Module Code & Module Title

CC6012NT – Data and Web Development

Assessment Weightage & Type

4st Milestone (individual Coursework 40%)

Year and Semester

2024-25 Autumn

Student Name: Aashish Neupane

London Met ID: 22072025

College ID: NP05CP4A220004

Assignment Due Date: 17th Jan 2025

Assignment Submission Date: 17th Jan 2025

I confirm that I understand my coursework needs to be submitted online via My second teacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table Of Contents

1 Introduction 1

1.1 Aims: 1

1.2 Objectives: 1

1.3 Business Rules 2

1.3.1 User and Project: 2

1.3.2 User and Task 2

1.3.3 Project and Task: 2

1.3.4 Task and Sub_task: 2

1.3.5 Task and Resource: 3

1.3.6 User and Comment: 3

1.3.7 Task and Comment: 3

2 Identification of Entities and Attributes: 4

2.1 Relationships 7

2.2 Initial ERD: 8

3 Normalization 9

3.1 UNF (Un-Normalized Form): 9

3.2 1 NF (First Normalized Form): 10

3.3 2 NF (Second Normalized Form) 11

3.4 3 NF (Third Normalized Form): 13

4 Final ERD 15

5 Implementation 17

5.1 Connect System and Creating User 17

5.2 Connecting user with SQL Developer 18

5.3 Creation of Table 19
5.3.1 Project_details Table 19
5.3.2 User_details Table: 21
5.3.3 Project_Info Table: 23
5.3.4 Task_details Table: 25
5.3.5 Task_Info Table 27
5.3.6 Sub_task_details Table 30
5.3.7 Milestone_details Table 32
6 Basic Web Forms 34
6.1 Task Tracking Dashboard 34
6.2 User_details Web Form 35
6.3 Project_details Web Form 36
6.4 Task_details Web Form 37
6.5 Sub_task_details Web Form 38
6.6 Milestone_details Web Form 39
7 Complex Web Form 40
7.1 Updated Dashboard 40
7.2 User Project (for any user, show details of the user and the details of all the projects that he/she has been working or has worked on. 41
7.3 Project Milestone (For any Project, show the detail of the project with all the milestone of the project) 42
7.4 Top Performer (for any project, show the details of top 3 user who has done the most task) 44
8 Conclusion 45
9 References 46

Table of Figure

Figure 1: Relationships among Entities 7
Figure 2: Initial ERD diagram 8
Figure 3: Final ERD as per Normalization 15
Figure 4: Final ERD from Modeler 16
Figure 5: Connect system and creating user 17
Figure 6: showing connection with SQL Developer 18
Figure 7: Creating Project_details Table 19
Figure 8: DESC Project_details Table 19
Figure 9: Inserting Value in Project_details table 20
Figure 10: Showing Project_details value 20
Figure 11: Creating User_details table 21
Figure 12: DESC User_details table 21
Figure 13: Inserting User_details value 22
Figure 14: Showing User_details Values 22
Figure 15: Creating Project_Info table 23
Figure 16: DESC Project_Info table 23
Figure 17: Inserting values in Project_Info table 23
Figure 18: Showing Project_Info values 24
Figure 19: Inserting Values in Task_details table 25
Figure 20: DESC Task_detailsTable 25
Figure 21: Inserting values in Task_details table 26
Figure 22: Showing values in Task_details values 26
Figure 23: creating Task_Info table 27
Figure 24: DESC Task_Info table 27
Figure 25: Inserting values in Task_Info Table; 28
Figure 26: Showing Task_Info values 29
Figure 27: Creating Sub_task_details table 30
Figure 28: DESC Sub_task_details 30
Figure 29: Inserting values in Sub_task_Info table 31
Figure 30: Showing Sub_task_Info Values 31

Figure 31: Creating Milestone_details table 32
Figure 32: DESC Milestone_details table 32
Figure 33: Inserting values in Milestone_details table 33
Figure 34: Showing Milestone_details values 33
Figure 35: Main Dashboard 34
Figure 36: User_details Web Form 35
Figure 37: Project_details Web Form 36
Figure 38: Task_details Web Form 37
Figure 39: Sub_task_details Web Form 38
Figure 40: Milestone_Details Web Form 39
Figure 41: Updated Dashboard 40
Figure 42: Before showing details of user and their associated Project 41
Figure 43: After showing the details of User and their project 41
Figure 44: Before showing the projects details and all their milestones 42
Figure 45: After choosing the Project and output as milestone 43
Figure 46: Before Selecting the Project with top performer 44
Figure 47: After selecting the Project with top performer 44
Figure 48: Similarity Report 47

1 Introduction

LS Corporation which is a mid-size technology company has struggled to manage multiple project due to fragmented system which includes the problems like poor communication, inefficiencies and difficult in tracking the progress. As we know that, the effective project management is very essential in any types of business mainly in those technology firms where communication and precision are key for their success.

So, to overcome these issues, LS corporation want to develop a comprehensive project management system that aimed at centralizing data, optimizing work allocations and improve project visibility. This helps company staff like project manager, developer and designer to work effectively and collaboratively that offers many features like multi-project management, resources allocation and so on. The system main priority is to focus on milestone tracking so that the project can divides into manageable phases that include prototype completion, beta release and others with due date.

The prototype for this system will be build using Oracle SQL Developer Data Modeler for data modeling and ASP.NET with C# for application development. This approach makes our system user-friendly, provide more scalable solution so that there will be easy to make changes for this company.

1.1 Aims:

The main aim of the coursework is to design and develop a centralized data, streamline task and robust system for LS corporation that increase efficiency in operations, make better communication and provide best visibility to make project progress.

1.2 Objectives:

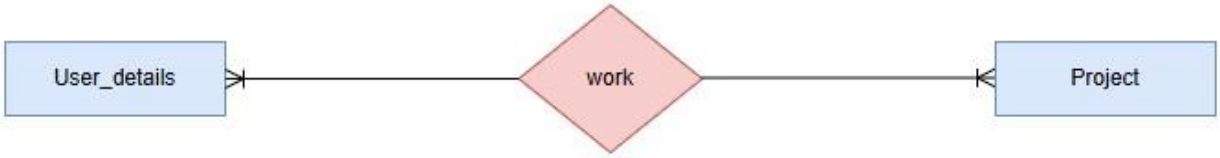
- Developing a problem-solving database system solution for LS Corporation.
- Develop a system to eradicate decentralized data storage issues that is currently present in the organization.
- Develop a simple CRUD operation friendly database system.
- Develop a final database system that meets the organization’s expectation.

1.3 Business Rules

Some business rules are provided below for the successful operation of this system:

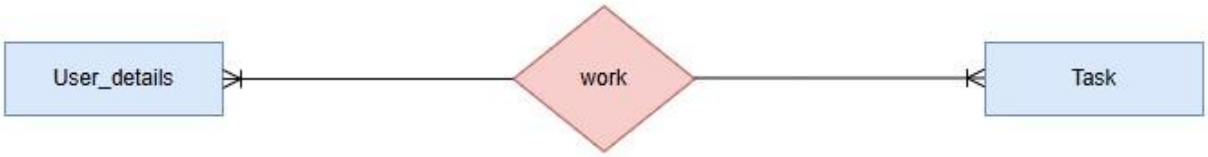
1.3.1 User and Project:

There will be many-to-many relationship between User and Project. A user can manage multiple projects, and one project is associated with many users.



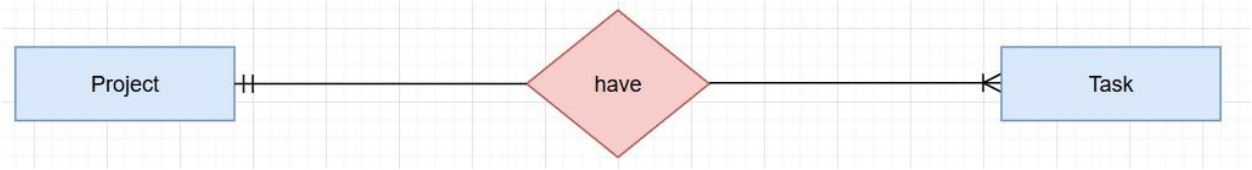
1.3.2 User and Task

There is Many-to-Many relationship between user and task. A user can perform multiple tasks and, each task can have multiple users.



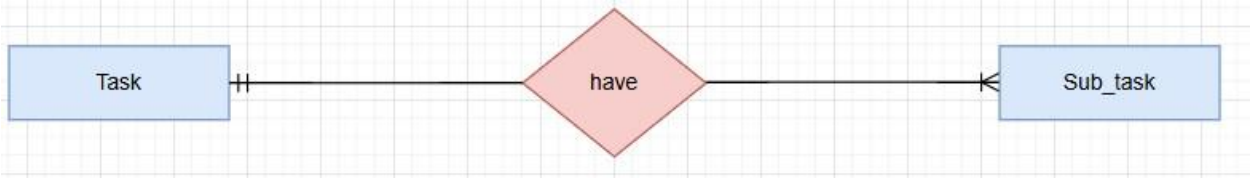
1.3.3 Project and Task:

There will be one-to-one relationship between Project and Task. A project can have multiple tasks, but one task is related with one project.



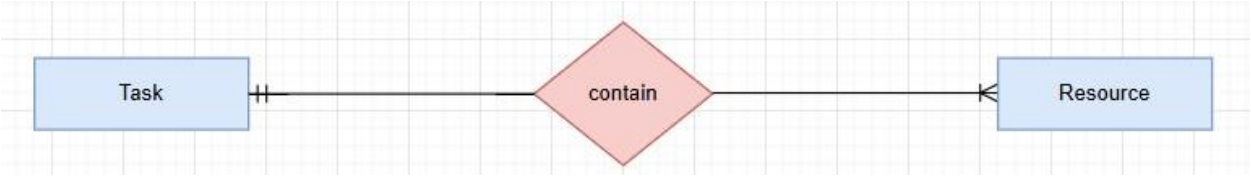
1.3.4 Task and Sub_task:

There is one-to-many relationships between task and sub_task. A task can have many sub_task but one sub_task is associated one task.



1.3.5 Task and Resource:

There is one-to-many relationship between Task and Resource. Here, one task is associated with multiple resources but, one resource is related with one task.

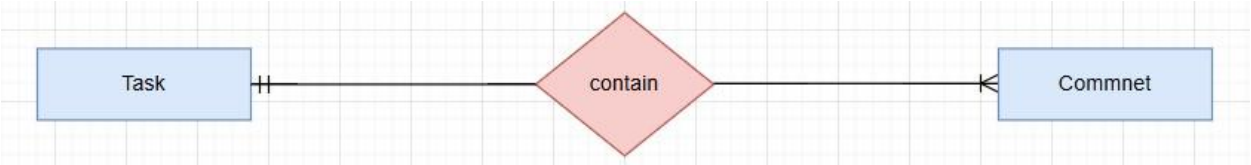


1.3.6 User and Comment:

There will be one-to-many relationship between user and comment. Here, one user can have multiple comment but, one comment is related to a user.

1.3.7 Task and Comment:

There will be one-to-many relationship between Task and Comment. One task can have multiple comment, but a comment is associated with a task.



2 Identification of Entities and Attributes:

The Entity Relationship Model is a high-level data model that describes the conceptual representation of a database. This ER model is considered a best choice for database design as it deals with global entities and their relationships between entities. It contains three main components:

I) Entity: Entity can be defined as any things or object that can be uniquely recognized in the real world. For instance: person, products, organizations etc.)

II) Attributes: All the entities are represented by its properties called attributes. In general, the attributes which are accessible to an entity are determined by its type or subtype. (Mondal, 2014)

III) Relationships: Relationships describe how the association between two entity types are related. For examples: a mother loves her baby. In this case, 'Loves' is the relationships. Relationships allows datasets to share and store data in several tables. (Soni, 2024)

As the discussion of business rules is already done above. So, looking at that the required attributes of each entity are presented below:

1	**Attributes of User_details**
a. User_id	- Unique identifier for each user
b. name	- User's Name
c. email	- User's Email
d. Address	- User's Address
e. contact	- User's Contact

2	Attributes of Project
a. project_id	- Unique identifier for each project
b. project_name	- Project's name
c. project_start_date	- Project's start date
d. project_end_date	- Project's end date
e. project_status	- Project's status
f. milestone_name	- Milestone Name
g. milestone_description	- Milestone Description
h. milestone_start_date	- Milestone start date
i. milestone_end_date	- Milestone end date

3	Attributes of Task
a. task _id	- Unique identifier for each task
b. task _name	- Task name
c. task _description	- Description of task
d. task_start_date	- Start date for task
e. task_end_date	- End date for task
f. task_status	- Task status

4	Attributes of Sub_task
a. sub_task_id	- Unique identifier for each sub_task
b. sub_task_name	- Sub task name
c. sub_task_desc	- Sub task description
d. sub_task_start_date	- Sub task start date

e. sub_task_end_date	- Sub task end date
f. sub_task_status	- Sub task status

5	**Attributes of Resource**
a. resource_id	- Unique identifier for each resource
b. resource_name	- name of resource
c. resource_type	- resource type
d. resource_description	- Description of task

6	**Attributes of Comment**
a. comment_id	- Unique identifier for each comment
b. comment_details	- details of comment
c. comment_type	- type of comment
d. comment_date	- date of comment

