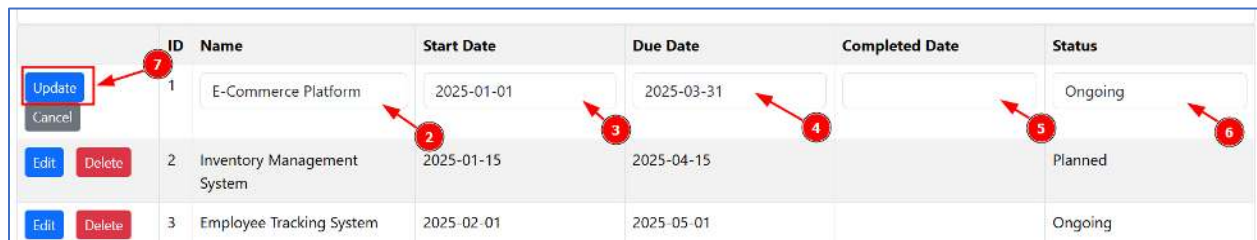
Figure 58: Adding new project (User Manual)

To edit a project or delete a project following actions can be performed.



	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	1	E-Commerce Platform	2025-01-01	2025-03-31		Ongoing
Edit Delete	2	Inventory Management System	2025-01-15	2025-04-15		Planned
Edit Delete	3	Employee Tracking System	2025-02-01	2025-05-01		Ongoing
Edit Delete	4	Customer Support Tool	2025-02-15	2025-04-30		Planned
Edit Delete	5	Logistics Optimization	2025-03-01	2025-06-01		Planned
Edit Delete	6	Healthcare System	2025-01-10	2025-04-10		Ongoing
Edit Delete	7	Educational Portal	2025-01-20	2025-05-01		Ongoing
Edit Delete	8	Travel Booking System	2025-02-05	2025-04-20		Planned
Edit Delete	9	Social Media Platform	2025-03-01	2025-06-15		Planned
Edit Delete	10	Event Management Tool	2025-01-25	2025-05-25		Ongoing

Figure 59: Editing User (1) (User Manual)



	ID	Name	Start Date	Due Date	Completed Date	Status
Update Cancel	1	E-Commerce Platform	2025-01-01	2025-03-31		Ongoing
Edit Delete	2	Inventory Management System	2025-01-15	2025-04-15		Planned
Edit Delete	3	Employee Tracking System	2025-02-01	2025-05-01		Ongoing

Figure 60: Editing User (2) (User Manual)

Similarly User can press delete button for deleting a project

IV) Actions on Milestones page

To add a new milestones following steps should be followed.

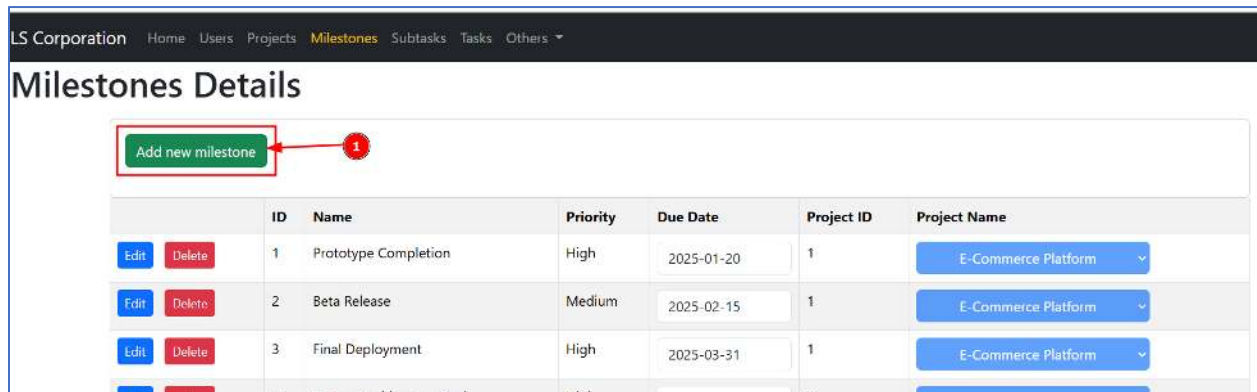


Figure 61: Adding Milestone (1) (User Manual)

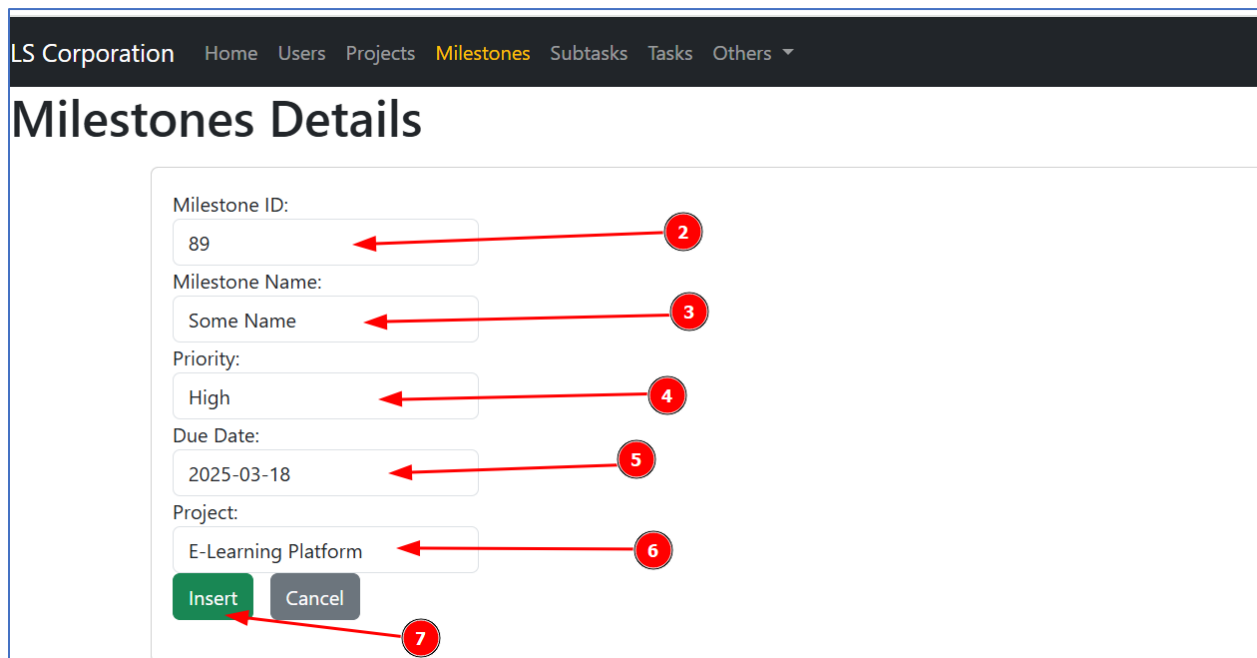


Figure 62: Adding Milestone (2) (User Manual)

To edit a milestone user should click on edit button on the milestone page and update the field.

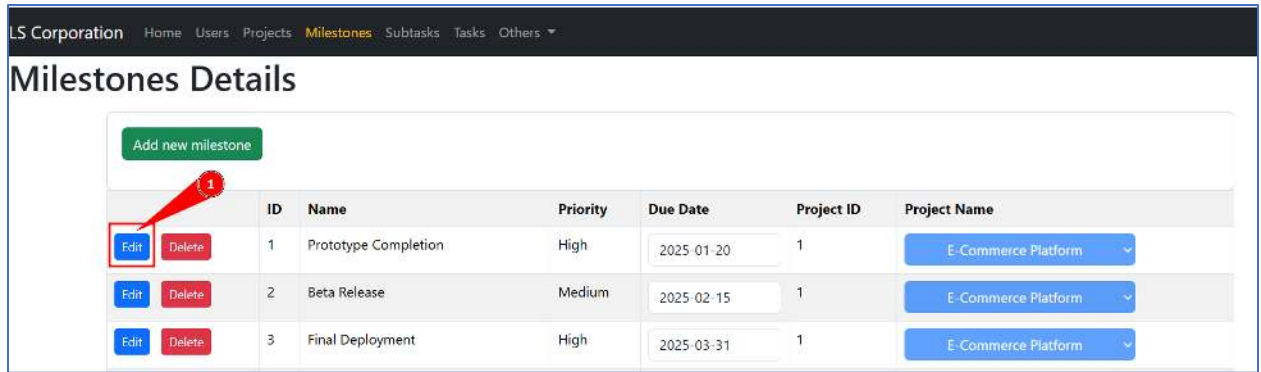


Figure 63: Editing Milestone (1) (User Manual)

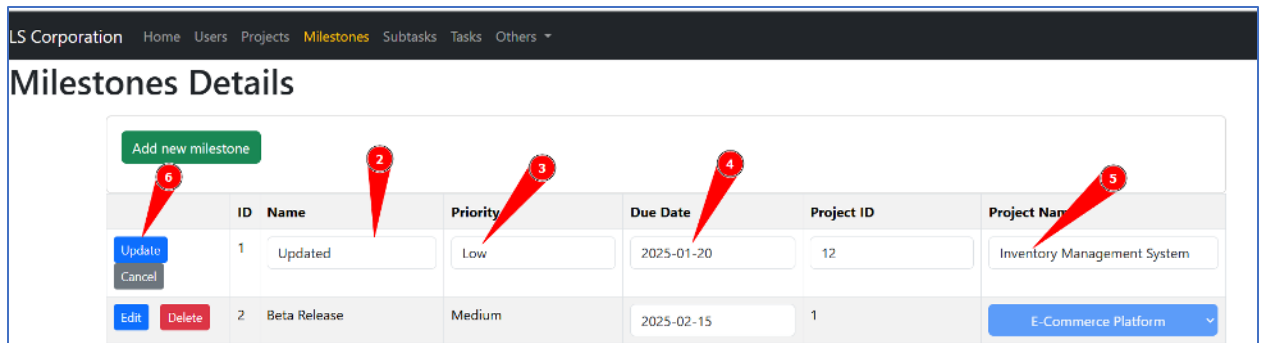


Figure 64: Editing Milestone (2) (User Manual)

To delete any milestones user can click on delete button on the milestone page.

V) Actions on Subtask page

To add a new a subtask following steps needs to be performed.

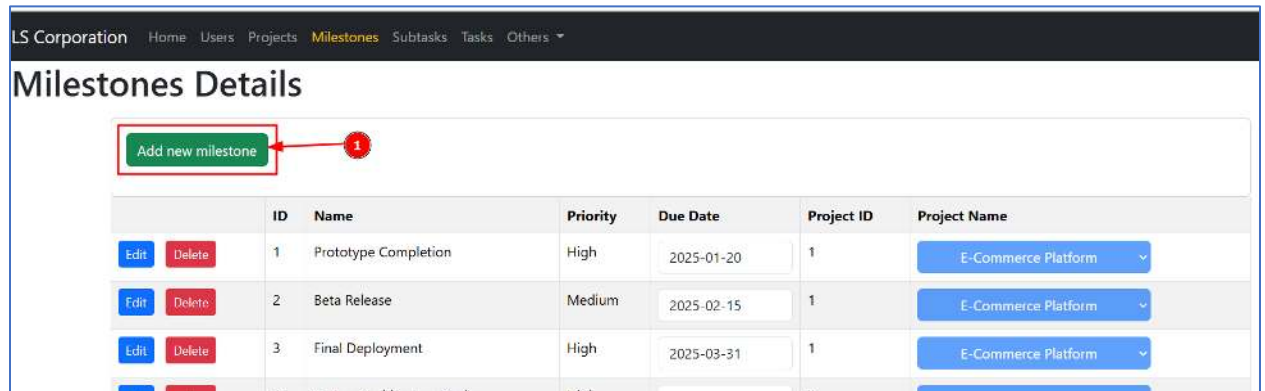


Figure 65: Add Milestone (1) (User Manual)

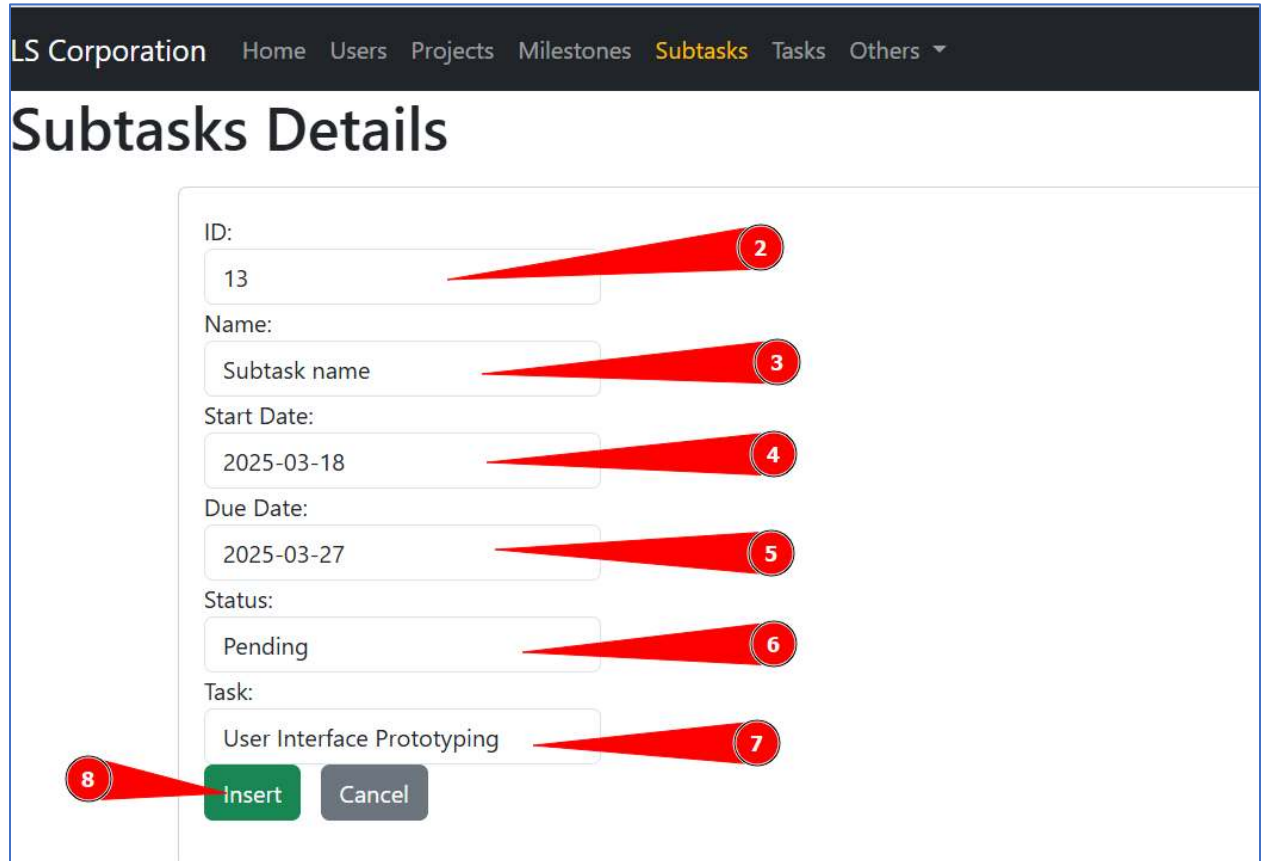


Figure 66: Add milestone (2) (User Manual)

To edit existing milestone following steps shown in the picture can be performed.

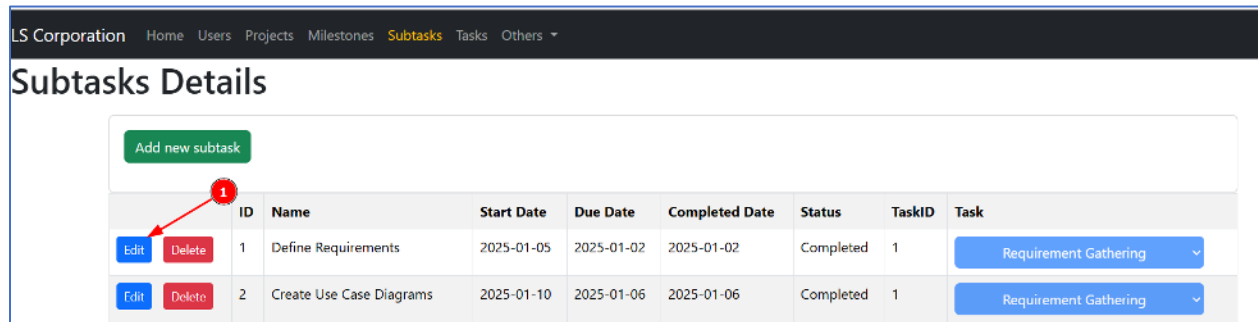


Figure 67: Editing Subtask (1) (User Manual)

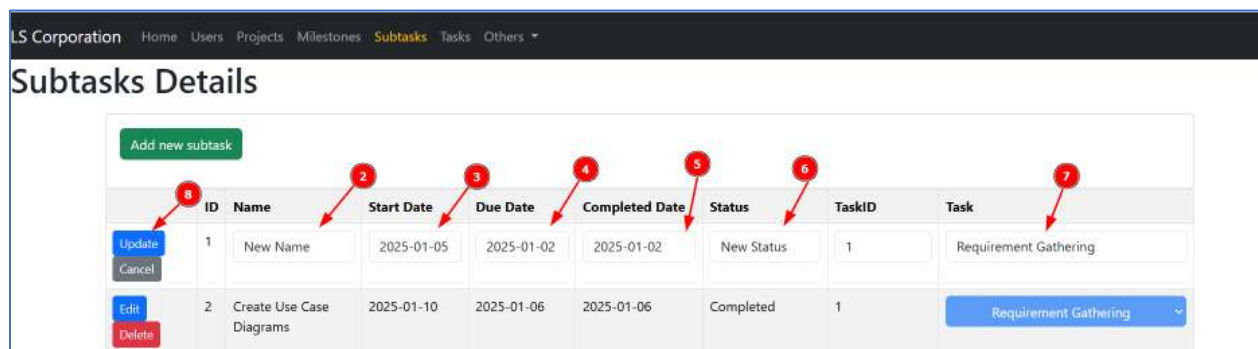


Figure 68: Editing Subtask (2) (User Manual)

To delete any subtask user can click on delete button on the subtask page.

VI) Actions on Task page

To add a new a subtask following steps needs to be performed.

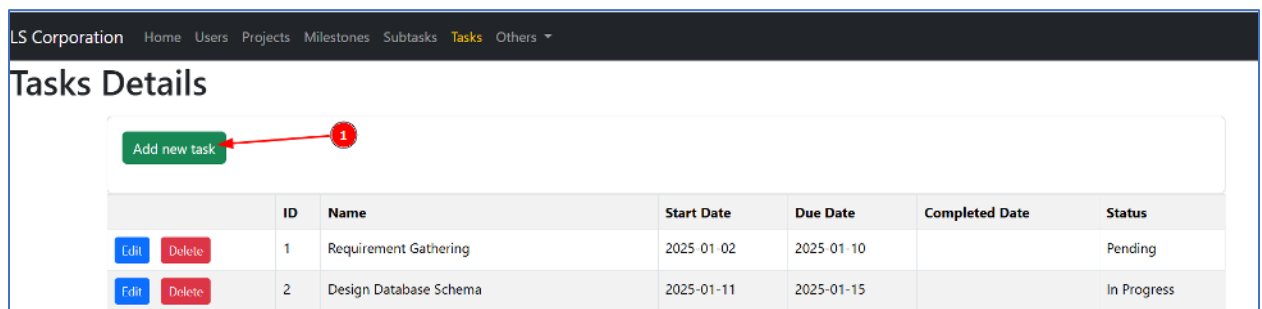


Figure 69: Add new task (1) (User Manual)

LS Corporation Home Users Projects Milestones Subtasks **Tasks** Others ▾

Tasks Details

ID:

442

2

Name:

Task Name

3

Start Date:

2025-03-18

4

Due Date:

2025-03-28

5

Status:

Pending

6

7

insert

Cancel

Figure 70: Add new task (2) (User Manual)

To edit any existing task following steps should be performed.