2.1 Relationships

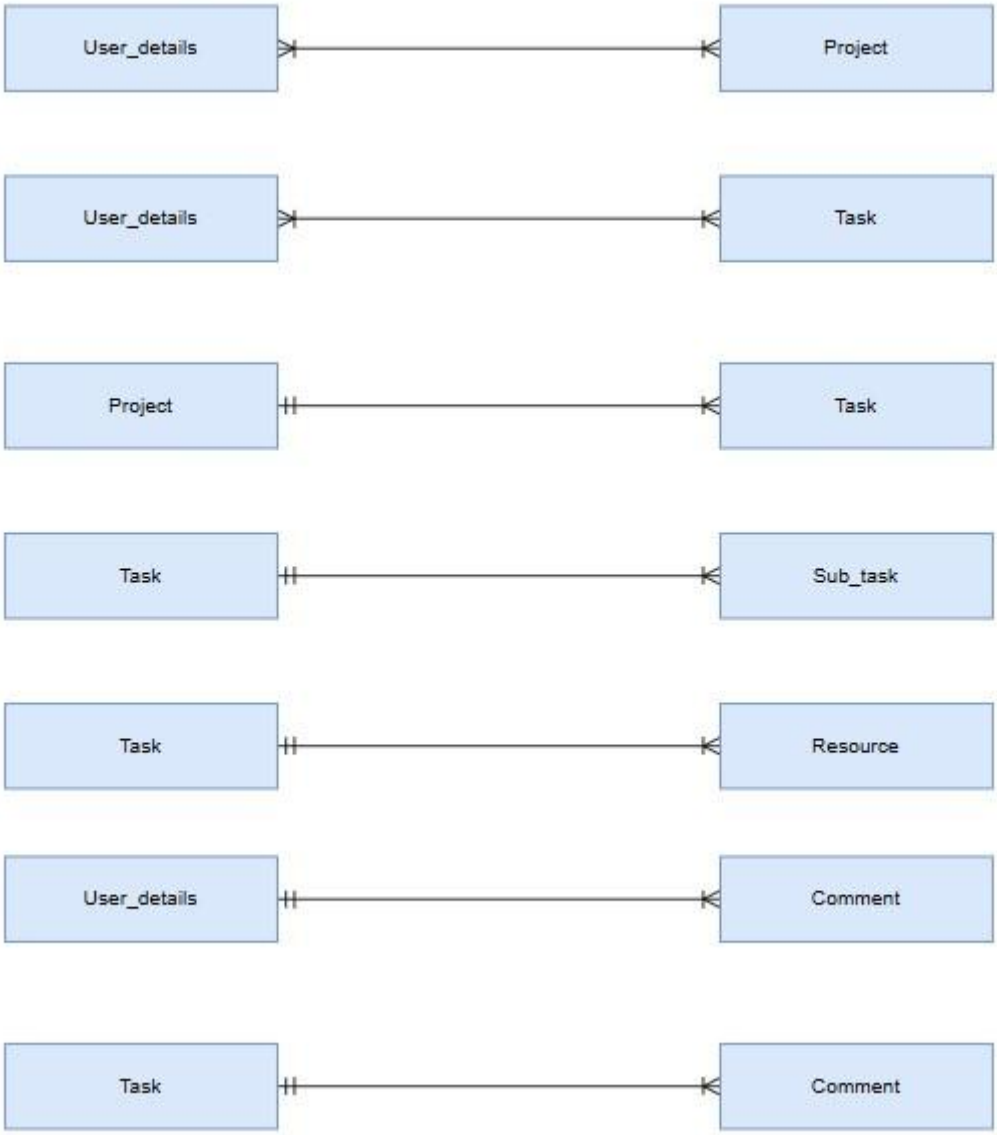


Figure 1: Relationships among Entities

2.2 Initial ERD:

The initial Entity Relationship Diagram required for the system is shown below where we collect Six entities: User, Project, Task, Sub_task, Resource and Comment. We connect these entities by relationships which reflect the properties of LS Corporation. Also, the entities have their attributes which describe the features of each entity and the relationships among them respectively.

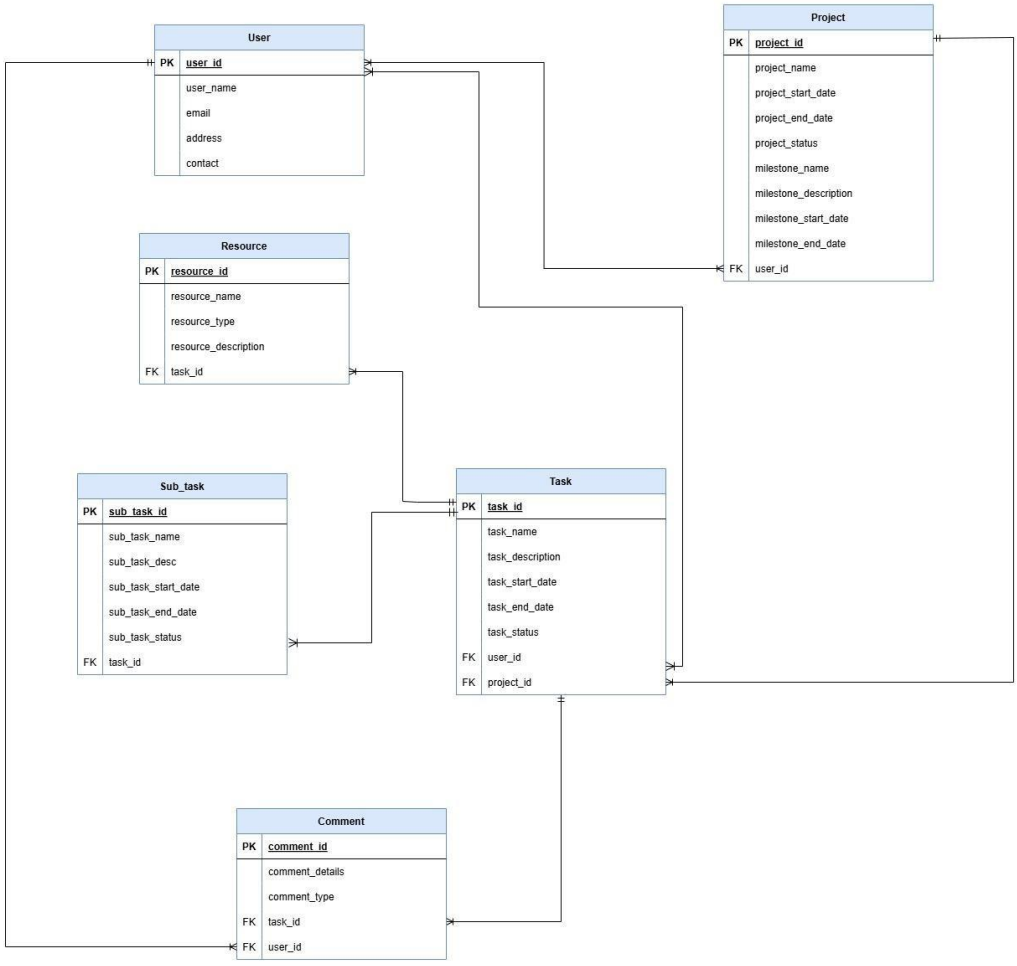


Figure 2: Initial ERD diagram

3 Normalization

In simple word, the way of arranging data into tables in such a way that queries executed on the database always accurate and reasonable results. Also, this is the way of getting data into simple forms which defines the separate entities and their respective attributes and relationships between them. (Margaret Rouse, 2024)

There are different Normalization degree of relational database tables and provided below:

3.1 UNF (Un-Normalized Form):

This In the first stage of the Normalization the task is to choose one unique identifier and proper mention all the attributes for the entity. Additionally, Curly braces '{ }' is for showing repeating group respectively.

Here, I am illustrating repeating data and repeating groups of my projects:

Assumption:

A single user can be a part of many projects and, one project is associated with many users. A user is assigned multiple tasks, and every task can be related to multiple users. One project can have multiple tasks, but one task is related with single project. A project has multiple milestones but, one milestone is related one project. Each task can have multiple sub tasks, but one sub task is related to one task.

User_details (**user_id**, user_name, email, address, contact, { project_id, project_name, project_start_date, project_end_date, project_status, { milestone_name, milestone_description, milestone_start_date, milestone_end_date}, { task_id, task_name, task_description, task_start_date, task_end_date, task_status, {sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status }}})

3.2 1 NF (First Normalized Form):

This is the first degree of normalization in we removed the repeating groups and separate relation which makes simplicity and Consistent access. It also defines a Primary Key for each relation.

The 1 NF process of the system is provided below:

Here, we separate the repeating groups from non-repeating groups, and the result is:

User_details (**user_id**, user_name, email, address, contact)

Project (**project_id**, project_name, project_start_date, project_end_date, project_status, **user_id*)**

Milestone_details (**milestone_id**, milestone_name, milestone_description, milestone_start_date, milestone_end_date, **project_id***)

Task (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status, **project_Id*, user_id***)

Sub_task (**sub_task_id**, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, **task_id***)

Final 1- NF:

User_details -1 (**user_id**, user_name, email, address, contact)

Project -1 (**project_id**, project_name, project_start_date, project_end_date, project_status, **user_id*)**

Milestone_details -1 (**milestone_id**, milestone_name, milestone_description, milestone_start_date, milestone_end_date, **project_id***)

Task -1 (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status, **project_Id*, user_id***)

Sub_task-1 (**sub_task_id**, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, **task_id***)

3.3 2 NF (Second Normalized Form)

This is one of the most important stages of normalization because we check for the partial dependency and separates entities in two groups based on partial and full dependency and all non-key attributes are fully functional dependent on the primary key respectively.

The 2 NF process of my project is shown below:

Checking partial dependency on Project table:

project_id, user_id

project_id ⟹ project_name, project_sart_date, project_end_Date, ⟹ project_status

user_id

⟹

The output after these partial dependencies is:

Project_info (user_id*, project_id*)

Project_details (project_id, project_name, project_sart_date, project_end_Date, project_status)

Checking partial dependency on Task table

task_id, project_id, ⟹

user_id ⟹

project_id ⟹

user_id

task_id ⟹ task_name, task_description, task_start_date, task_end_date, task_status

The output after these partial dependencies is:

Task_details (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status)

Task_info (**task_id***, **project_id*, user_id***)

Final – 2NF:

User_details -2 (**user_id**, user_name, email, address, contact)

Project_details -2 (**project_id**, project_name, project_start_date, project_end_date, project_status)

Project_info -2 (**user_id***, **project_id***)

Task_details-2 (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status)

Task_info -2 (**task_id***, **project_id*, user_id***)

Sub_task -2 (**sub_task_id**, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, **task_id***)

Milestine_details -2 (**milestone_id**, milestone_name, milestone_description, milestone_start_date, milestone_end_date, **project_id***)

3.4 3 NF (Third Normalized Form):

This is the final stages of Normalization where we establish the primary key and ensure that all non-key attributes are fully functional dependent on primary key. Then, we look after the transitive dependency, which is defined as attributes depends on other non-key attributes, to handle that we establish a new table for each group of independent attributes and use foreign key to construct links between each table respectively.

As per above solution, there are not any transitive dependencies for the final stages of Normalization,

The 3 NF process of this system is provided below:

User_details (**user_id**, user_name, email, address, contact)

Project_details (**project_id**, project_name, project_start_date, project_end_date, project_status)

Project_info (**user_id***, **project_id***)

Task_details (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status)

Task_info (**task_id***, **project_id*, user_id***)

Sub_task (**sub_task_id**, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, **task_id***)

Milestine_details (**milestone_id,** milestone_name, milestone_description, milestone_start_date, milestone_end_date, **project_id***)

Final – 3 NF:

User_details -3 (**user_id**, user_name, email, address, contact)

Project_details -3 (**project_id**, project_name, project_start_date, project_end_date, project_status)

Project_info -3 (**user_id***, **project_id***)

Task_details -3 (**task_id**, task_name, task_description, task_start_date, task_end_date, task_status)

Task_info -3 (**task_id***, **project_id*, user_id***)

Sub_task -3 (**sub_task_id**, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, **task_id***)

Milestine_details -3 (**milestone_id,** milestone_name, milestone_description, milestone_start_date, milestone_end_date, **project_id***)

4 Final ERD

At, first the initial ERD was sketched in which in follow the case study and completed drawing the necessary entities, attributes and their relations between them. Then, the next process i.e. Normalization where we normalize the table to reduce data redundancy and minimize anomalies that can occurs while inserting, deleting or updating the data in the table. Now, the next step is to draw the Final ERD which is obtained from the normalization.