LS Corporation Home Users Projects Milestones Subtasks **Tasks** Others ▾

Tasks Details

Add new task

1

	ID	Name	Start Date	Due Date	Completed Date	Status
edit delete	1	Requirement Gathering	2025-01-02	2025-01-10		Pending
edit delete	2	Design Database Schema	2025-01-11	2025-01-15		In Progress
edit delete	3	Set Up Development Environment	2025-01-16	2025-01-20	2025-01-20	Completed

Figure 71: Edit Task (1) (User Manual)

LS Corporation Home Users Projects Milestones Subtasks **Tasks** Others ▾

Tasks Details

Add new task

	ID	Name	Start Date	Due Date	Completed Date	Status
Update Cancel	1	New task name	2025-01-02	2025-01-10	2025-03-23	Completed
Edit Delete	2	Design Database Schema	2025-01-11	2025-01-15		In Progress
Edit Delete	3	Set Up Development Environment	2025-01-16	2025-01-20	2025-01-20	Completed

Figure 72: Edit Task (2) (User Manual)

To delete any task user can click on delete button on the task page.

13. Testing

13.1. Users basic webform testing

Before performing test cases on users, users page from the navigation bar should be opened.

I) Adding a user

Test No	1
Objective	To check if new user can be added.
Actions	a) Click add new user. b) Fill out the form with following values: UserID → 4312 Username → testuser Email Address → testuser@example.com Contact Number → 9991112234 c) Click insert button.
Expected Result	New user should be added at the end of the list.
Actual Result	New user was added at the end of the list.
Conclusion	Test was successful.

Table 13: Adding a user (test)

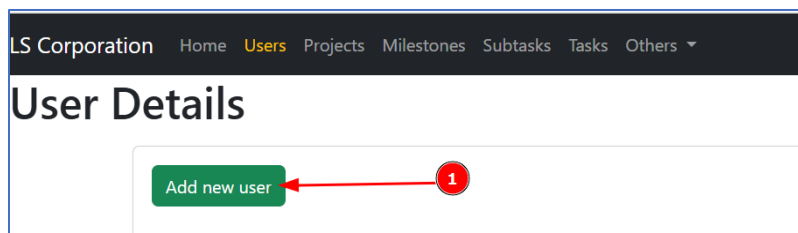


Figure 73: Adding a user (1)

LS Corporation Home **Users** Projects Milestones Subtasks Tasks Others ▾

User Details

User ID: 4321

Username: testuser

Email Address: testuser@example.com

Contact Number: 9991112234

User Role: Security Analyst

ID	Username	Email Address

Figure 74: Adding a user (2)

LS Corporation Home **Users** Projects Milestones Subtasks Tasks Others ▾

User Details

User ID: 4321

Username: testuser

Email Address: testuser@example.com

Contact Number: 9991112234

User Role: Security Analyst

ID	Username	Email Address

Figure 75: Adding a user (3)

II) Editing a user

Test No	2
Objective	To check if the user is editable.
Actions	a) Navigate to last table (4) b) Click on edit on with user id of 4321 from user page. c) Change username as trusteduser, and useremail as updatedemail@example.com. d) Click update button
Expected Result	Username and UserEmail should be changed.
Actual Result	Username and UserEmail was changed.
Conclusion	Test was successful.

Table 14: Editing a user (test)

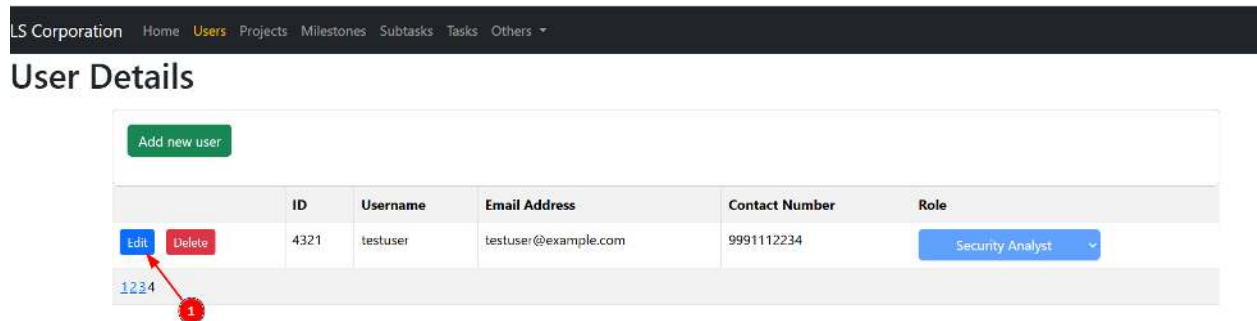


Figure 76: Editing a user (1)

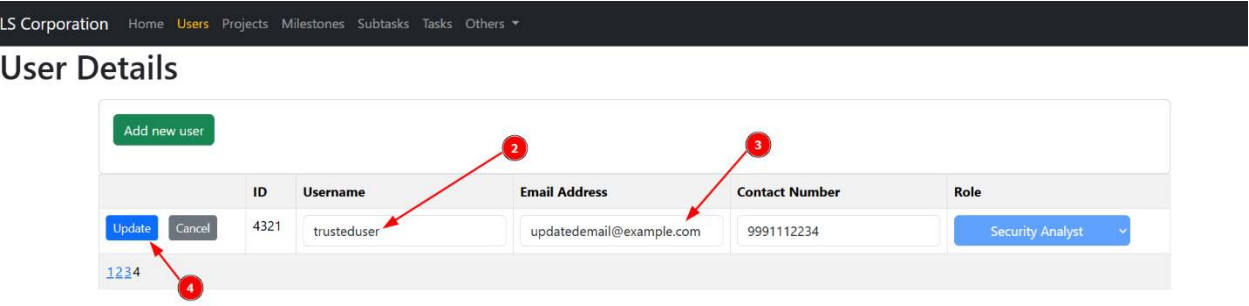


Figure 77: Editing a user (2)

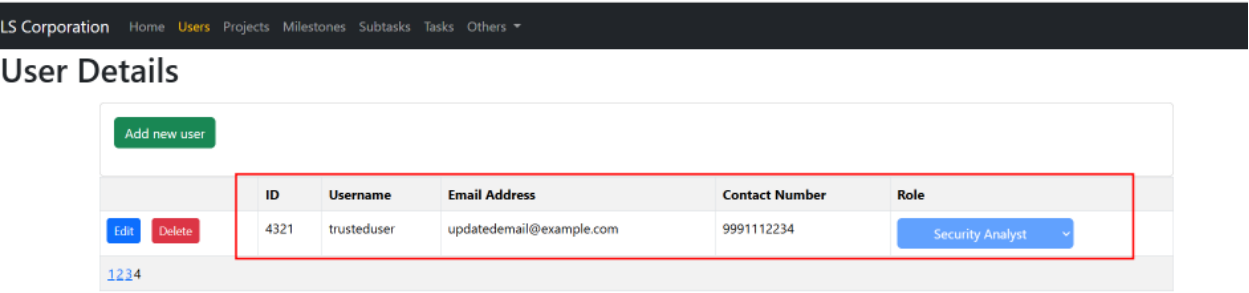


Figure 78: Editing a user (3)

III) Deleting a user

Test No	3
Objective	To check if the user can be deleted.
Actions	a) Navigate to users page b) Click on delete button with user id 4321 on the last paginated page of users table.
Expected Result	User should be deleted.
Actual Result	User was deleted.
Conclusion	Test was successful.

Table 15: Deleting a user (test)

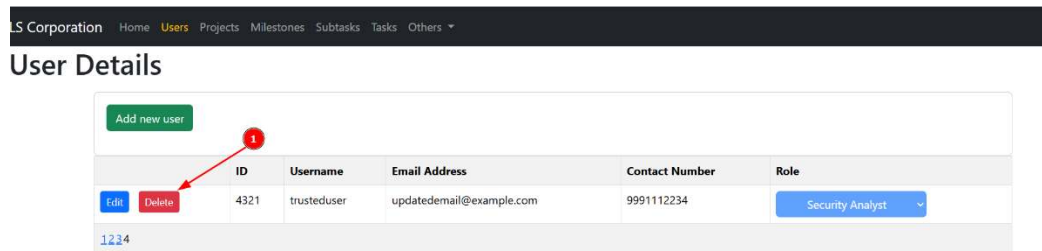


Figure 79: Deleting a user (1)

Add new user						
	ID	Username	Email Address	Contact Number	Role	
Edit Delete	21	Abhishek Khadka	abhishek.khadka@example.com	9836666666	Quality Analyst	
Edit Delete	22	Sulav Budhathoki	sulav.budhathoki@example.com	9837777777	UI/UX Designer	
Edit Delete	23	Binod Chaudhary	binod.chaudhary@example.com	9838888888	Database Administrator	
Edit Delete	24	Sisan Baniya	sisan.baniya@example.com	9839999999	Network Engineer	
Edit Delete	25	Aadarsha Lamichhane	aadarsha.lamichhane@example.com	9841111111	Software Engineer	
Edit Delete	26	Rajesh Hamal	rajesh.hamal@example.com	9842222222	Technical Support	
Edit Delete	27	KP Oli	kp.oli@example.com	9843333333	Security Analyst	
Edit Delete	28	Lekhnath Adhikari	lekhnath.adhikari@example.com	9844444444	Research Analyst	
Edit Delete	29	Sushil Koirala	sushil.koirala@example.com	9845555555	IT Consultant	
Edit Delete	30	Sajit Shakya	sajit.shakya@example.com	9846666666	Product Manager	

123

Figure 80: Deleting a user (2)

13.2. Projects basic web form testing

Before testing any project related tasks, projects page should be opened from the navigation bar.

I) Adding a project

Test No	4
Objective	To check if a new project can be added.
Actions	a) Click on add new project. b) Fill out the details with following values Project ID → 4554 Project Name → Test Project Start Date: 2025-03-12 Due Date: 2025-03-13 Status: Pending (select from drop down) c) Click on insert button
Expected Result	A new project should be added to end of the list
Actual Result	A new project was added to end of the list.
Conclusion	Test was successful.