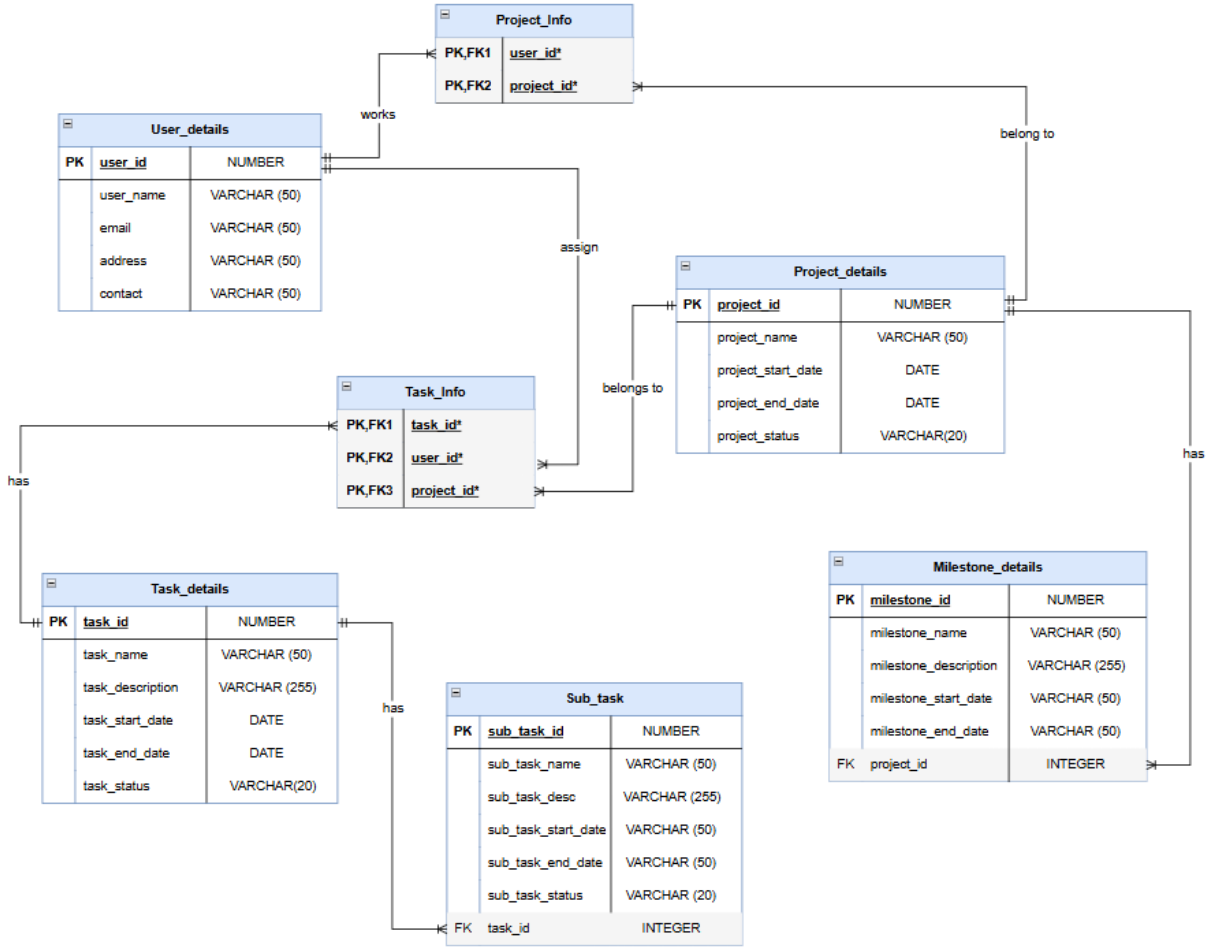


Figure 3: Final ERD as per Normalization

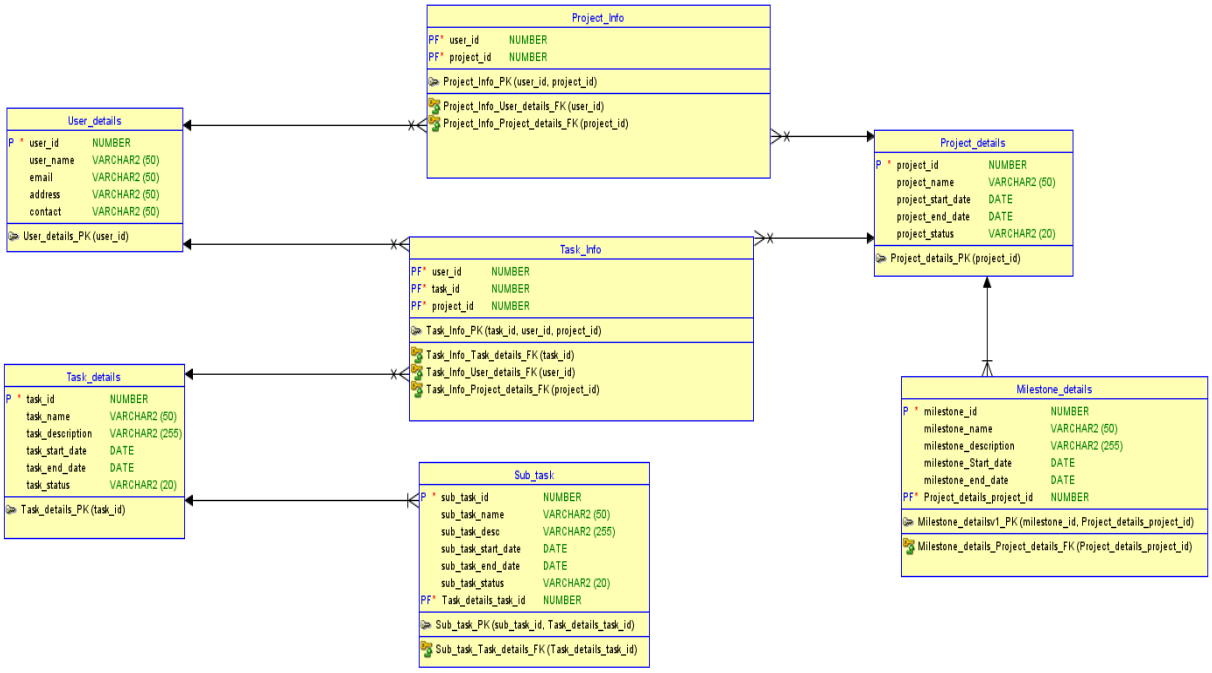


Figure 4: Final ERD from Modeler

5 Implementation

5.1 Connect System and Creating User

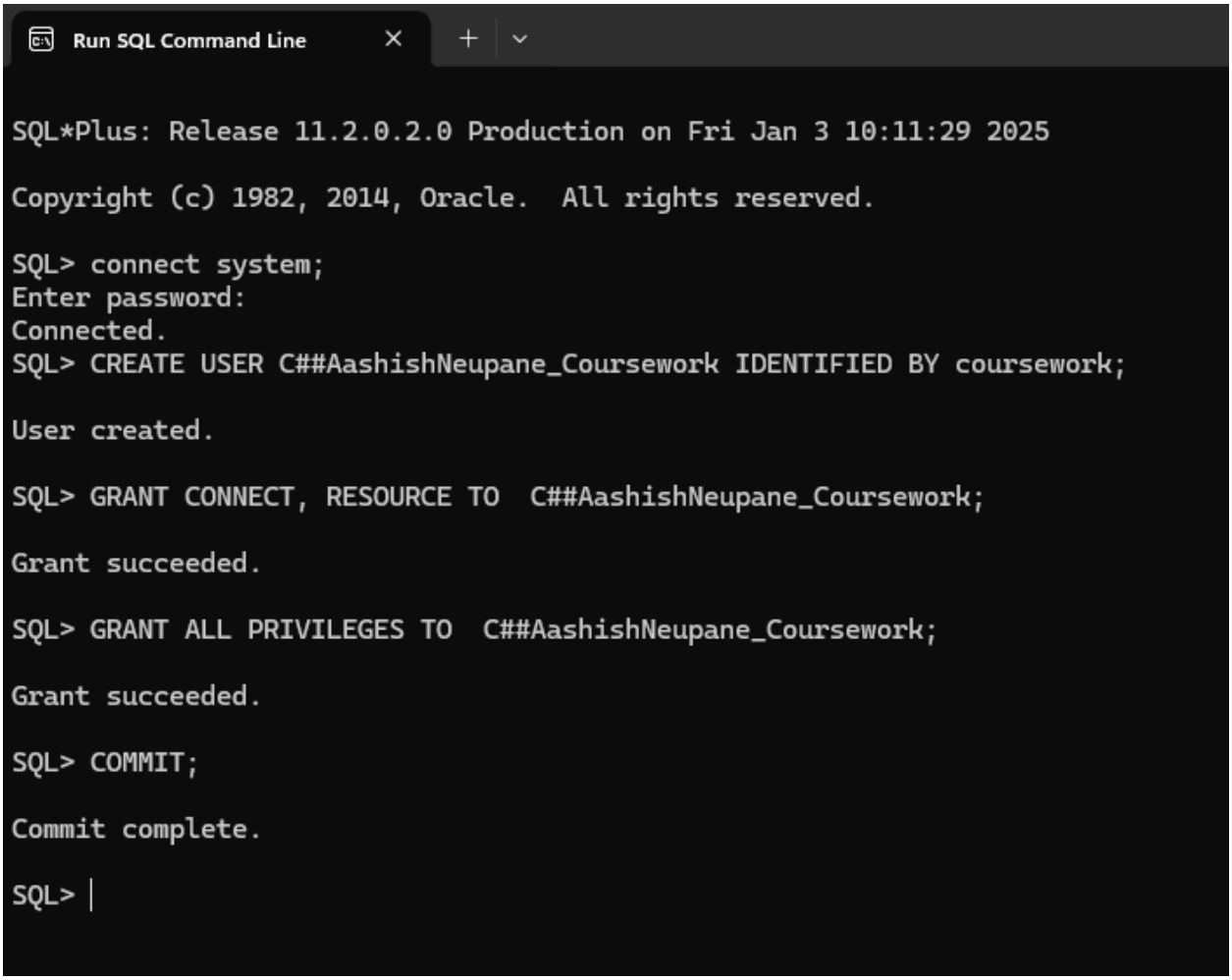


Figure 5: Connect system and creating user

5.2 Connecting user with SQL Developer

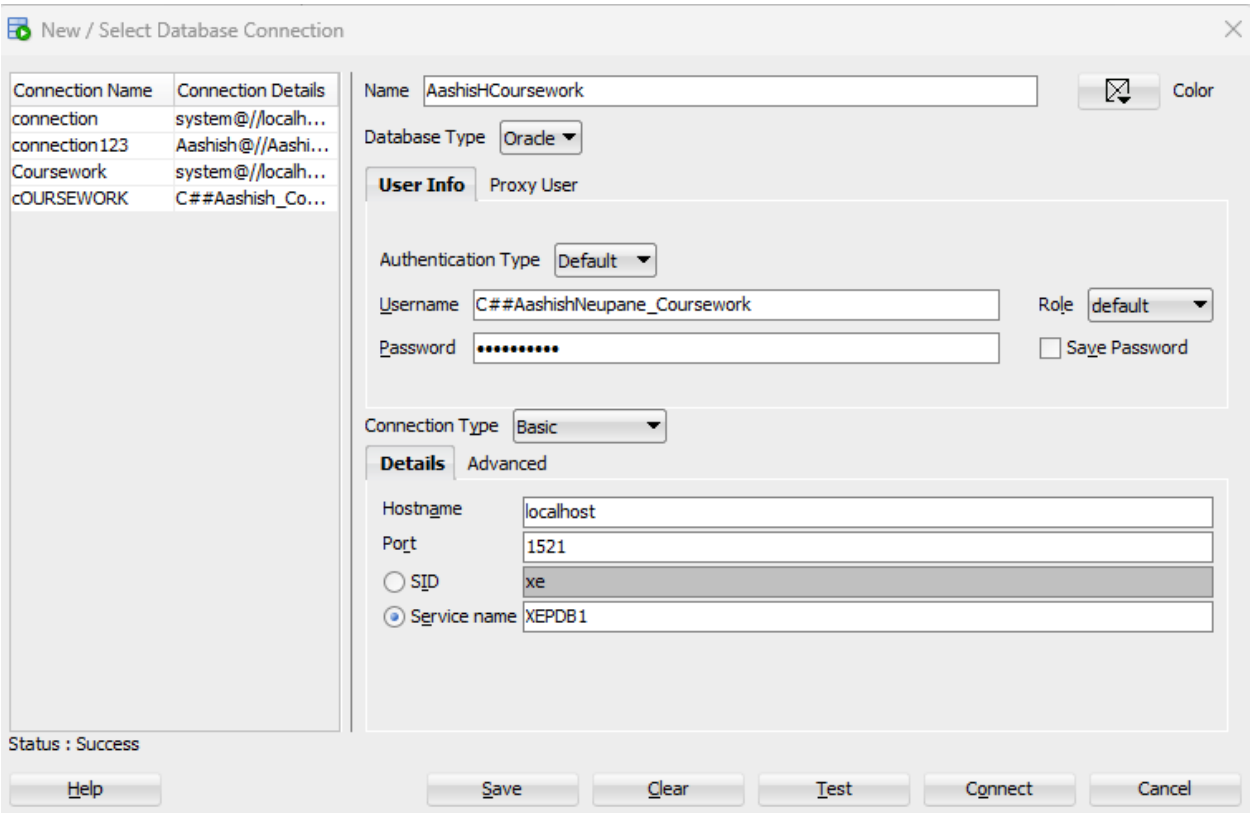


Figure 6: showing connection with SQL Developer

5.3 Creation of Table

5.3.1 Project_details Table

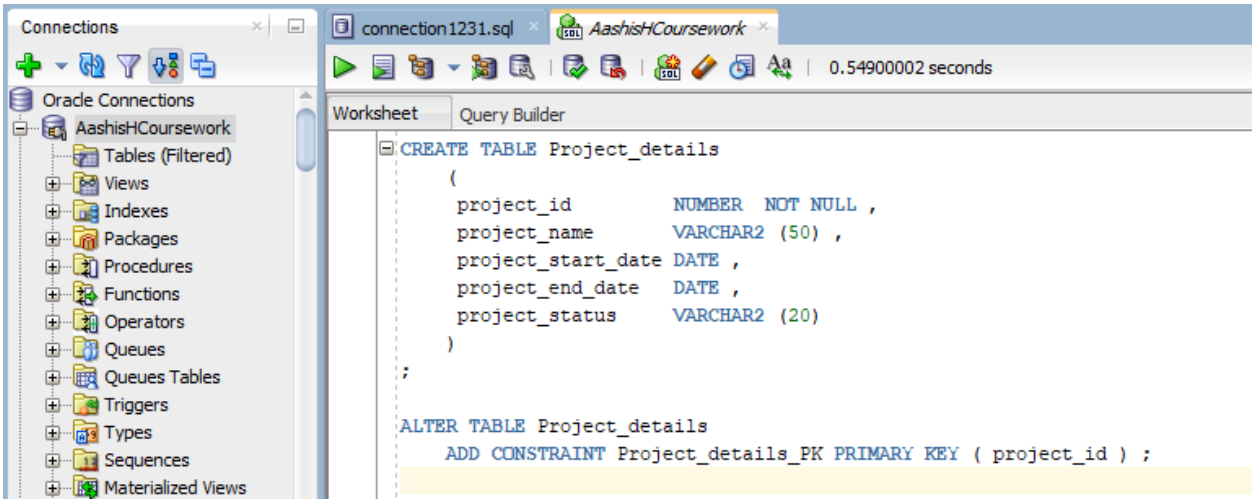


Figure 7: Creating Project_details Table

```
------------------ -------- ------------
PROJECT_ID         NOT NULL NUMBER
PROJECT_NAME                VARCHAR2(50)
PROJECT_START_DATE          DATE
PROJECT_END_DATE            DATE
PROJECT_STATUS              VARCHAR2(20)
```