Table 16: Adding a project (Test)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	1	E-Commerce Platform	2025-01-01	2025-03-31		Ongoing
Edit Delete	2	Inventory Management System	2025-01-15	2025-04-15		Planned
Edit Delete	3	Employee Tracking System	2025-02-01	2025-05-01		Ongoing

Figure 81: Adding a project (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Project Details

Project ID:
4554
Project Name:
Test Project
Start Date:
2025-03-12
Due Date:
2025-03-13
Status:
Pending
Insert
Cancel

	ID	Name	Start Date	Due Date	Completed Date	Status

Figure 82: Adding a project (2)

LS Corporation Home Users **Projects** Milestones Subtasks Tasks Others ▾

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
Edit Delete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
Edit Delete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
Edit Delete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
Edit Delete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
Edit Delete	16	Newly Assigned WebAdmin	2025-01-01	2025-03-12		Ongoing
Edit Delete	4554	Test Project	2025-03-12	2025-03-13		Pending

12

Figure 83: Adding a project (3)

II) Editing a project

Test No	5
Objective	To check if the project is editable
Actions	a) Click on edit button with ID 4554 on project page b) Make changes in name as 'Editing a test proj' and status field as 'Cancelled' c) Click on update
Expected Result	The name and status field should be updated of the selected project.
Actual Result	The name and status field was updated of the selected project.
Conclusion	Test was successful.

Table 17: Editing a project (test)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
Edit Delete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
Edit Delete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
Edit Delete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
Edit Delete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
Edit Delete	16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12		Ongoing
Edit Delete	4554	Test Project	2025-03-12	2025-03-13		Pending

12
1

Figure 84: Editing a project (1)

LS Corporation

[Home](#)
[Users](#)
[Projects](#)
[Milestones](#)
[Subtasks](#)
[Tasks](#)
[Others](#)

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
Edit Delete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
Edit Delete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
Edit Delete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
Edit Delete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
Edit Delete	16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12		Ongoing
Update Cancel	4554	Editing a test proj	2025-03-12	2025-03-13		Cancelled

Figure 85: Editing a project (2)

LS Corporation

[Home](#)
[Users](#)
[Projects](#)
[Milestones](#)
[Subtasks](#)
[Tasks](#)
[Others](#)

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
Edit Delete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
Edit Delete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
Edit Delete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
Edit Delete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
Edit Delete	16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12		Ongoing
Edit Delete	4554	Editing a test proj	2025-03-12	2025-03-13		Cancelled

Figure 86: Editing a project (3)

III) Deleting a project

Test No	6
Objective	To check if the project can be deleted.
Actions	a) Click on the delete button on the last record of the table with ID of 4554.
Expected Result	The selected record should be deleted.
Actual Result	The selected record was deleted.
Conclusion	Test was successful.

Table 18: Deleting a project (Test)

LS Corporation

HomeUsersProjectsMilestonesSubtasksTasksOthers

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
EditDelete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
EditDelete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
EditDelete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
EditDelete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
EditDelete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
EditDelete	16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12		Ongoing
EditDelete	4554	Editing a test proj	2025-03-12	2025-03-13		Cancelled

12

Figure 87: Deleting a project (1)

Project Details

Add new project

	ID	Name	Start Date	Due Date	Completed Date	Status
<a>Edit <a>Delete	11	Banking Application	2025-02-10	2025-05-30		Ongoing
<a>Edit <a>Delete	12	Mobile Wallet App	2025-03-15	2025-07-15		Planned
<a>Edit <a>Delete	13	E-Learning Platform	2025-01-05	2025-04-05	2025-04-05	Completed
<a>Edit <a>Delete	14	Retail POS System	2025-02-25	2025-06-30		Ongoing
<a>Edit <a>Delete	15	IoT Home Automation	2025-01-30	2025-05-31		Ongoing
<a>Edit <a>Delete	16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12		Ongoing

12

Figure 88: Deleting a project (2)

13.3. Milestones basic web form testing

Before testing any milestones related tasks, milestone page should be opened from the navigation bar.

I) Adding a milestone

Test No	7
Objective	To check if new milestone can be added
Actions	a) Click on 'Add new milestone' button. b) Fill up the required data with following values Milestone ID → 238 Milestone Name → 'New Milestone' Priority → 'High' Due Date → 2025-03-23 Project → IoT Home Automation (select from drop down) c) Click on insert.
Expected Result	New data should be added.
Actual Result	New data was added.
Conclusion	Test was successful.

Table 19: Adding a milestone (Test)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Milestones Details

Add new milestone

	ID	Name	Priority	Due Date	Project ID	Project Name
Edit Delete	1	Prototype Completion	High	2025-01-20	1	E-Commerce Platform
Edit Delete	2	Beta Release	Medium	2025-02-15	1	E-Commerce Platform
Edit Delete	3	Final Deployment	High	2025-03-31	1	E-Commerce Platform

Figure 89: Adding a milestone (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Milestones Details

Milestone ID:
238

Milestone Name:
New Milestone

Priority:
High

Due Date:
2025-03-23

Project:
IoT Home Automation

Insert
Cancel

Figure 90: Adding a milestone (2)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Milestones Details

Add new milestone

	ID	Name	Priority	Due Date	Project ID	Project Name
Edit Delete	238	New Milestone	High	2025-03-23	15	IoT Home Automation

1234

Figure 91: Adding a milestone (3)

II) Editing a milestone

Test No	8
Objective	To check if the milestone is editable.
Actions	a) Click on “Edit” button for any record. b) Change name as ‘New old mile’ and priority field as ‘Low’ (select from drop down). c) Click on “Update” button
Expected Result	Selected milestone name and priority field should be updated.
Actual Result	Selected milestone name and priority field was updated.
Conclusion	Test was successful.

Table 20: Editing a milestone (Test)

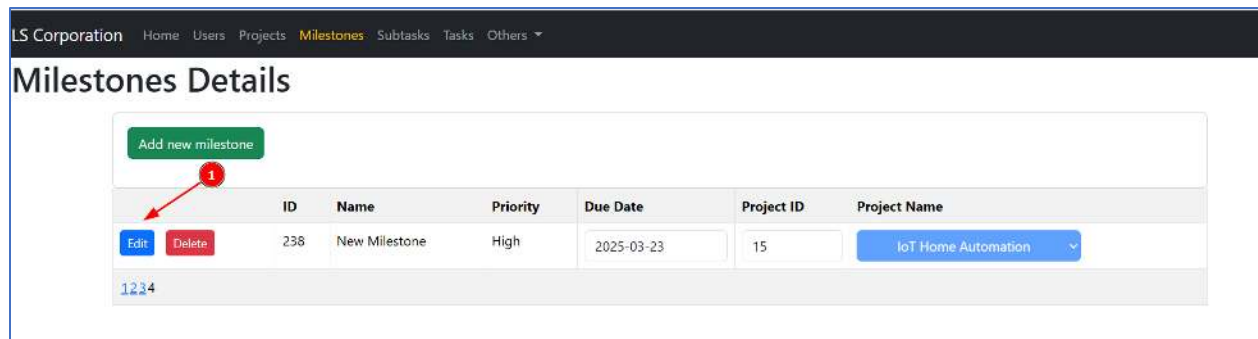


Figure 92: Editing a milestone (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Milestones Details

Add new milestone

	ID	Name	Priority	Due Date	Project ID	Project Name
Update Cancel	238	New Old Mile	Low	2025-03-23	15	IoT Home Automation

1234

Figure 93: Editing a milestone (2)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Milestones Details

Add new milestone

	ID	Name	Priority	Due Date	Project ID	Project Name
Edit Delete	238	New Old Mile	Low	2025-03-23	15	IoT Home Automation

1234

Figure 94: Editing a milestone (3)

III) Deleting a milestone

Test No	9
Objective	To check if a milestone record can be deleted.
Actions	a) Click on “Delete” button of the last record with ID 238 from milestones page.
Expected Result	The selected record should be deleted.
Actual Result	The selected record was deleted.
Conclusion	Test was successful.

Table 21: Deleting a milestone (Test)

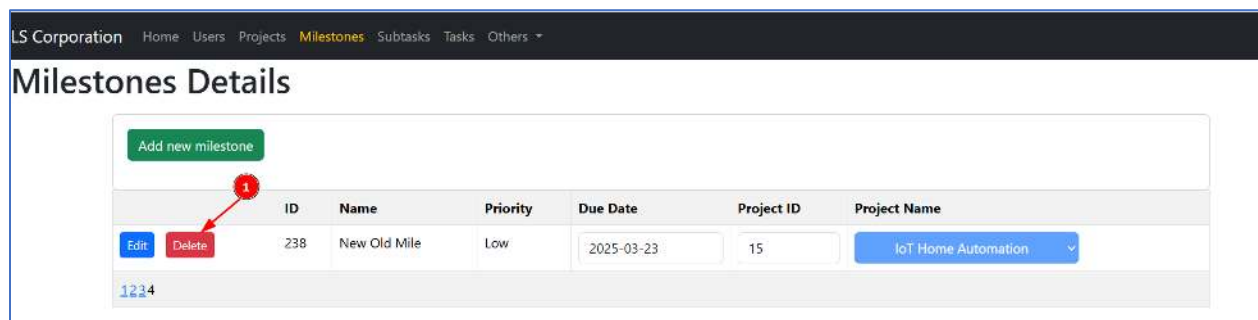


Figure 95: Deleting a milestone (1)

		ID	Name	Priority	Due Date	Project ID	Project Name
Edit	Delete	21	User Testing	High	2025-05-01	7	Educational Portal
Edit	Delete	22	Feature Planning	High	2025-01-25	8	Travel Booking System
Edit	Delete	23	Feature Implementation	Medium	2025-03-01	8	Travel Booking System
Edit	Delete	24	Beta Launch	High	2025-04-20	8	Travel Booking System
Edit	Delete	25	Requirements Finalization	High	2025-02-01	9	Social Media Platform
Edit	Delete	26	Development	Medium	2025-04-01	9	Social Media Platform
Edit	Delete	27	Launch	High	2025-06-15	9	Social Media Platform
Edit	Delete	28	Planning Phase	High	2025-01-15	10	Event Management Tool
Edit	Delete	29	Development Completion	Medium	2025-03-15	10	Event Management Tool
Edit	Delete	30	Final Deployment	High	2025-05-25	10	Event Management Tool
123							

Figure 96: Deleting a record (2)

13.4. Subtasks basic webform testing

Before testing any subtasks related tasks, subtask page should be opened from the navigation bar.

I) Adding a subtask

Test No	10
Objective	To check if new subtask can be added.
Actions	a) Click on “Add New Subtask” button b) Fill up required details with following values: ID: 243 Name: ‘New Subtask’ Start Date: 2025-03-10 Due Date: 2025-03-19 Status: Completed Task: Post-Deployment Support (select from drop down) c) Click on “Insert button”
Expected Result	A new data with filled information should be added.
Actual Result	A new data with filled information was added.
Conclusion	Test was successful.