Figure 8: DESC Project_details Table

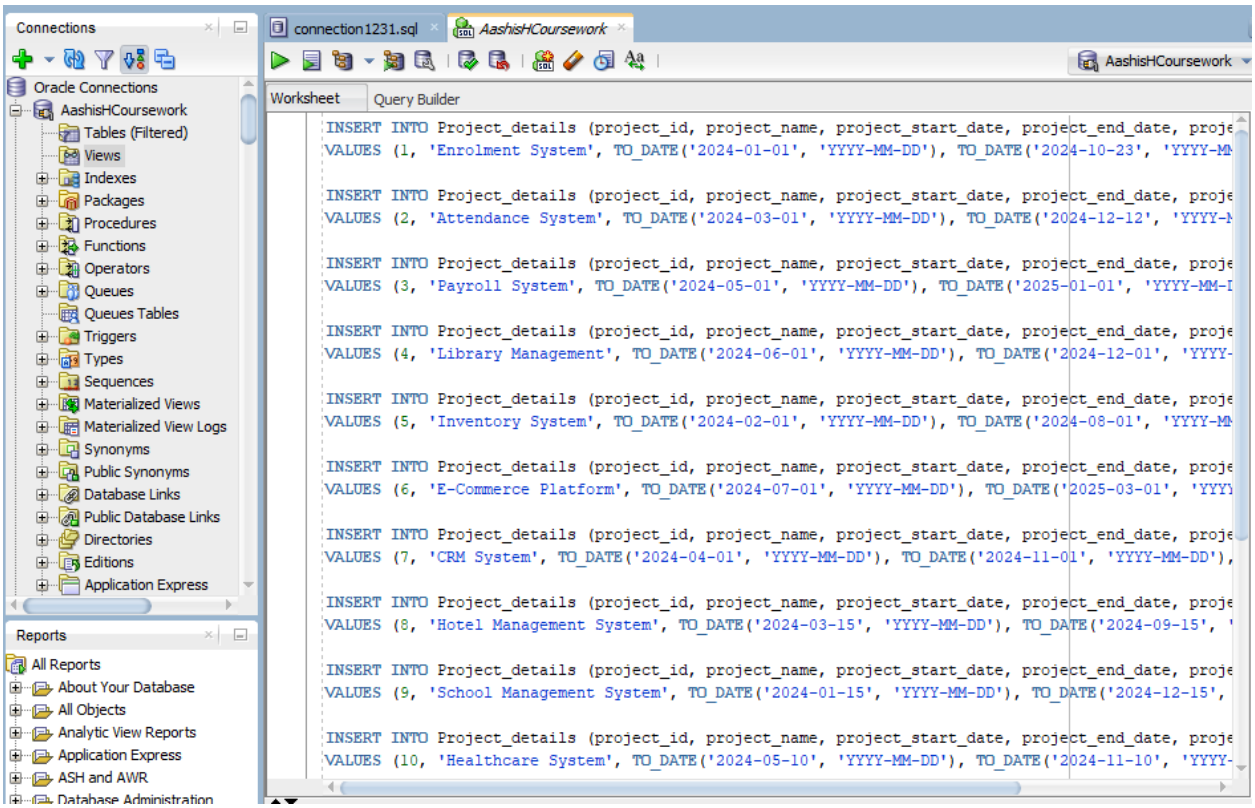
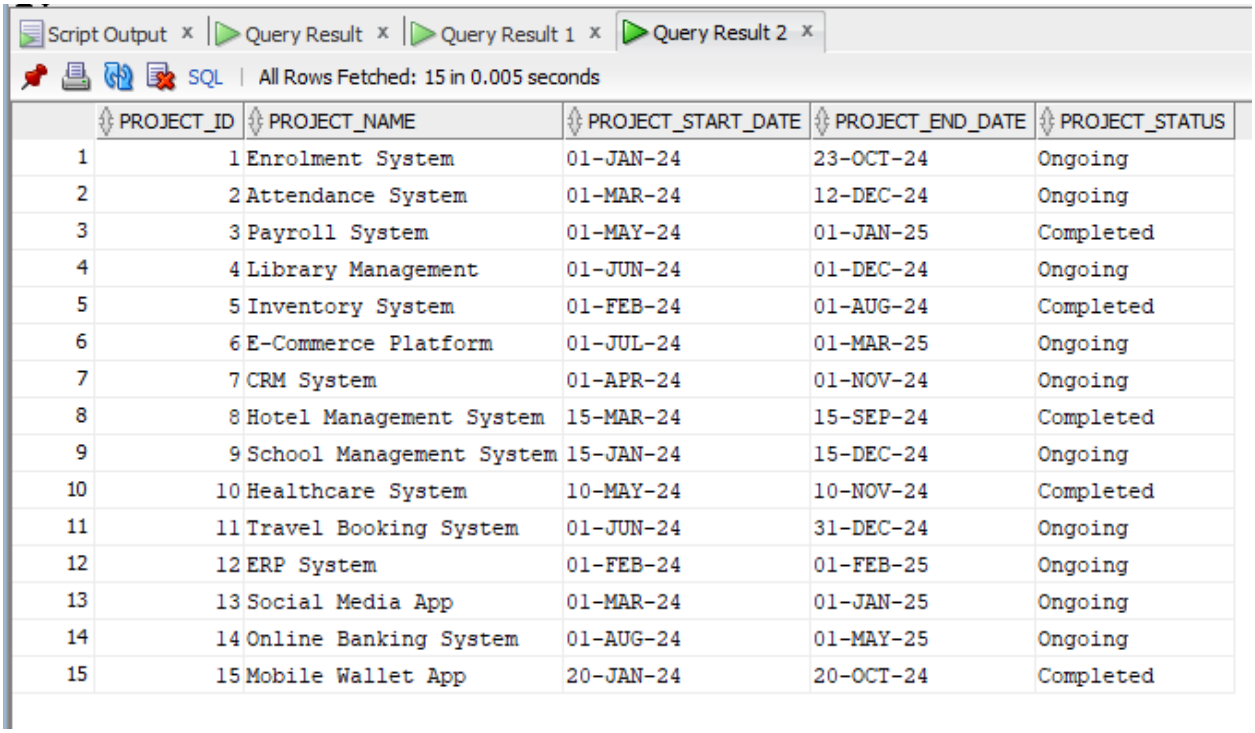


Figure 9: Inserting Value in Project_details table



	PROJECT_ID	PROJECT_NAME	PROJECT_START_DATE	PROJECT_END_DATE	PROJECT_STATUS
1	1	Enrolment System	01-JAN-24	23-OCT-24	Ongoing
2	2	Attendance System	01-MAR-24	12-DEC-24	Ongoing
3	3	Payroll System	01-MAY-24	01-JAN-25	Completed
4	4	Library Management	01-JUN-24	01-DEC-24	Ongoing
5	5	Inventory System	01-FEB-24	01-AUG-24	Completed
6	6	E-Commerce Platform	01-JUL-24	01-MAR-25	Ongoing
7	7	CRM System	01-APR-24	01-NOV-24	Ongoing
8	8	Hotel Management System	15-MAR-24	15-SEP-24	Completed
9	9	School Management System	15-JAN-24	15-DEC-24	Ongoing
10	10	Healthcare System	10-MAY-24	10-NOV-24	Completed
11	11	Travel Booking System	01-JUN-24	31-DEC-24	Ongoing
12	12	ERP System	01-FEB-24	01-FEB-25	Ongoing
13	13	Social Media App	01-MAR-24	01-JAN-25	Ongoing
14	14	Online Banking System	01-AUG-24	01-MAY-25	Ongoing
15	15	Mobile Wallet App	20-JAN-24	20-OCT-24	Completed

Figure 10: Showing Project_details value

5.3.2 User_details Table:

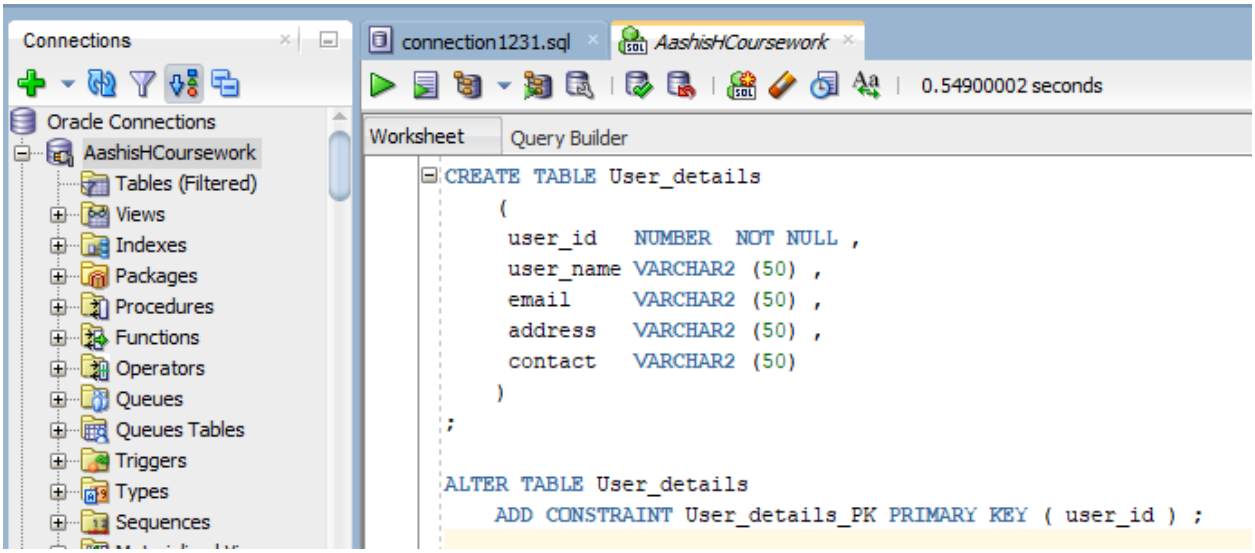


Figure 11: Creating User_details table

```
Name          Null?    Type
------------- -------- ------------
USER_ID       NOT NULL NUMBER
USER_NAME              VARCHAR2(50)
EMAIL                  VARCHAR2(50)
ADDRESS                VARCHAR2(50)
CONTACT                VARCHAR2(50)
```

Figure 12: DESC User_details table

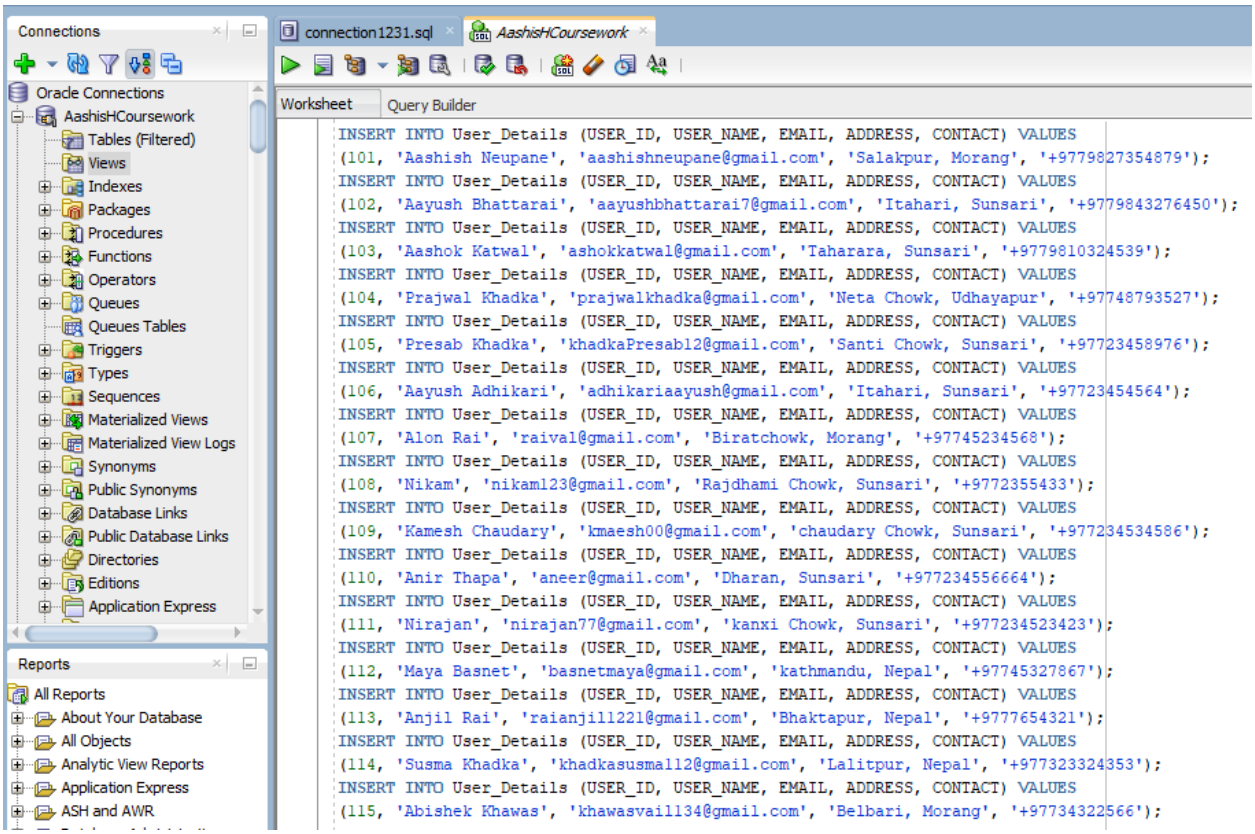
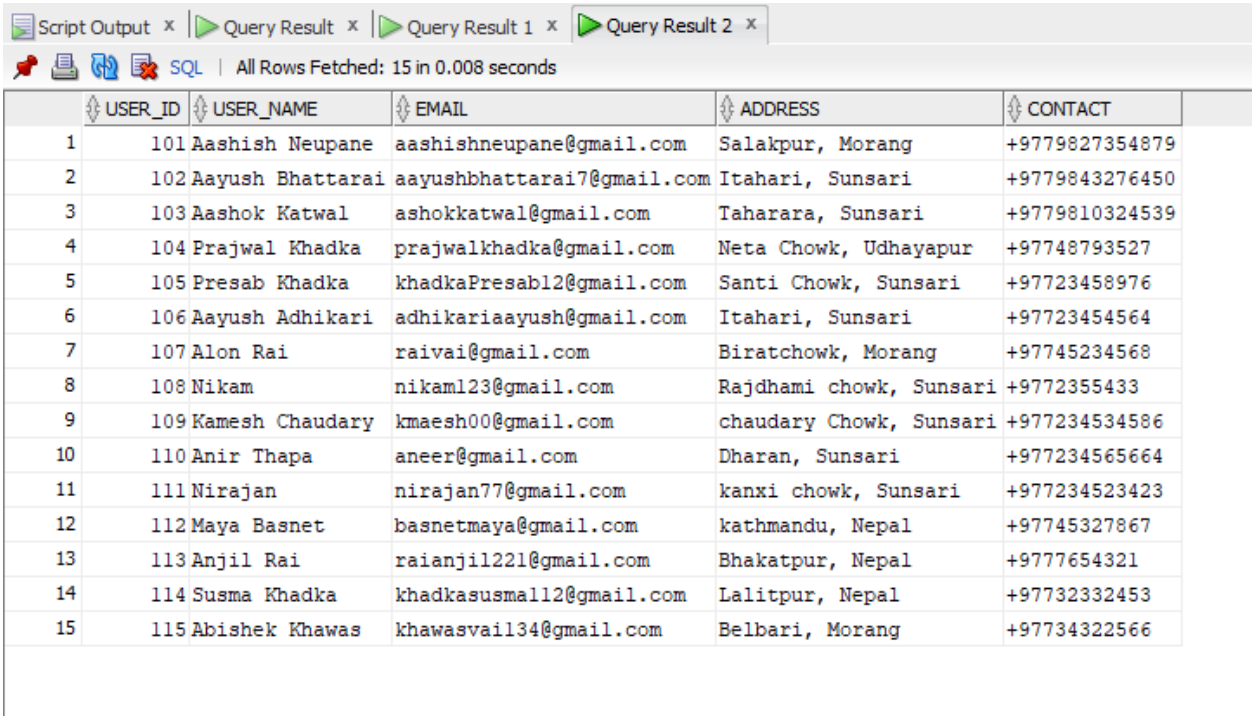