Table 22: Adding a subtask (Test)

LS Corporation Home Users Projects Milestones Subtasks Tasks Others ▾

Subtasks Details

Add new subtask

	ID	Name	Start Date	Due Date	Completed Date	Status	Task ID	Task
<a>Edit <a>Delete	1	Define Requirements	2025-01-05	2025-01-02	2025-01-02	Completed	1	Requirement Gathering ▾
<a>Edit <a>Delete	2	Create Use Case Diagrams	2025-01-10	2025-01-06	2025-01-06	Completed	1	Requirement Gathering ▾
<a>Edit <a>Delete	3	Design ER Diagram	2025-01-13	2025-01-11		In Progress	2	Design Database Schema ▾
<a>Edit <a>Delete	4	Normalize Database Schema	2025-01-15	2025-01-14		Pending	2	Design Database Schema ▾

Figure 97: Adding a subtask (1)

LS Corporation Home Users Projects Milestones Subtasks Tasks Others ▾

Subtasks Details

ID:

243

Name:

New Subtask

Start Date:

2025-03-10

Due Date:

2025-03-19

Status:

Completed

Task:

Post-Deployment Support

Insert

Cancel

Figure 98: Adding a subtask (2)

Subtasks Details

Add new subtask

Edit

Delete

ID	Name	Start Date	Due Date	Completed Date	Status	Task ID	Task
243	New Subtask	2025-03-10	2025-03-19		Completed	15	Post-Deployment Support ▾

1234

Figure 99: Adding a subtask (3)

II) Editing a subtask

Test No	11
Objective	To check if subtasks records are editable.
Actions	a) Click on “Edit “ button of the record with ID of 243 b) Change “Name” field to ‘Old Subtask’ and add/change a completed date Field to 2025-03-30.
Expected Result	Name and completed date should be changed of the selected record.
Actual Result	Name and completed date was changed of the selected record.
Conclusion	Test was successful.

Table 23: Editing a subtask (Test)

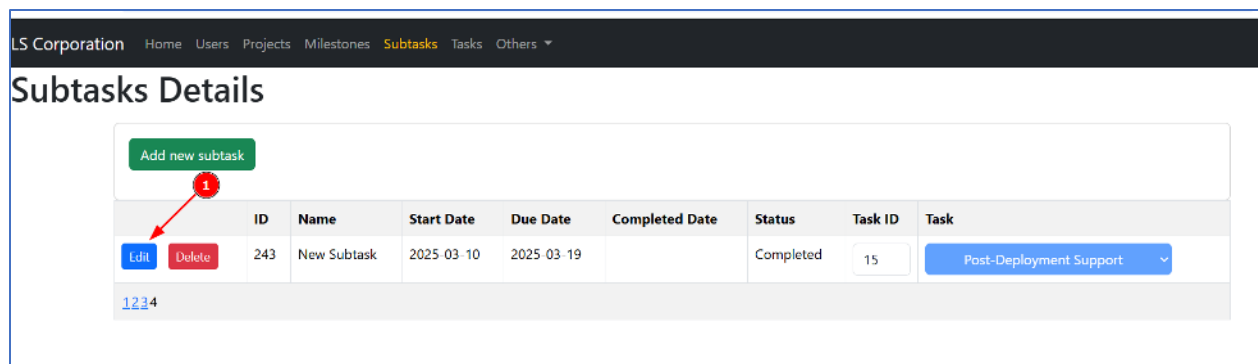


Figure 100: Editing a subtask (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Subtasks Details

Add new subtask

	ID	Name	Start Date	Due Date	Completed Date	Status	Task ID	Task
Update Cancel	243	Old Subtask	2025-03-10	2025-03-19	2025-03-30	Completed	15	Post-Deployment Supp

1234

Figure 101: Editing a subtask (2)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Subtasks Details

Add new subtask

	ID	Name	Start Date	Due Date	Completed Date	Status	Task ID	Task
Edit Delete	243	Old Subtask	2025-03-10	2025-03-19	2025-03-30	Completed	15	Post-Deployment Support

1234

Figure 102: Editing a subtask (3)

III) Deleting a subtask

Test No	12
Objective	To check if records can be delete from subtask table.
Actions	a) Click on delete button of last record with ID of 243.
Expected Result	Selected record should be deleted.
Actual Result	Selected record was deleted.
Conclusion	Test was successful.

Table 24: Deleting a subtask (Test)

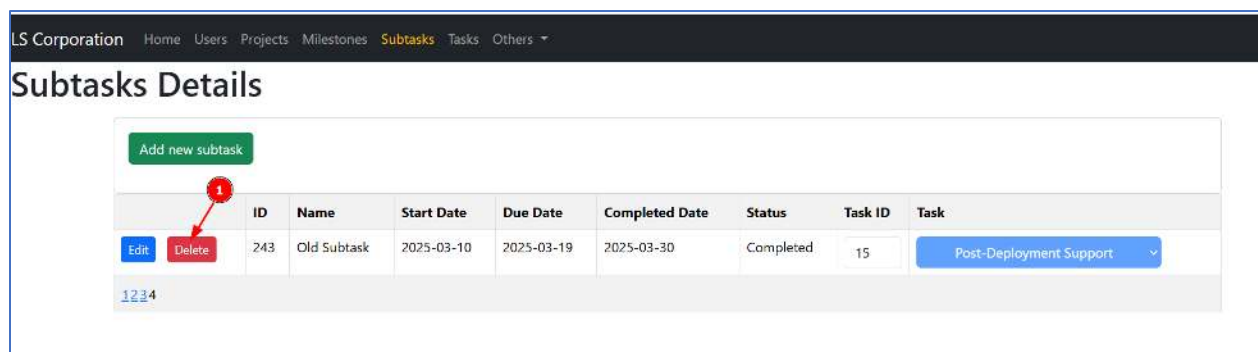


Figure 103: Deleting a subtask (1)

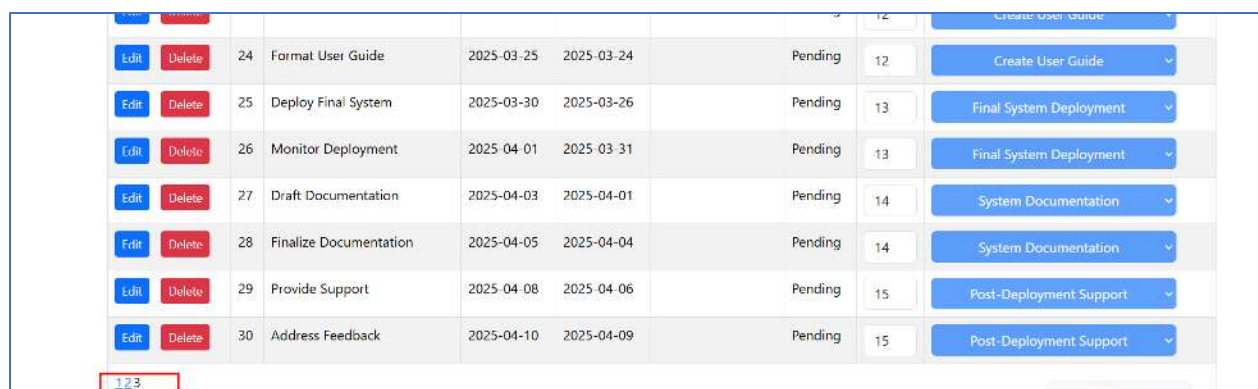


Figure 104: Deleting a subtask (2)

13.5. Tasks basic webform testing

Before testing any tasks related operations, tasks page should be opened from the navigation bar.

I) Adding a task

Test No	13
Objective	To check if new task can be added.
Actions	a) Click on “Add new task” button. b) Fill up the required fields with following values: ID → 11324 Name: ‘New Task’ Start Date: 2025-03-04 Due Date: 2025-03-18 Status: Pending (select from drop down) c) Click on “Insert button” d) Navigate to last table page
Expected Result	Newly added task should appear at the last table page.
Actual Result	Newly added task appeeared at the last table page.
Conclusion	Test was successful.

Table 25: Adding a task (Test)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Tasks Details

Add new task

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	1	Requirement Gathering	2025-01-02	2025-01-10		Pending
Edit Delete	2	Design Database Schema	2025-01-11	2025-01-15		In Progress
Edit Delete	3	Set Up Development Environment	2025-01-16	2025-01-20	2025-01-20	Completed
Edit Delete	4	Develop User Login Module	2025-01-21	2025-01-25		Pending
Edit Delete	5	Implement Payment Gateway	2025-01-26	2025-02-01		Pending
Edit Delete	6	Create Admin Dashboard	2025-02-02	2025-02-10		Pending

Figure 105: Add a task (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Tasks Details

ID:

Name:

Start Date:

Due Date:

Status:

Pending

Figure 106: Add a task (2)

LS Corporation

[Home](#)[Users](#)[Projects](#)[Milestones](#)[Subtasks](#)[Tasks](#)[Others](#)

Tasks Details

Add new task

Edit

Delete

ID	Name	Start Date	Due Date	Completed Date	Status
11324	New Task	2025-03-04	2025-03-18		Pending

1234

Figure 107: Add a task (3)

II) Editing a task

Test No	14
Objective	To check if a task is editable.
Actions	a) Click on “Edit” button of record with ID of 11324. b) Change name as ‘Old Task’, start date as 2025-03-29 and status as ‘Completed’ of the record. c) Click “Update” button
Expected Result	The data should be updated with changed values.
Actual Result	The data was updated with changed values.
Conclusion	Test was successful.