Figure 13: Inserting User_details value



	USER_ID	USER_NAME	EMAIL	ADDRESS	CONTACT
1	101	Aashish Neupane	aashishneupane@gmail.com	Salakpur, Morang	+9779827354879
2	102	Aayush Bhattarai	aayushbhattarai7@gmail.com	Itahari, Sunsari	+9779843276450
3	103	Aashok Katwal	ashokkatwal@gmail.com	Taharara, Sunsari	+9779810324539
4	104	Prajwal Khadka	prajwalkhadka@gmail.com	Neta Chowk, Udhayapur	+97748793527
5	105	Presab Khadka	khadkaPresab12@gmail.com	Santi Chowk, Sunsari	+97723458976
6	106	Aayush Adhikari	adhikariaayush@gmail.com	Itahari, Sunsari	+97723454564
7	107	Alon Rai	raivai@gmail.com	Biratchowk, Morang	+97745234568
8	108	Nikam	nikam123@gmail.com	Rajdhami chowk, Sunsari	+9772355433
9	109	Kamesh Chaudary	kmaesh00@gmail.com	chaudary Chowk, Sunsari	+977234534586
10	110	Anir Thapa	aneer@gmail.com	Dharan, Sunsari	+977234565664
11	111	Nirajan	nirajan77@gmail.com	kanxi chowk, Sunsari	+977234523423
12	112	Maya Basnet	basnetmaya@gmail.com	kathmandu, Nepal	+97745327867
13	113	Anjil Rai	raianjil221@gmail.com	Bhakatpur, Nepal	+9777654321
14	114	Susma Khadka	khadkasusmal12@gmail.com	Lalitpur, Nepal	+97732332453
15	115	Abishek Khawas	khawasvail34@gmail.com	Belbari, Morang	+97734322566

Figure 14: Showing User_details Values

5.3.3 Project_Info Table:

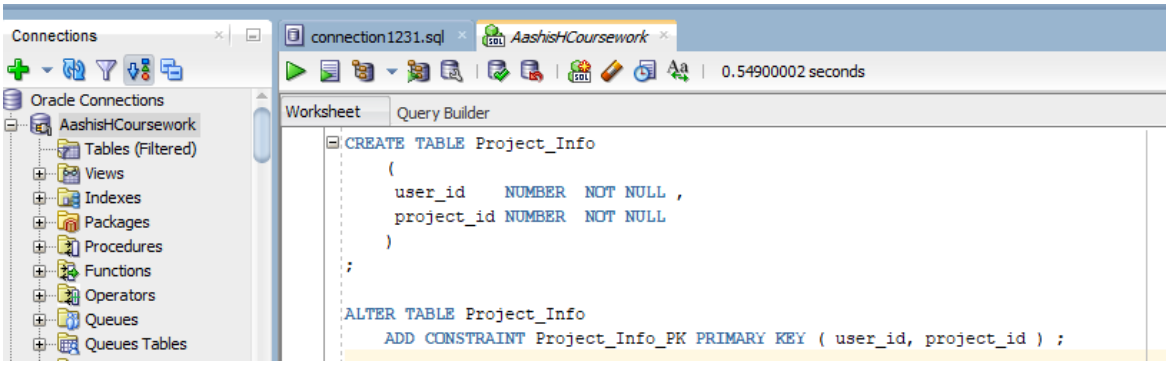


Figure 15: Creating Project_Info table

```
Name          Null?    Type
------------- -------- ------
USER_ID       NOT NULL NUMBER
PROJECT_ID    NOT NULL NUMBER
```

Figure 16: DESC Project_Info table

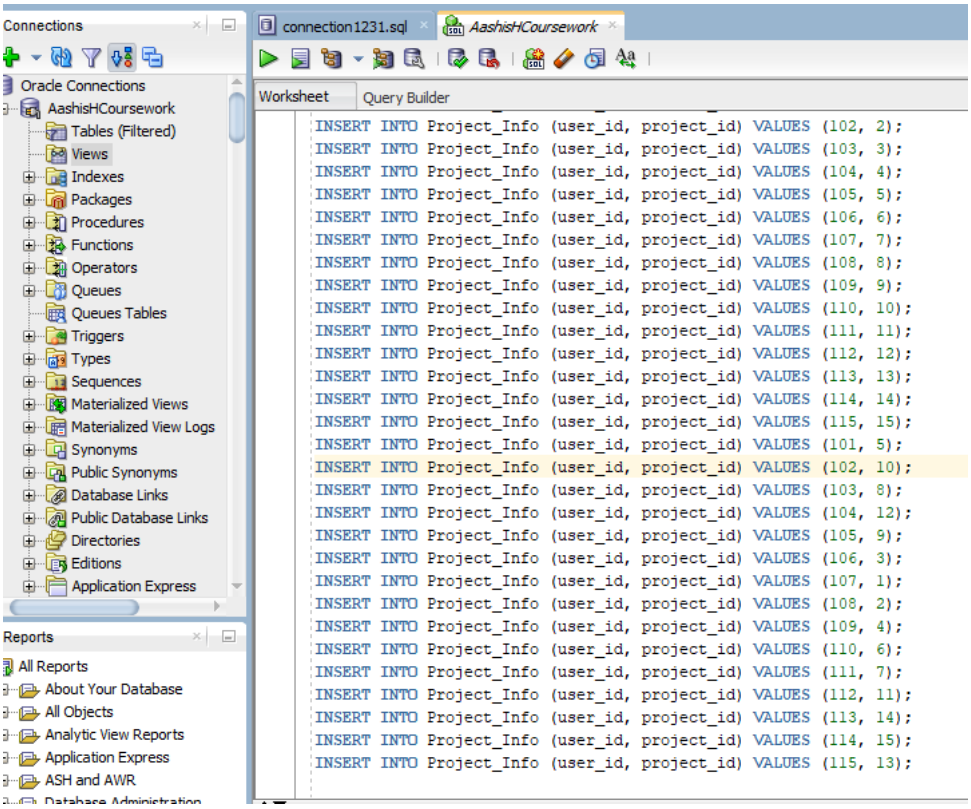


Figure 17: Inserting values in Project_Info table

	USER_ID	PROJECT_ID
1	101	1
2	101	5
3	102	2
4	102	10
5	103	3
6	103	8
7	104	4
8	104	12
9	105	5
10	105	9
11	106	3
12	106	6
13	107	1
14	107	7
15	108	2
16	108	8
17	109	4
18	109	9
19	110	6
20	110	10
21	111	7
22	111	11
23	112	11
24	112	12
25	113	13
26	113	14
27	114	14
28	114	15
29	115	13
30	115	15

Figure 18: Showing Project_Info values

5.3.4 Task_details Table:

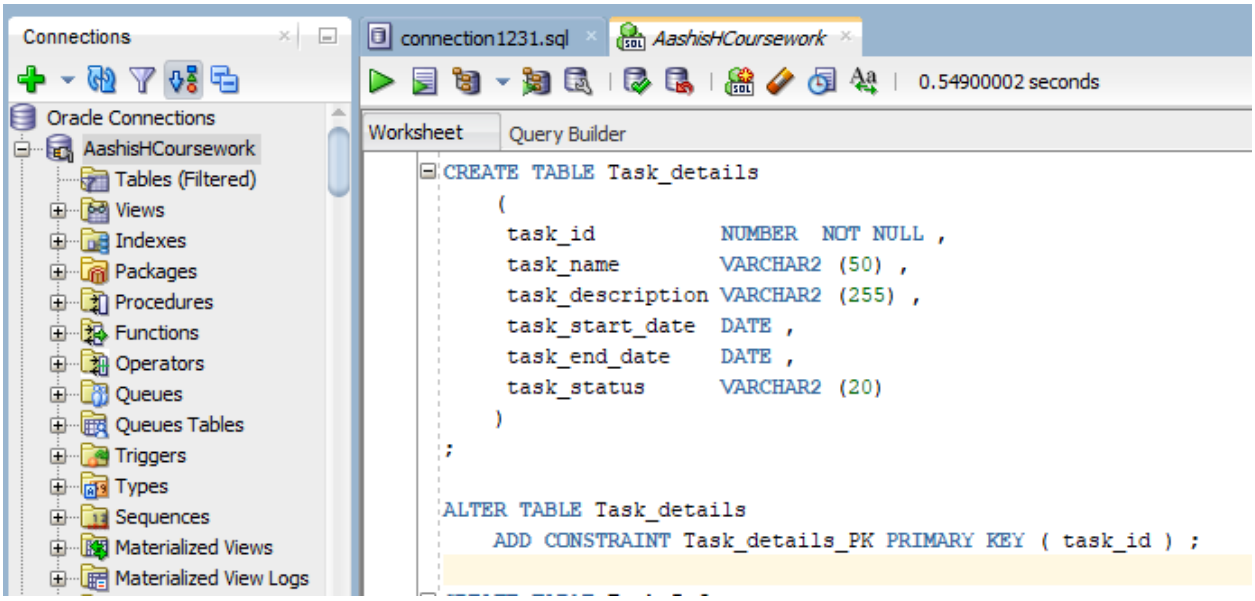
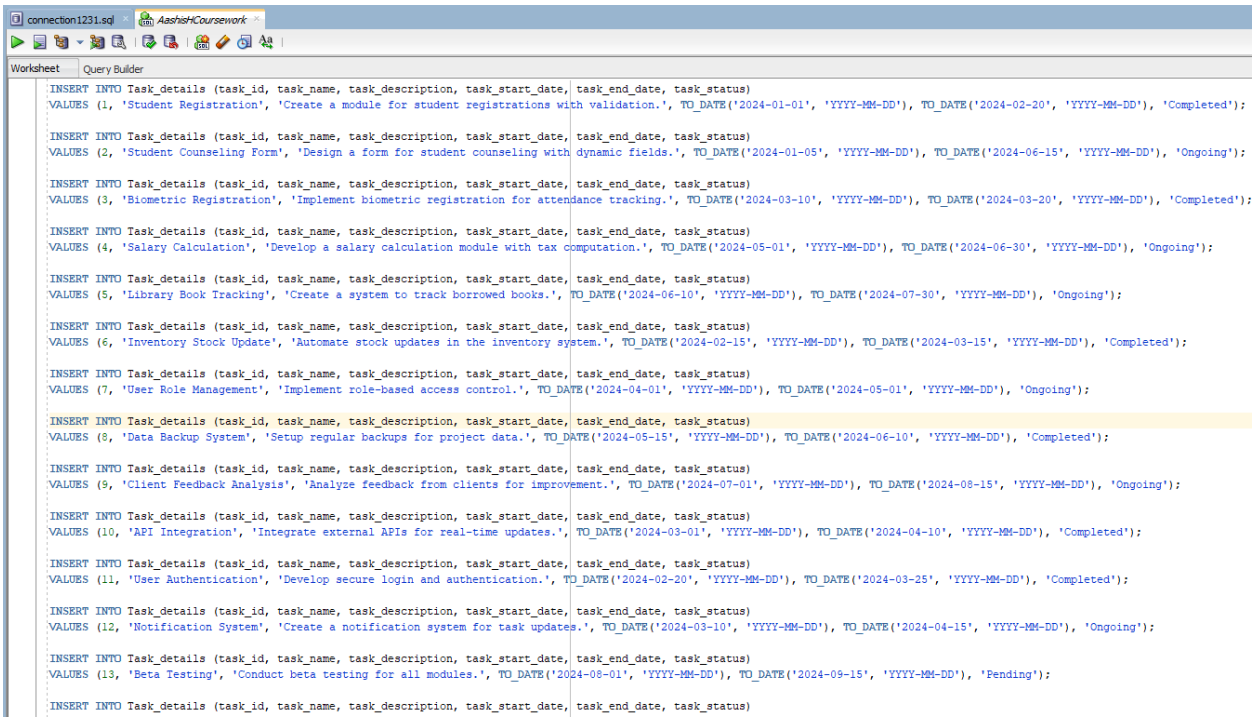


Figure 19: Inserting Values in Task_details table

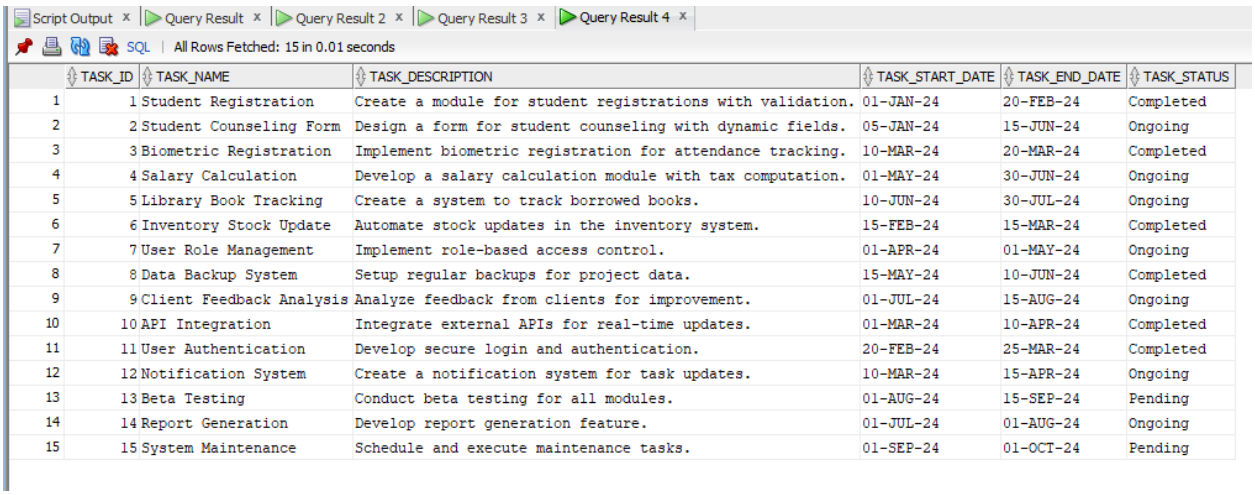
```
Name                 Null?    Type
-------------------- -------- -------------
TASK_ID              NOT NULL NUMBER
TASK_NAME                     VARCHAR2(50)
TASK_DESCRIPTION              VARCHAR2(255)
TASK_START_DATE               DATE
TASK_END_DATE                 DATE
TASK_STATUS                   VARCHAR2(20)
```