Table 26: Editing a task (Testing)

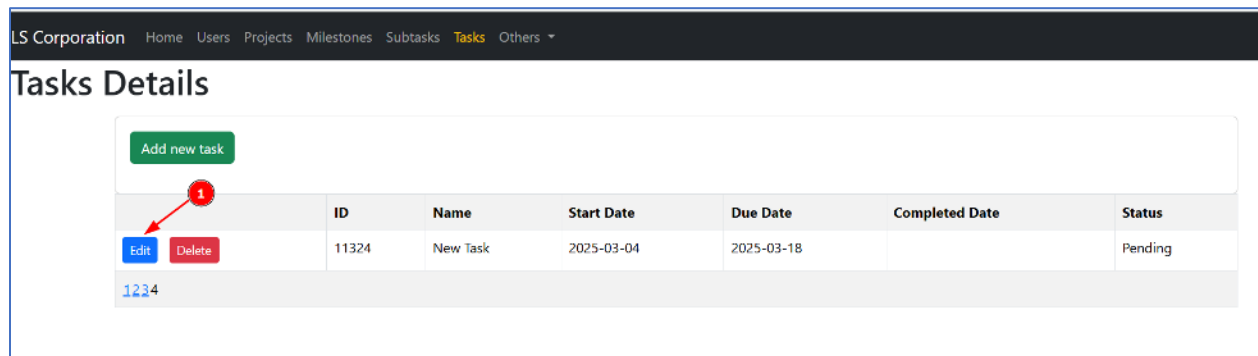


Figure 108: Editing a task (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Tasks Details

Add new task

	ID	Name	Start Date	Due Date	Completed Date	Status
Update Cancel	11324	Old task	2025-03-29	2025-03-18		Completed

1234

Figure 109: Editing a task (2)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Tasks Details

Add new task

	ID	Name	Start Date	Due Date	Completed Date	Status
Edit Delete	11324	Old task	2025-03-29	2025-03-18		Completed

1234

Figure 110: Editing a task (3)

III) Deleting a task

Test No	15
Objective	To check if a task can be deleted.
Actions	a) Click on “Delete” button with ID of 11324.
Expected Result	The selected record should be deleted
Actual Result	The selected record was deleted
Conclusion	Test was successful

Table 27: Deleting a task (Testing)

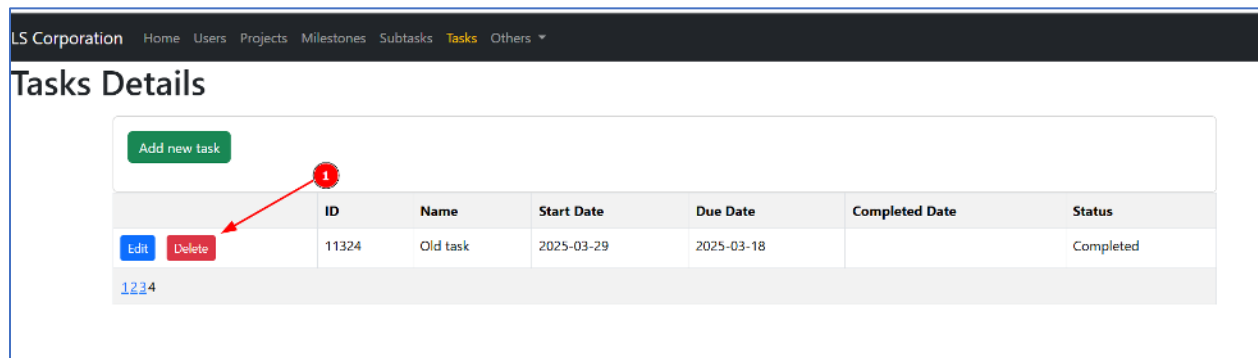


Figure 111: Deleting a task (1)

Edit	Delete	28	User Feedback Collection	2025-02-20	2025-03-01	
Edit	Delete	29	Content Management System	2025-02-25	2025-03-05	
Edit	Delete	30	Backup and Recovery Planning	2025-03-01	2025-03-10	

Figure 112: Deleting a task (2)

13.6. Top performer complex webform testing

Test No	16
Objective	To view the correct top performer of any project.
Actions	a) Navigate to “Top Performer” page via navigation bar. b) Choose “Inventory Management Sytem” c) Choose “Newly-Assigned-WebAdmin”
Expected Result	For any project top 3 performer who has done the most task should be displayed (only completed tasks).
Actual Result	Even if there were some users who completed the tasks no results were shown.
Conclusion	Test was not successful.

Table 28: Top performer (Test Failure)

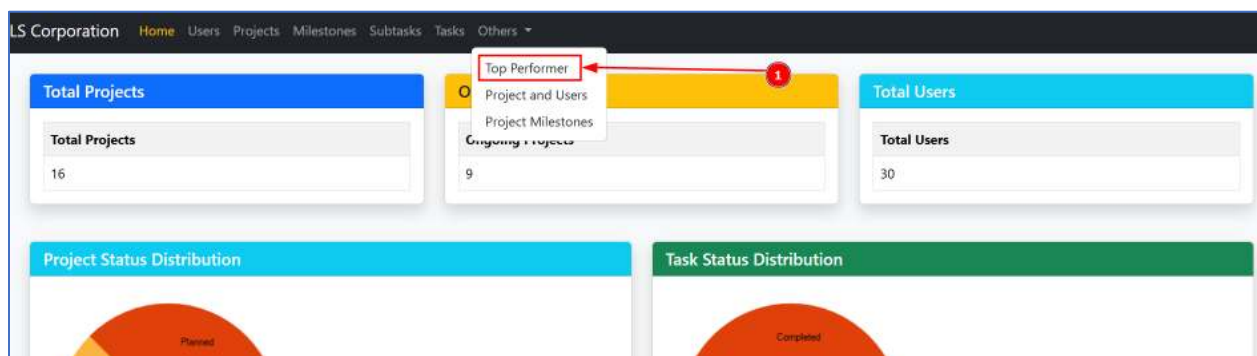


Figure 113: Top Performer (1)

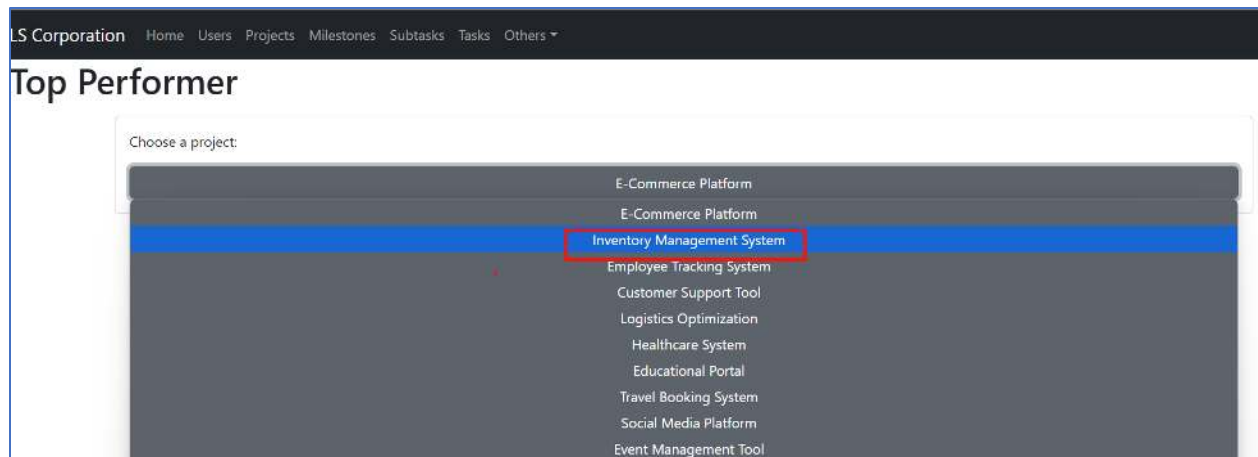


Figure 114: Top performer (2)

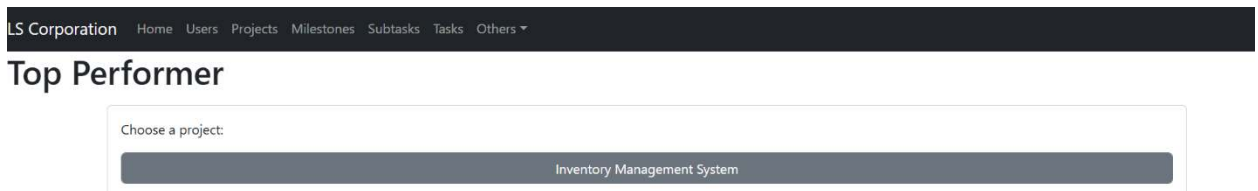


Figure 115: Top performer (3)

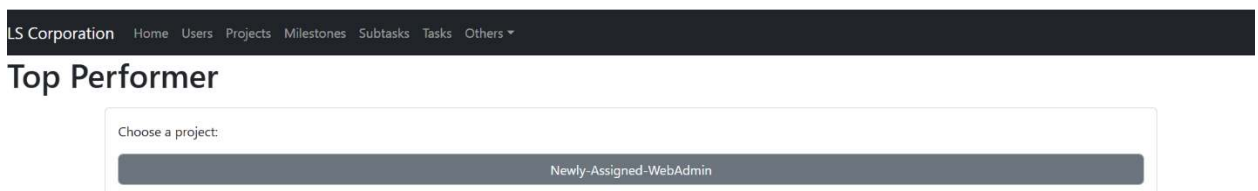


Figure 116: Top performer (4)

Correction measure: When checking for the completed task previously the sql query had this error: `SELECT taskid FROM tasks WHERE LOWER(taskstatus)=`

'Completed', this always returns false as lower function will convert all the letters to lower case where as its comparison has first letter in upper case i.e. "C". Hence, changing 'Completed' to 'completed' would solve the problem.

Test No	16
Objective	To view the correct top performer of any project.
Actions	d) Navigate to "Top Performer" page via navigation bar. e) Choose "Inventory Management Sytem" f) Choose "Newly-Assigned-WebAdmin"
Expected Result	For any project top 3 performer who has done the most task should be displayed (only completed tasks).
Actual Result	For any project top 3 perform who has don't the most task was displayed (only completed tasks).
Conclusion	Test was successful.

Table 29: Top performer (Testing)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Top Performer

Choose a project:

Inventory Management System

User ID	Username	Email	Contact	Roles	Task Count
3	Riyon Paudel	riyon.paudel@example.com	9813333333	Designer	1
4	Sudarshan Gyawali	sudarshan.gyawali@example.com	9814444444	Project Manager	1

Figure 117: Top Performer (3)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

Top Performer

Choose a project:

Inventory Management System

E-Commerce Platform
Inventory Management System
Employee Tracking System
Customer Support Tool
Logistics Optimization
Healthcare System
Educational Portal
Travel Booking System
Social Media Platform
Event Management Tool
Banking Application
Mobile Wallet App
E-Learning Platform
Retail POS System
IoT Home Automation
Newly-Assigned-WebAdmin

Figure 118: Top Performer (4)

LS Corporation

[Home](#)[Users](#)[Projects](#)[Milestones](#)[Subtasks](#)[Tasks](#)[Others](#)

Top Performer

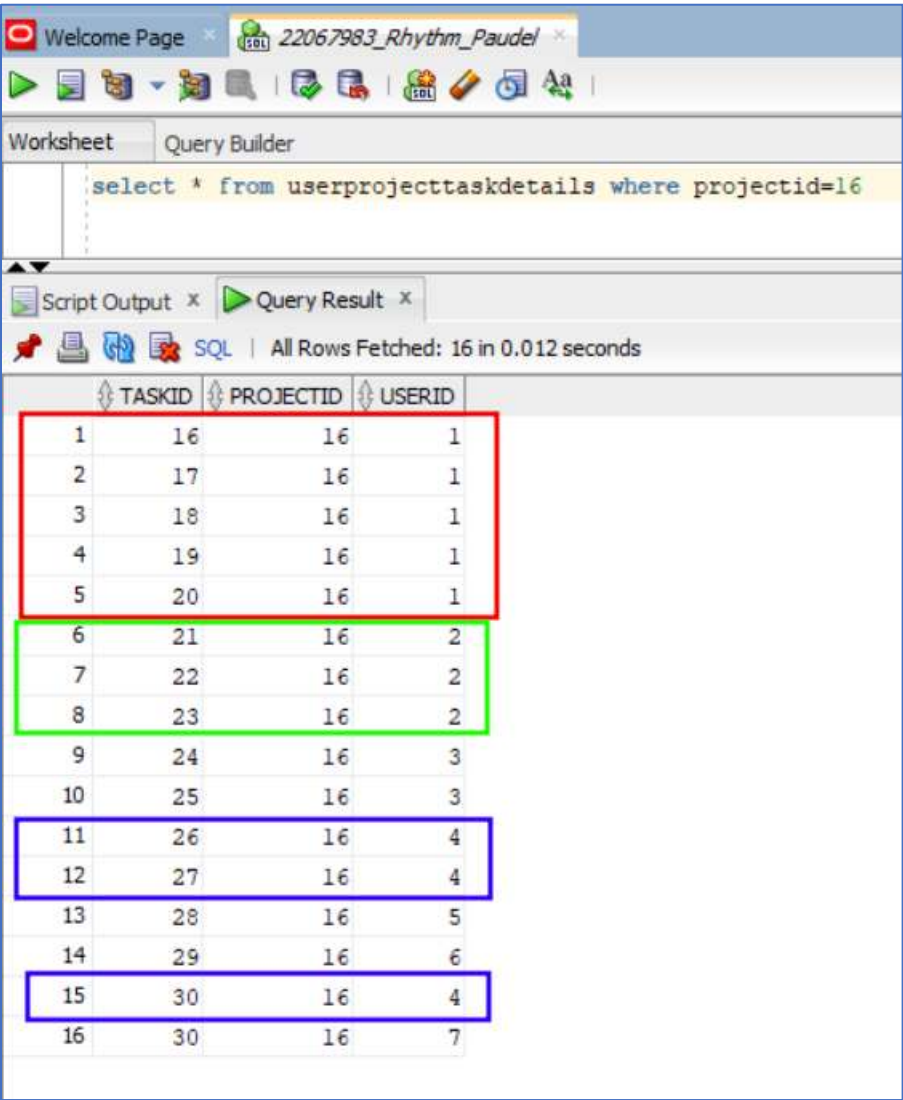
Choose a project:

Newly-Assigned-WebAdmin

User ID	Username	Email	Contact	Roles	Task Count
1	Rhythm Paudel	rhythm.paudel@example.com	9811111111	Manager	5
2	Sampada Bastola	sampada.bastola@example.com	9812222222	Developer	3
4	Sudarshan Gyawali	sudarshan.gyawali@example.com	9814444444	Project Manager	3

Figure 119: Top Performer (5)

It can be seen that users with user id 1,2 and 4 have done the most task in project 16 which is “Newly-Assigned-Webadmin” project.



Script Output x Query Result x

SQL | All Rows Fetched: 16 in 0.012 seconds

TASKID	PROJECTID	USERID
1	16	1
2	17	1
3	18	1
4	19	1
5	20	1
6	21	2
7	22	2
8	23	2
9	24	3
10	25	3
11	26	4
12	27	4
13	28	5
14	29	6
15	30	4
16	30	7

Figure 120: Top performer (6)

13.7. User and Project complex webform testing

Test No	17
Objective	To check if the user and their involvement in projects are being shown properly
Actions	a) Navigate to “Project and Users” page. (By default already a user is selected) b) Change a user to view their active projects.
Expected Result	As user changes project should also change unless they are enrolled in same projects.
Actual Result	As user changed project was also changed.
Conclusion	Test was successful.

Table 30: User and Project (Testing)

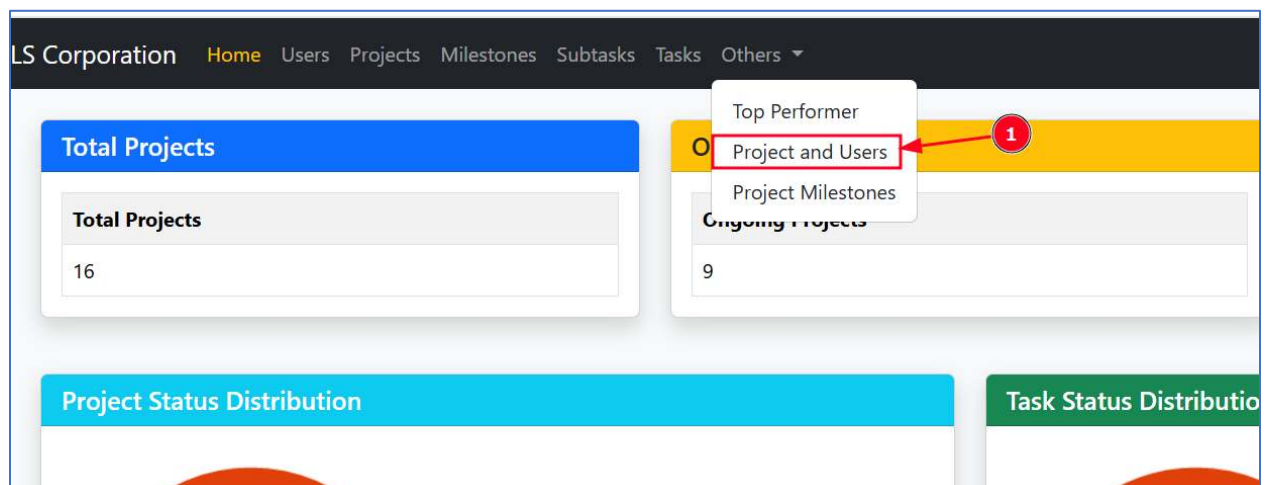


Figure 121: User and project (1)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

User Projects

Choose a user:

Rhythm Paudel

User ID	Username	Email	Contact	ROLENAME
1	Rhythm Paudel	rhythm.paudel@example.com	9811111111	Manager

Project ID	Project Name	Start date	Due Date	Status
1	E-Commerce Platform	2025-01-01	2025-03-31	Ongoing
16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12	Ongoing

Figure 122: User and project (2)

LS Corporation
Home
Users
Projects
Milestones
Subtasks
Tasks
Others

User Projects

Choose a user:

Rhythm Paudel
Rhythm Paudel
Sampada Bastola
Riyon Paudel
Sudarshan Gyawali
Gitu Paudel
Dhruba Paudel
Aliza Shrestha
Sudin Thapa
Samir Thapaliya
Ram Dhakal
Bishal Prasai
Risab Bhandari

Figure 123: User and project (3)

LS Corporation

[Home](#)[Users](#)[Projects](#)[Milestones](#)[Subtasks](#)[Tasks](#)[Others](#)

User Projects

Choose a user:

Dhruba Paudel

User ID	Username	Email	Contact	ROLENAME
6	Dhruba Paudel	dhruba.paudel@example.com	9816666666	Quality Analyst

Project ID	Project Name	Start date	Due Date	Status
3	Employee Tracking System	2025-02-01	2025-05-01	Ongoing
16	Newly-Assigned-WebAdmin	2025-01-01	2025-03-12	Ongoing

Figure 124: User and project (4)

13.8. Project and milestones

Test No	18
Objective	To check if project details and milestone details are displayed properly.
Actions	a) Navigate to “Project Milestones” from the navigation bar. (A default project is already selected.) b) Change a project
Expected Result	Milestones should be changed respective to the project when the project changes.
Actual Result	Milestones changed according to related projects.
Conclusion	The test was successful.

Table 31: Project and milestones (Test)

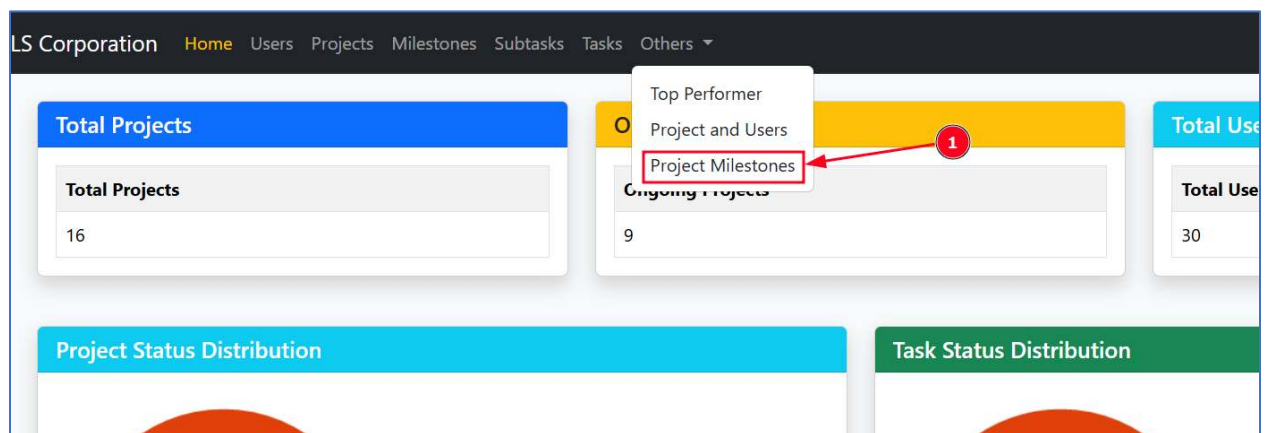


Figure 125: Project and milestones (1)