Figure 20: DESC Task_detailsTable



```
INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (1, 'Student Registration', 'Create a module for student registrations with validation.', TO_DATE('2024-01-01', 'YYYY-MM-DD'), TO_DATE('2024-02-20', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (2, 'Student Counseling Form', 'Design a form for student counseling with dynamic fields.', TO_DATE('2024-01-05', 'YYYY-MM-DD'), TO_DATE('2024-06-15', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (3, 'Biometric Registration', 'Implement biometric registration for attendance tracking.', TO_DATE('2024-03-10', 'YYYY-MM-DD'), TO_DATE('2024-03-20', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (4, 'Salary Calculation', 'Develop a salary calculation module with tax computation.', TO_DATE('2024-05-01', 'YYYY-MM-DD'), TO_DATE('2024-06-30', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (5, 'Library Book Tracking', 'Create a system to track borrowed books.', TO_DATE('2024-06-10', 'YYYY-MM-DD'), TO_DATE('2024-07-30', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (6, 'Inventory Stock Update', 'Automate stock updates in the inventory system.', TO_DATE('2024-02-15', 'YYYY-MM-DD'), TO_DATE('2024-03-15', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (7, 'User Role Management', 'Implement role-based access control.', TO_DATE('2024-04-01', 'YYYY-MM-DD'), TO_DATE('2024-05-01', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (8, 'Data Backup System', 'Setup regular backups for project data.', TO_DATE('2024-05-15', 'YYYY-MM-DD'), TO_DATE('2024-06-10', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (9, 'Client Feedback Analysis', 'Analyze feedback from clients for improvement.', TO_DATE('2024-07-01', 'YYYY-MM-DD'), TO_DATE('2024-08-15', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (10, 'API Integration', 'Integrate external APIs for real-time updates.', TO_DATE('2024-03-01', 'YYYY-MM-DD'), TO_DATE('2024-04-10', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (11, 'User Authentication', 'Develop secure login and authentication.', TO_DATE('2024-02-20', 'YYYY-MM-DD'), TO_DATE('2024-03-25', 'YYYY-MM-DD'), 'Completed');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (12, 'Notification System', 'Create a notification system for task updates.', TO_DATE('2024-03-10', 'YYYY-MM-DD'), TO_DATE('2024-04-15', 'YYYY-MM-DD'), 'Ongoing');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
VALUES (13, 'Beta Testing', 'Conduct beta testing for all modules.', TO_DATE('2024-08-01', 'YYYY-MM-DD'), TO_DATE('2024-09-15', 'YYYY-MM-DD'), 'Pending');

INSERT INTO Task_details (task_id, task_name, task_description, task_start_date, task_end_date, task_status)
```

Figure 21: Inserting values in Task_details table



All Rows Fetched: 15 in 0.01 seconds

	TASK_ID	TASK_NAME	TASK_DESCRIPTION	TASK_START_DATE	TASK_END_DATE	TASK_STATUS
1	1	Student Registration	Create a module for student registrations with validation.	01-JAN-24	20-FEB-24	Completed
2	2	Student Counseling Form	Design a form for student counseling with dynamic fields.	05-JAN-24	15-JUN-24	Ongoing
3	3	Biometric Registration	Implement biometric registration for attendance tracking.	10-MAR-24	20-MAR-24	Completed
4	4	Salary Calculation	Develop a salary calculation module with tax computation.	01-MAY-24	30-JUN-24	Ongoing
5	5	Library Book Tracking	Create a system to track borrowed books.	10-JUN-24	30-JUL-24	Ongoing
6	6	Inventory Stock Update	Automate stock updates in the inventory system.	15-FEB-24	15-MAR-24	Completed
7	7	User Role Management	Implement role-based access control.	01-APR-24	01-MAY-24	Ongoing
8	8	Data Backup System	Setup regular backups for project data.	15-MAY-24	10-JUN-24	Completed
9	9	Client Feedback Analysis	Analyze feedback from clients for improvement.	01-JUL-24	15-AUG-24	Ongoing
10	10	API Integration	Integrate external APIs for real-time updates.	01-MAR-24	10-APR-24	Completed
11	11	User Authentication	Develop secure login and authentication.	20-FEB-24	25-MAR-24	Completed
12	12	Notification System	Create a notification system for task updates.	10-MAR-24	15-APR-24	Ongoing
13	13	Beta Testing	Conduct beta testing for all modules.	01-AUG-24	15-SEP-24	Pending
14	14	Report Generation	Develop report generation feature.	01-JUL-24	01-AUG-24	Ongoing
15	15	System Maintenance	Schedule and execute maintenance tasks.	01-SEP-24	01-OCT-24	Pending

Figure 22: Showing values in Task_details values

5.3.5 Task_Info Table

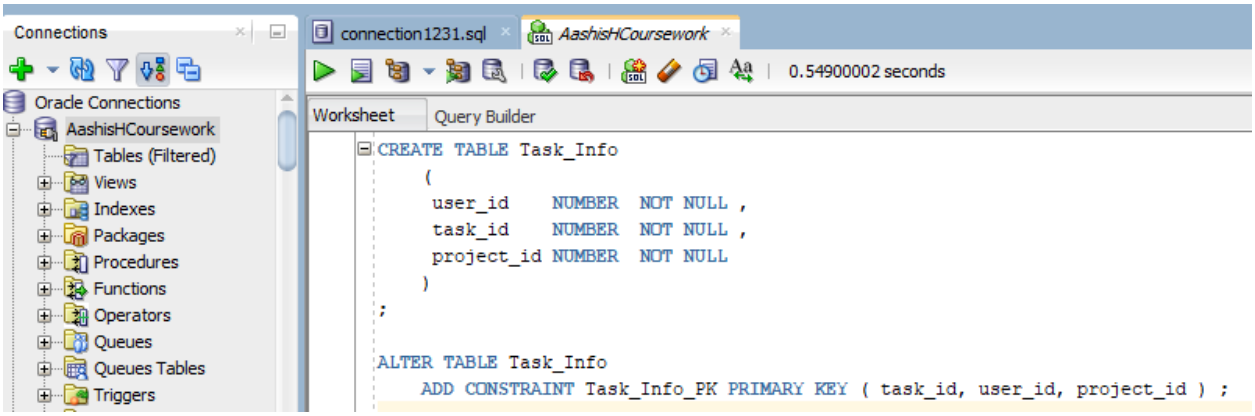


Figure 23: creating Task_Info table

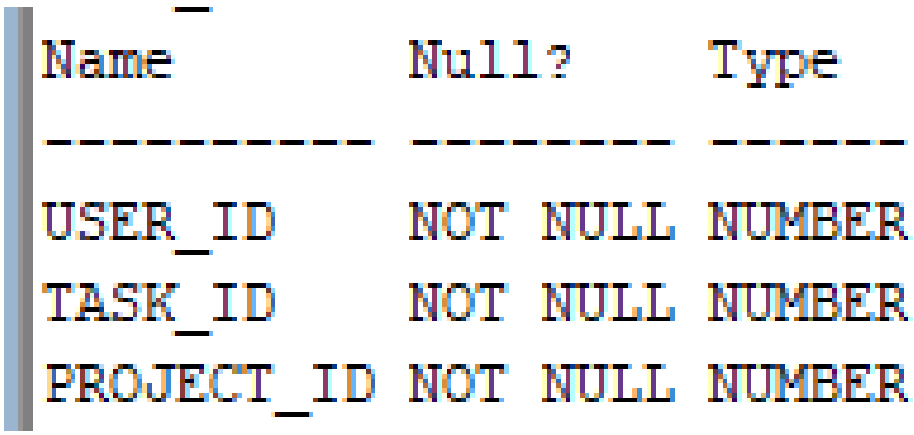


Figure 24: DESC Task_Info table

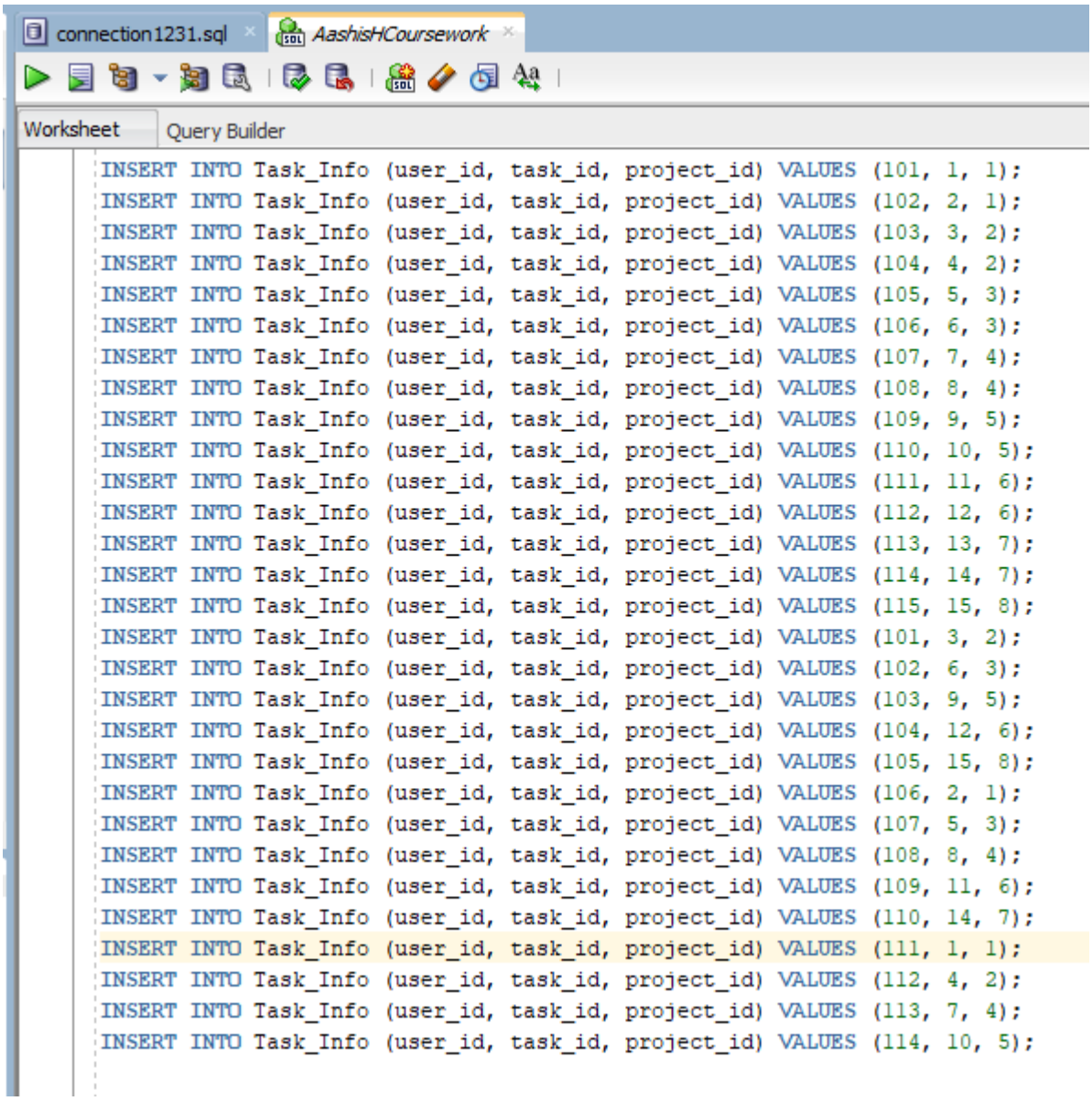


Figure 25: Inserting values in Task_Info Table;

Script Output | Query Result | Query Result 2

SQL | All Rows Fetched: 28 in 0.005 seconds

	USER_ID	TASK_ID	PROJECT_ID
1	101	1	1
2	111	1	1
3	102	2	1
4	106	2	1
5	101	3	2
6	103	3	2
7	104	4	2
8	112	4	2
9	105	5	3
10	107	5	3
11	102	6	3
12	106	6	3
13	107	7	4
14	113	7	4
15	108	8	4
16	103	9	5
17	109	9	5
18	110	10	5
19	114	10	5
20	109	11	6
21	111	11	6
22	104	12	6
23	112	12	6
24	113	13	7
25	110	14	7
26	114	14	7
27	105	15	8
28	115	15	8

Figure 26: Showing Task_Info values

5.3.6 Sub_task_details Table

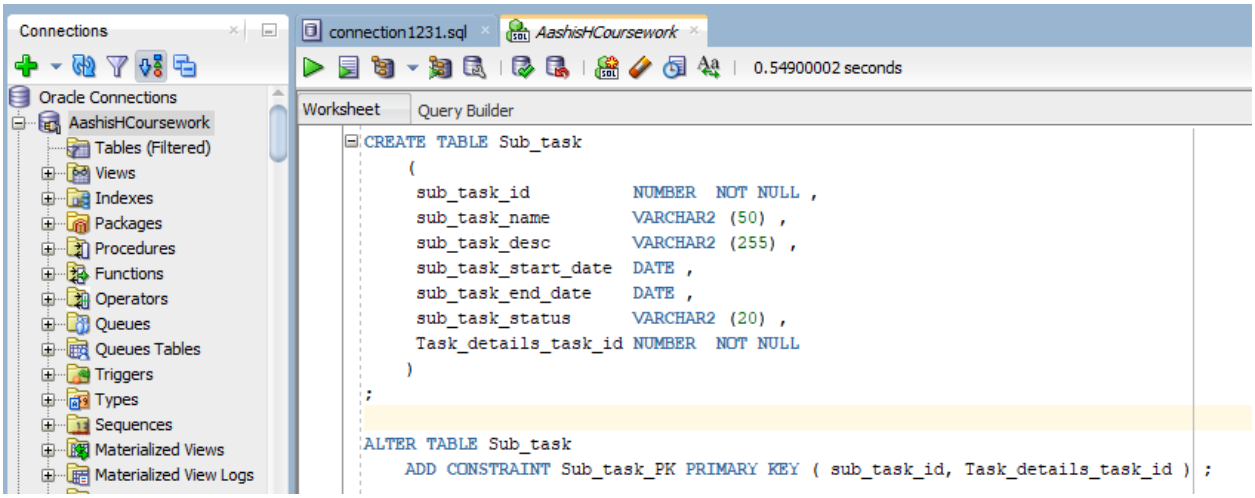
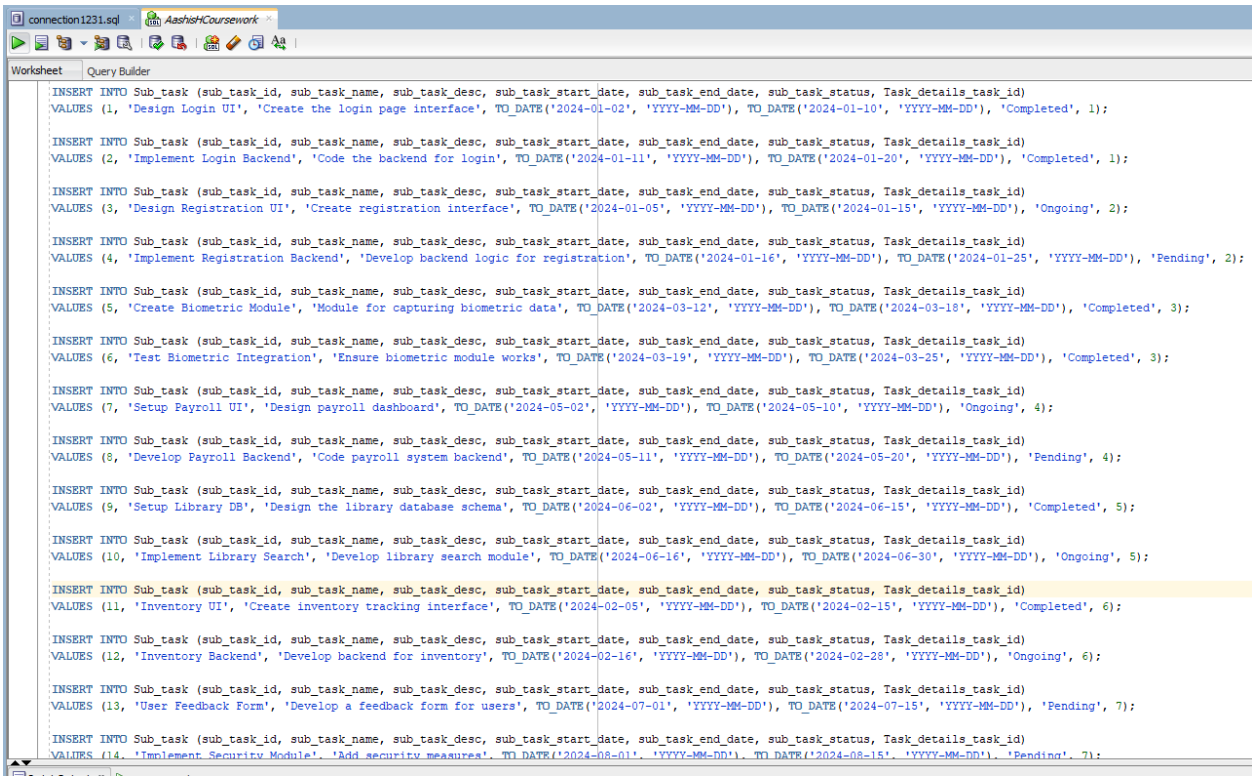


Figure 27: Creating Sub_task_details table

```
Name                 Null?    Type
-------------------- -------- -------------
SUB_TASK_ID          NOT NULL NUMBER
SUB_TASK_NAME                 VARCHAR2(50)
SUB_TASK_DESC                 VARCHAR2(255)
SUB_TASK_START_DATE           DATE
SUB_TASK_END_DATE             DATE
SUB_TASK_STATUS               VARCHAR2(20)
TASK_DETAILS_TASK_ID NOT NULL NUMBER
```

Figure 28: DESC Sub_task_details



```
INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (1, 'Design Login UI', 'Create the login page interface', TO_DATE('2024-01-02', 'YYYY-MM-DD'), TO_DATE('2024-01-10', 'YYYY-MM-DD'), 'Completed', 1);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (2, 'Implement Login Backend', 'Code the backend for login', TO_DATE('2024-01-11', 'YYYY-MM-DD'), TO_DATE('2024-01-20', 'YYYY-MM-DD'), 'Completed', 1);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (3, 'Design Registration UI', 'Create registration interface', TO_DATE('2024-01-05', 'YYYY-MM-DD'), TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Ongoing', 2);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (4, 'Implement Registration Backend', 'Develop backend logic for registration', TO_DATE('2024-01-16', 'YYYY-MM-DD'), TO_DATE('2024-01-25', 'YYYY-MM-DD'), 'Pending', 2);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (5, 'Create Biometric Module', 'Module for capturing biometric data', TO_DATE('2024-03-12', 'YYYY-MM-DD'), TO_DATE('2024-03-18', 'YYYY-MM-DD'), 'Completed', 3);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (6, 'Test Biometric Integration', 'Ensure biometric module works', TO_DATE('2024-03-19', 'YYYY-MM-DD'), TO_DATE('2024-03-25', 'YYYY-MM-DD'), 'Completed', 3);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (7, 'Setup Payroll UI', 'Design payroll dashboard', TO_DATE('2024-05-02', 'YYYY-MM-DD'), TO_DATE('2024-05-10', 'YYYY-MM-DD'), 'Ongoing', 4);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (8, 'Develop Payroll Backend', 'Code payroll system backend', TO_DATE('2024-05-11', 'YYYY-MM-DD'), TO_DATE('2024-05-20', 'YYYY-MM-DD'), 'Pending', 4);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (9, 'Setup Library DB', 'Design the library database schema', TO_DATE('2024-06-02', 'YYYY-MM-DD'), TO_DATE('2024-06-15', 'YYYY-MM-DD'), 'Completed', 5);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (10, 'Implement Library Search', 'Develop library search module', TO_DATE('2024-06-16', 'YYYY-MM-DD'), TO_DATE('2024-06-30', 'YYYY-MM-DD'), 'Ongoing', 5);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (11, 'Inventory UI', 'Create inventory tracking interface', TO_DATE('2024-02-05', 'YYYY-MM-DD'), TO_DATE('2024-02-15', 'YYYY-MM-DD'), 'Completed', 6);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (12, 'Inventory Backend', 'Develop backend for inventory', TO_DATE('2024-02-16', 'YYYY-MM-DD'), TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Ongoing', 6);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (13, 'User Feedback Form', 'Develop a feedback form for users', TO_DATE('2024-07-01', 'YYYY-MM-DD'), TO_DATE('2024-07-15', 'YYYY-MM-DD'), 'Pending', 7);

INSERT INTO Sub_task (sub_task_id, sub_task_name, sub_task_desc, sub_task_start_date, sub_task_end_date, sub_task_status, Task_details_task_id)
VALUES (14, 'Implement Security Module', 'Add security measures', TO_DATE('2024-08-01', 'YYYY-MM-DD'), TO_DATE('2024-08-15', 'YYYY-MM-DD'), 'Pending', 7);
```

Figure 29: Inserting values in Sub_task_Info table

	SUB_TASK_ID	SUB_TASK_NAME	SUB_TASK_DESC	SUB_TASK_START_DATE	SUB_TASK_END_DATE	SUB_TASK_STATUS	TASK_DETAILS_TASK_ID
1	1	Design Login UI	Create the login page interface	02-JAN-24	10-JAN-24	Completed	1
2	2	Implement Login Backend	Code the backend for login	11-JAN-24	20-JAN-24	Completed	1
3	3	Design Registration UI	Create registration interface	05-JAN-24	15-JAN-24	Ongoing	2
4	4	Implement Registration Backend	Develop backend logic for registration	16-JAN-24	25-JAN-24	Pending	2
5	5	Create Biometric Module	Module for capturing biometric data	12-MAR-24	18-MAR-24	Completed	3
6	6	Test Biometric Integration	Ensure biometric module works	19-MAR-24	25-MAR-24	Completed	3
7	7	Setup Payroll UI	Design payroll dashboard	02-MAY-24	10-MAY-24	Ongoing	4
8	8	Develop Payroll Backend	Code payroll system backend	11-MAY-24	20-MAY-24	Pending	4
9	9	Setup Library DB	Design the library database schema	02-JUN-24	15-JUN-24	Completed	5
10	10	Implement Library Search	Develop library search module	16-JUN-24	30-JUN-24	Ongoing	5
11	11	Inventory UI	Create inventory tracking interface	05-FEB-24	15-FEB-24	Completed	6
12	12	Inventory Backend	Develop backend for inventory	16-FEB-24	28-FEB-24	Ongoing	6
13	13	User Feedback Form	Develop a feedback form for users	01-JUL-24	15-JUL-24	Pending	7
14	14	Implement Security Module	Add security measures	01-AUG-24	15-AUG-24	Pending	7
15	15	Create Analytics Dashboard	Dashboard for project insights	01-SEP-24	20-SEP-24	Ongoing	8

Figure 30: Showing Sub_task_Info Values

5.3.7 Milestone_details Table

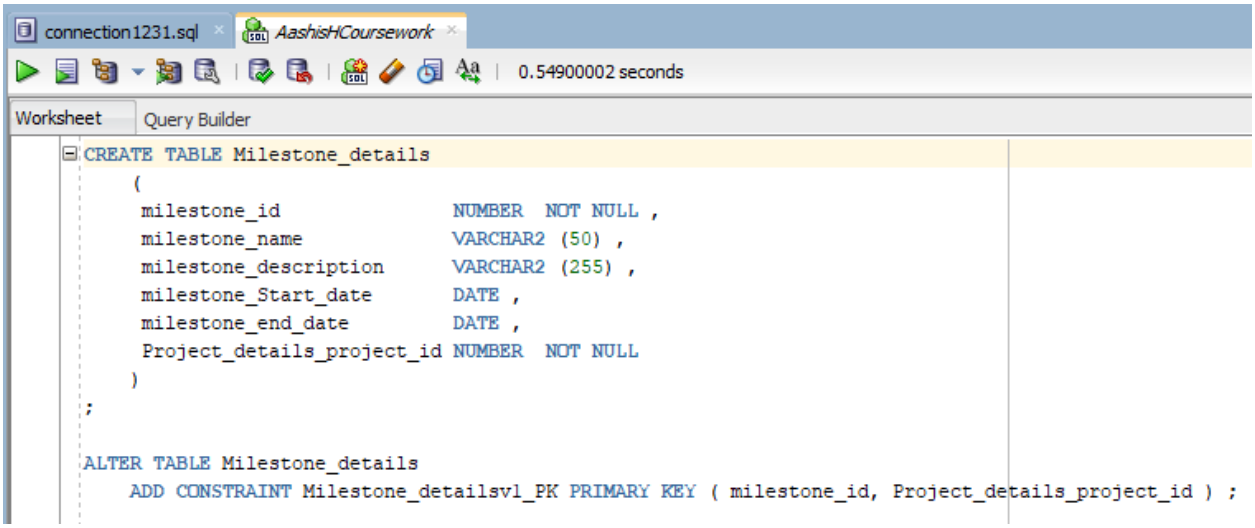


Figure 31: Creating Milestone_details table

```
Name                           Null?    Type
------------------------------ -------- -------------
MILESTONE_ID                   NOT NULL NUMBER
MILESTONE_NAME                          VARCHAR2(50)
MILESTONE_DESCRIPTION                   VARCHAR2(255)
MILESTONE_START_DATE                    DATE
MILESTONE_END_DATE                      DATE
PROJECT_DETAILS_PROJECT_ID NOT NULL     NUMBER
```

Figure 32: DESC Milestone_details table

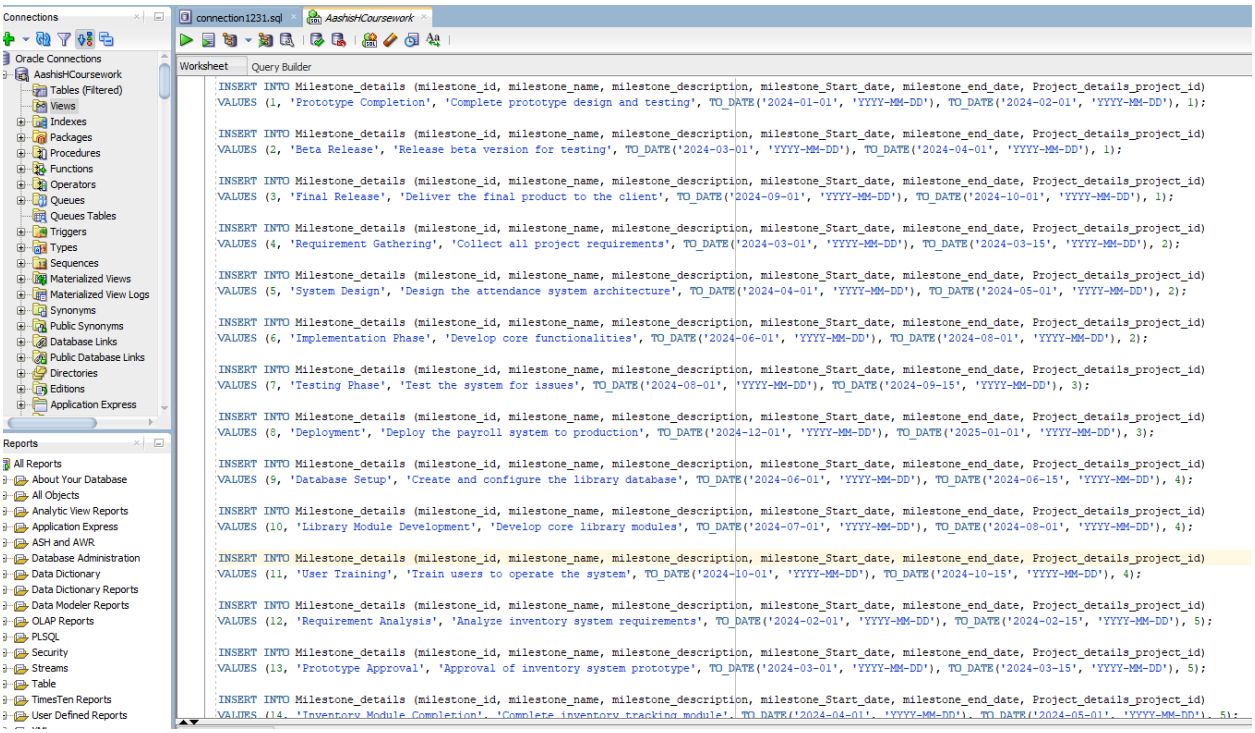


Figure 33: Inserting values in Milestone_details table

	MILESTONE_ID	MILESTONE_NAME	MILESTONE_DESCRIPTION	MILESTONE_START_DATE	MILESTONE_END_DATE	PROJECT_DETAILS_PROJECT_ID
1	1	Prototype Completion	Complete prototype design and testing	01-JAN-24	01-FEB-24	1
2	2	Beta Release	Release beta version for testing	01-MAR-24	01-APR-24	1
3	3	Final Release	Deliver the final product to the client	01-SEP-24	01-OCT-24	1
4	4	Requirement Gathering	Collect all project requirements	01-MAR-24	15-MAR-24	2
5	5	System Design	Design the attendance system architecture	01-APR-24	01-MAY-24	2
6	6	Implementation Phase	Develop core functionalities	01-JUN-24	01-AUG-24	2
7	7	Testing Phase	Test the system for issues	01-AUG-24	15-SEP-24	3
8	8	Deployment	Deploy the payroll system to production	01-DEC-24	01-JAN-25	3
9	9	Database Setup	Create and configure the library database	01-JUN-24	15-JUN-24	4
10	10	Library Module Development	Develop core library modules	01-JUL-24	01-AUG-24	4
11	11	User Training	Train users to operate the system	01-OCT-24	15-OCT-24	4
12	12	Requirement Analysis	Analyze inventory system requirements	01-FEB-24	15-FEB-24	5
13	13	Prototype Approval	Approval of inventory system prototype	01-MAR-24	15-MAR-24	5
14	14	Inventory Module Completion	Complete inventory tracking module	01-APR-24	01-MAY-24	5
15	15	Final Review and Handover	Review and handover inventory system	01-JUL-24	01-AUG-24	5