LS Corporation

[Home](#)
[Users](#)
[Projects](#)
[Milestones](#)
[Subtasks](#)
[Tasks](#)
[Others](#)

Project Milestones

Choose a project:

E-Commerce Platform

Project ID	Project Name	Start date	Due Date	Status
1	E-Commerce Platform	2025-01-01	2025-03-31	Ongoing

Milestone ID	Milestone name	Priority	Due Date of Milestone
1	Prototype Completion	High	2025-01-20
2	Beta Release	Medium	2025-02-15
3	Final Deployment	High	2025-03-31

Figure 126: Project and milestones (2)

LS Corporation

[Home](#)
[Users](#)
[Projects](#)
[Milestones](#)
[Subtasks](#)
[Tasks](#)
[Others](#)

Project Milestones

Choose a project:

Inventory Management System

E-Commerce Platform

Inventory Management System

Employee Tracking System

Customer Support Tool

Logistics Optimization

Healthcare System

Educational Portal

Travel Booking System

Social Media Platform

Event Management Tool

Banking Application

Mobile Wallet App

E-Learning Platform

Retail POS System

IoT Home Automation

Newly-Assigned-WebAdmin

Figure 127: Project and milestones (3)

LS Corporation

[Home](#) [Users](#) [Projects](#) [Milestones](#) [Subtasks](#) [Tasks](#) [Others](#)

Project Milestones

Choose a project:

Inventory Management System

Project ID	Project Name	Start date	Due Date	Status
2	Inventory Management System	2025-01-15	2025-04-15	Planned

Milestone ID	Milestone name	Priority	Due Date of Milestone
4	System Architecture Design	High	2025-02-01
5	Database Integration	Medium	2025-03-01
6	Final Testing	High	2025-04-15

Figure 128: Project and milestones (4)

13.9. Data validation in ProjectDetails

Test No	19
Objective	To check if the application crashes if when wrong data type is provided because of unknown date format.
Actions	a) Click on “Add new project” on the project page. b) On start data and due date, enter random letters fill out the rest of the form with following values: Project ID → 3432 Project Name → asd Start Date → ‘fhjk’ Due Date → ‘dfgh’ Status → ‘pending’ (select from drop down) c) Click on the “Insert” button
Expected Result	The application handles the error and does not cause the system to crash.
Actual Result	The web application crashed.
Conclusion	The test was not successful.

Table 32: Data Validation in ProjectDetails (Test Failure)

LS Corporation Home Users **Projects** Milestones Subtasks Tasks Others ▾

Project Details

Project ID:
3432

Project Name:
asd

Start Date:
fhjk

Due Date:
dfgh

Status:
pending

Insert Cancel

ID	Name	Start Date	Due Date	Completed Date	Status
----	------	------------	----------	----------------	--------

Figure 129: Incorrect data type entry (1)

Server Error in '/' Application.

The string was not recognized as a valid DateTime. There is an unknown word starting at index 0.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.FormatException: The string was not recognized as a valid DateTime. There is an unknown word starting at index 0.

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[FormatException: The string was not recognized as a valid DateTime. There is an unknown word starting at index 0.]
  System.DateTimeParse.Parse(String s, DateTimeFormatInfo dtfi, DateTimeStyles styles) +6394719
  System.Convert.ToDateTime(String value, IFormatProvider provider) +68
  System.Convert.ChangeType(Object value, TypeCode typeCode, IFormatProvider provider) +733
  System.Web.UI.WebControls.Parameter.GetValue(Object value, String defaultValue, TypeCode type, Boolean convertEmptyStringToNull, Boolean ignoreNullableTypeChanges) +89
  System.Web.UI.WebControls.SqlDataSourceView.AddParameters(DbCommand command, ParameterCollection reference, IDictionary parameters, IDictionary exclusions) +367
  System.Web.UI.WebControls.SqlDataSourceView.ExecuteInsert(IDictionary values) +179
  System.Web.UI.DataSourceView.Insert(IDictionary values, DataSourceViewOperationCallback callback) +502
  System.Web.UI.WebControls.FormView.HandleInsert(String commandArg, Boolean causesValidation) +725
  System.Web.UI.WebControls.FormView.HandleEvent(EventArgs e, Boolean causesValidation, String validationGroup) +50
  System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +135
  System.Web.UI.WebControls.FormViewRow.OnBubbleEvent(Object source, EventArgs e) +50
  System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +50
  System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5440
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.9282.0

Figure 130: Incorrect data type entry (2)

Correction Measure: Changing the text field to a calendar field, so the user can select a date from it. Previously, the user was not prompted with calendar which may rise confusions in date formats.

Test No	20
Objective	To check if the application crashes if when wrong data type is provided because of unknown date format.
Actions	a) Click on “Add new project” on the project page. b) Click on Start Date and Due Date
Expected Result	The user should be prompted with a calendar when clicking on a date field.
Actual Result	The user was prompted with a calendar when clicking on a date field.
Conclusion	The test was successful. This way user does not have to know the format of date and directly press the date on the prompter calendar.

Table 33: Data Validation in ProjectDetails (Testing)

Project Details

Project ID:

Project Name:

Start Date:

March, 2025

Su	Mo	Tu	We	Th	Fr	Sa
23	24	25	26	27	28	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today: March 16, 2025

	ID	Name	Start Date
Edit Delete	1	E-Commerce Platform	2025-01-01

Figure 131: Data validation in ProjectDetails

13.10. Invalid foreign key testing

Test No	21
Objective	To check if the application handles the error when provided with the wrong foreign key.
Actions	a) Navigate to the “MilestoneDetails” page. b) Fill out all the fields with the following values with improper foreign key Milestone ID: 123 Milestone Name: ‘ASDFASDF’ Due Date: 2025-03-03 Project ID: 10000 c) Enter “Insert” button
Expected Result	The system should handle the error properly.
Actual Result	The application crashed after an improper value.
Conclusion	The test was not successful.

Table 34: Invalid foreign key testing (failure)

Milestones Details

Milestone ID:

Milestone Name:

Priority:

Due Date:

Project ID:

	ID	Name	Priority	Due Date	Project ID
Edit Delete	1	Prototype Completion	High	2025-01-20	1
Edit Delete	2	Beta Release	Medium	2025-02-15	1

Figure 132: Milestones Details Improper foreign key (1)

Server Error in '/' Application.

ORA-02291: integrity constraint (22067983_Rhythm_Paudel.TABLE_9_PROJECTDETAILS_FK) violated - parent key not found

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.OracleClient.OracleException: ORA-02291: integrity constraint (22067983_Rhythm_Paudel.TABLE_9_PROJECTDETAILS_FK) violated - parent key not found

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[OracleException (0x80131938): ORA-02291: integrity constraint (22067983_Rhythm_Paudel.TABLE_9_PROJECTDETAILS_FK) violated - parent key not found]
System.Data.OracleClient.OracleConnection.CheckError(OciErrorHandle errorHandle, Int32 rc) +404
System.Data.OracleClient.OracleCommand.Execute(OciStatementHandle statementHandle, CommandBehavior behavior, Boolean needRowid, OciRowidDescriptor& rowid
```

Figure 133: Milestones Details Improper foreign key (2)

Correction Measure: Instead of letting the user input any number, a drop-down list with existing projects should be shown to prevent this.

Test No	22
Objective	To check if the application handles the error when provided with the wrong foreign key.

Actions	a) Navigate to the “MilestoneDetails” page. b) Click on Project ID field
Expected Result	The system should let the user select a project from the dropdown instead of typing project id to prevent application crash.
Actual Result	The system lets the user select a project from the dropdown to prevent application crash.
Conclusion	The test was successful.

Table 35: Milestones Details foreign key handling (Testing)

	ID	Name	Priority	Due Date	Project ID
Edit Delete	1	Prototype Completion	High	2025-01-20	1
Edit Delete	2	Beta Release	Medium	2025-02-15	1

Figure 134: Project name drop down

14. Further Discussion

When starting this project, mixed thoughts were in the mind. The requirements of the project seemed easy. However, doing a project in completely new programming languages with a variety of unknown tools raised a question of difficulty. Resources available online were very few for this project which further created problems when errors occurred. However, a variety of tools and techniques were learned when developing the project.

SQL Data Modeler was learned to map tables that reduced the complexities for creating tables and defining relationships between tables. Furthermore, time was also saved because of the DDL or table creation code provided by the application.

ASP.NET Web form was also learned, which helped in designing the front-end part for accessing and manipulating the data from the database. Most components were dragged and dropped, and I added some external CSS for beauty, which was very useful for a beginner in this framework. ASP.NET Web form helped me learn a new language in an effective way.

Database normalization was also learned and applied in the practical scenario, which was helpful for understanding the concept of normalization. Up to the third normal form, the data was normalized, which gave a clear understanding of how to manage the database in the most efficient way and how to work on improving redundancy.

A popular framework, AJAX Control toolkit, was implemented for implementation for calendars in certain fields in webform, which also taught about its usage and how to use it.

Proper optimization of queries was also learned when creating complex webforms. Various queries could result in the same outcome, but the importance of the optimization was understood.

15. Conclusion

A database was designed with a given scenario and a record of a user. Several steps were followed to achieve a well-structured database. Several assumptions were also made during the database designing process. After the designing of the database, it was further normalized to be more optimized and efficient.

Following these steps, a web application with ASP.NET was built using webforms. This web application was connected to the database that was created. Several actions like editing, deleting, adding, or viewing various types of data could be performed from the web application. The web application also offers a dashboard that displays the overall progress of the projects, milestones, and tasks.

The project made is used to perform basic CRUD (Create, Read, Update and Delete) operations, which also provides a well optimized database.

16. References

Belic, I. & Stryker, C., 2024. *What is an entity relationship diagram?* | IBM. [Online]
Available at: <https://www.ibm.com/think/topics/entity-relationship-diagram>
[Accessed 14 March 2025].

Chen, L. et al., 2024. *Database normalization description - Microsoft 365 Apps* | Microsoft Learn. [Online]
Available at: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>
[Accessed 14 March 2025].

UC MERCED LIBRARY, 2025. *What is a data dictionary?* | UC MERCED LIBRARY. [Online]
Available at: <https://library.ucmerced.edu/data-dictionaries>
[Accessed 15 March 2025].