Figure 34: Showing Milestone_details values

6 Basic Web Forms

6.1 Task Tracking Dashboard

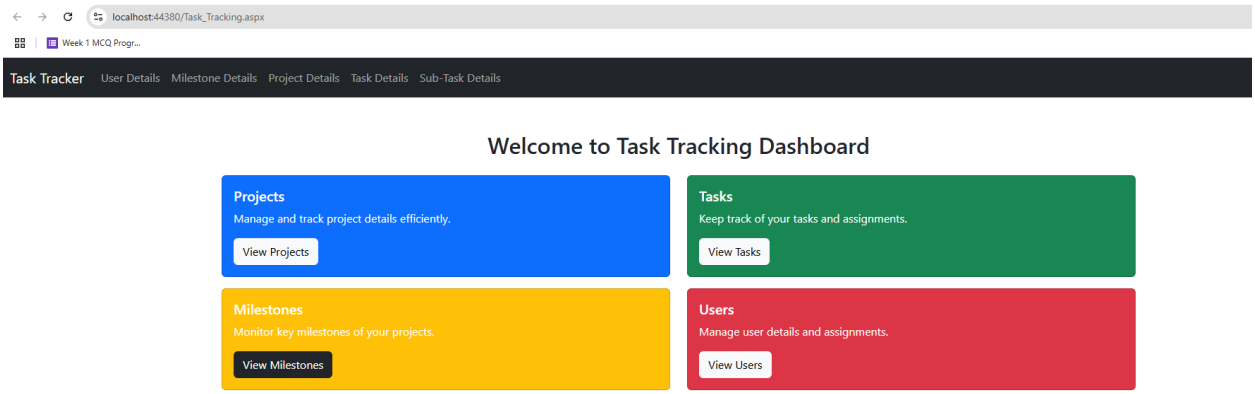


Figure 35: Main Dashboard

6.2 User_details Web Form

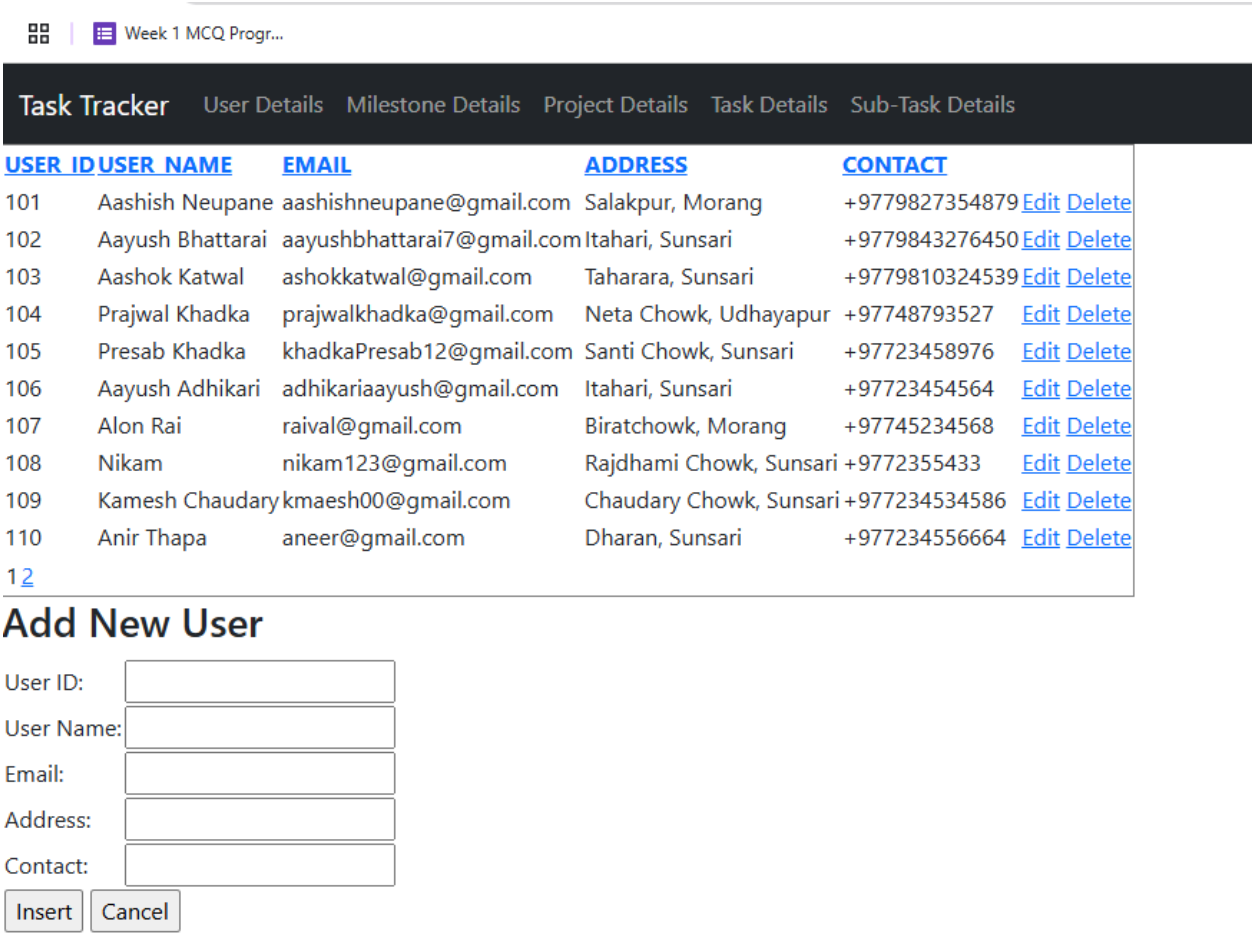


Figure 36: User_details Web Form

6.3 Project_details Web Form

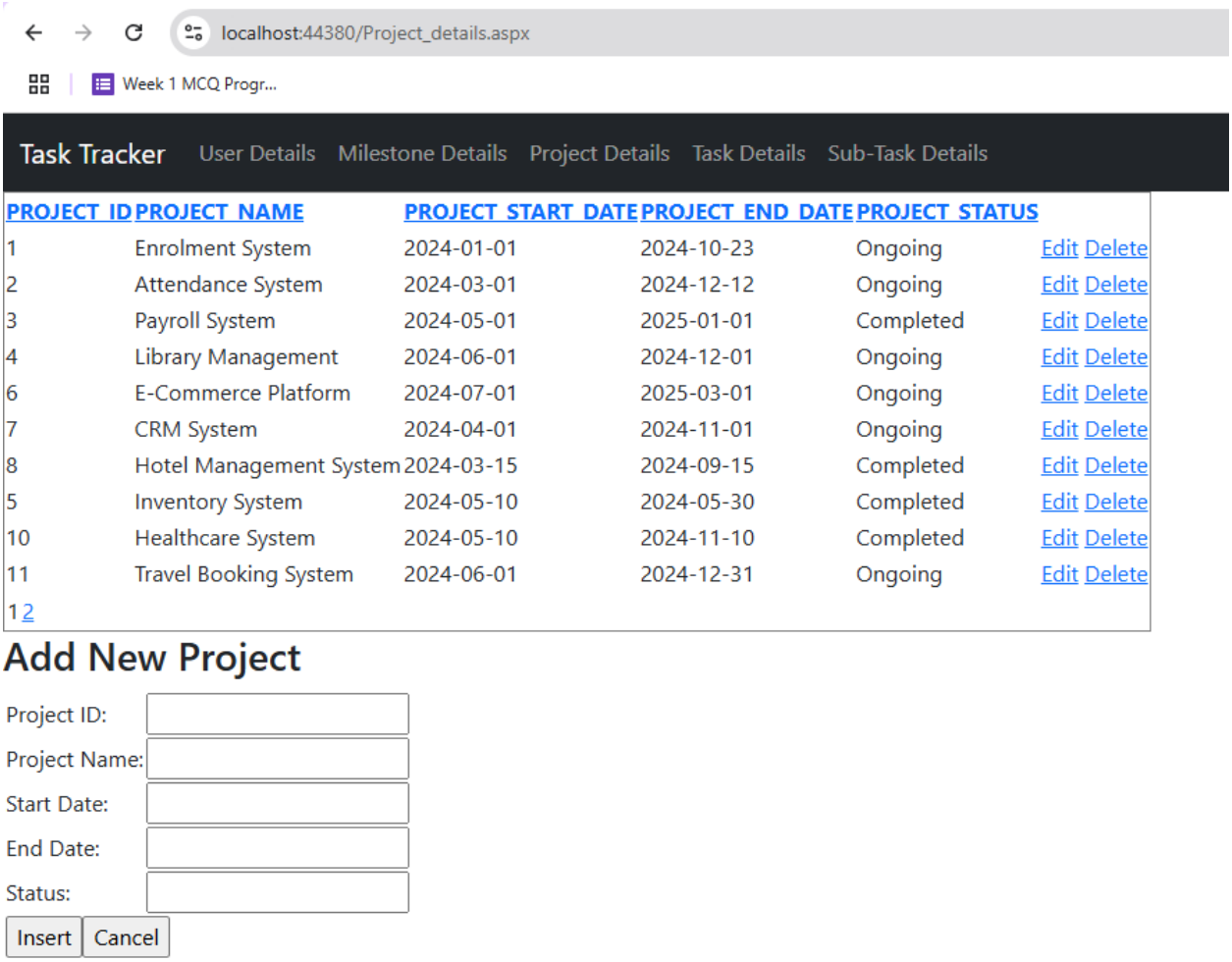


Figure 37: Project_details Web Form

6.4 Task_details Web Form

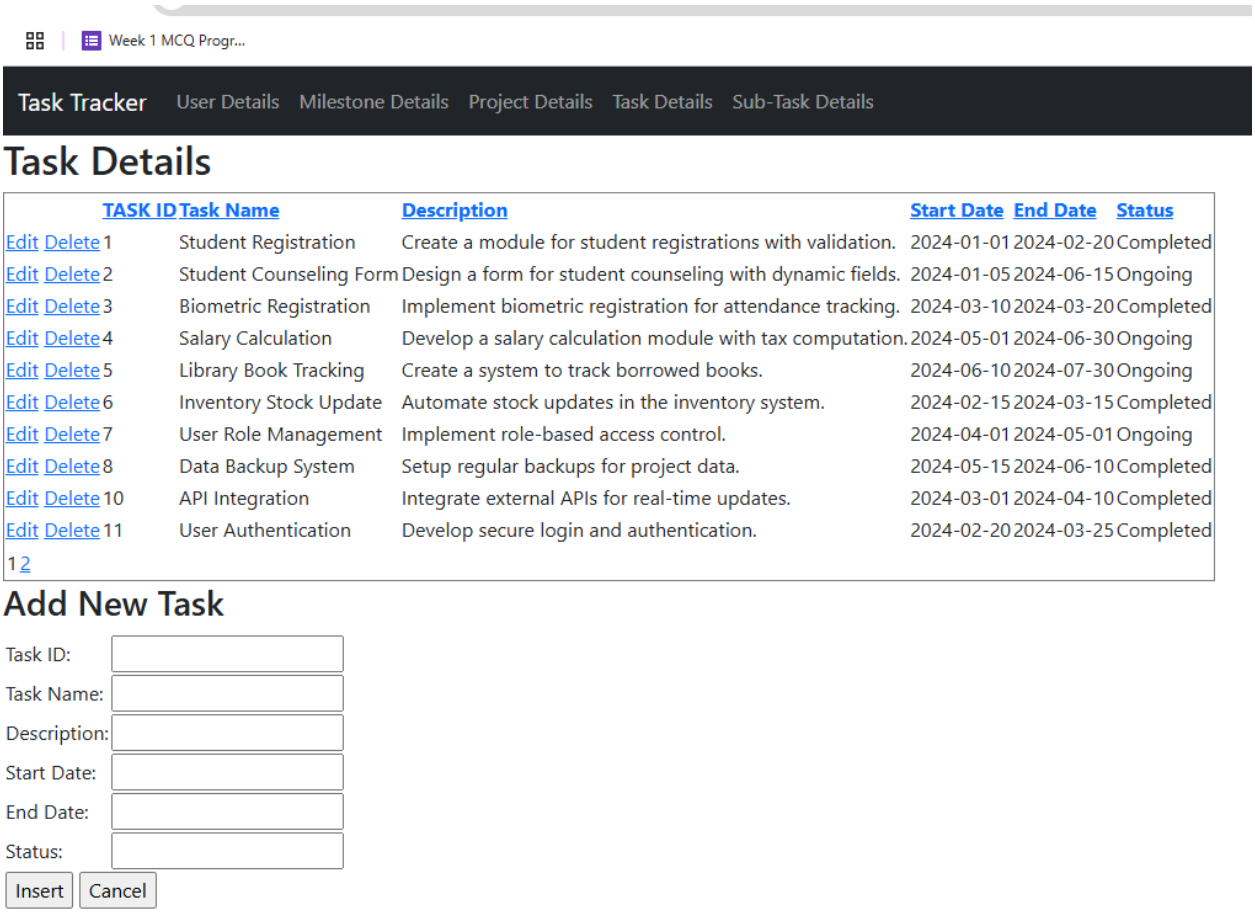


Figure 38: Task_details Web Form

6.5 Sub_task_details Web Form

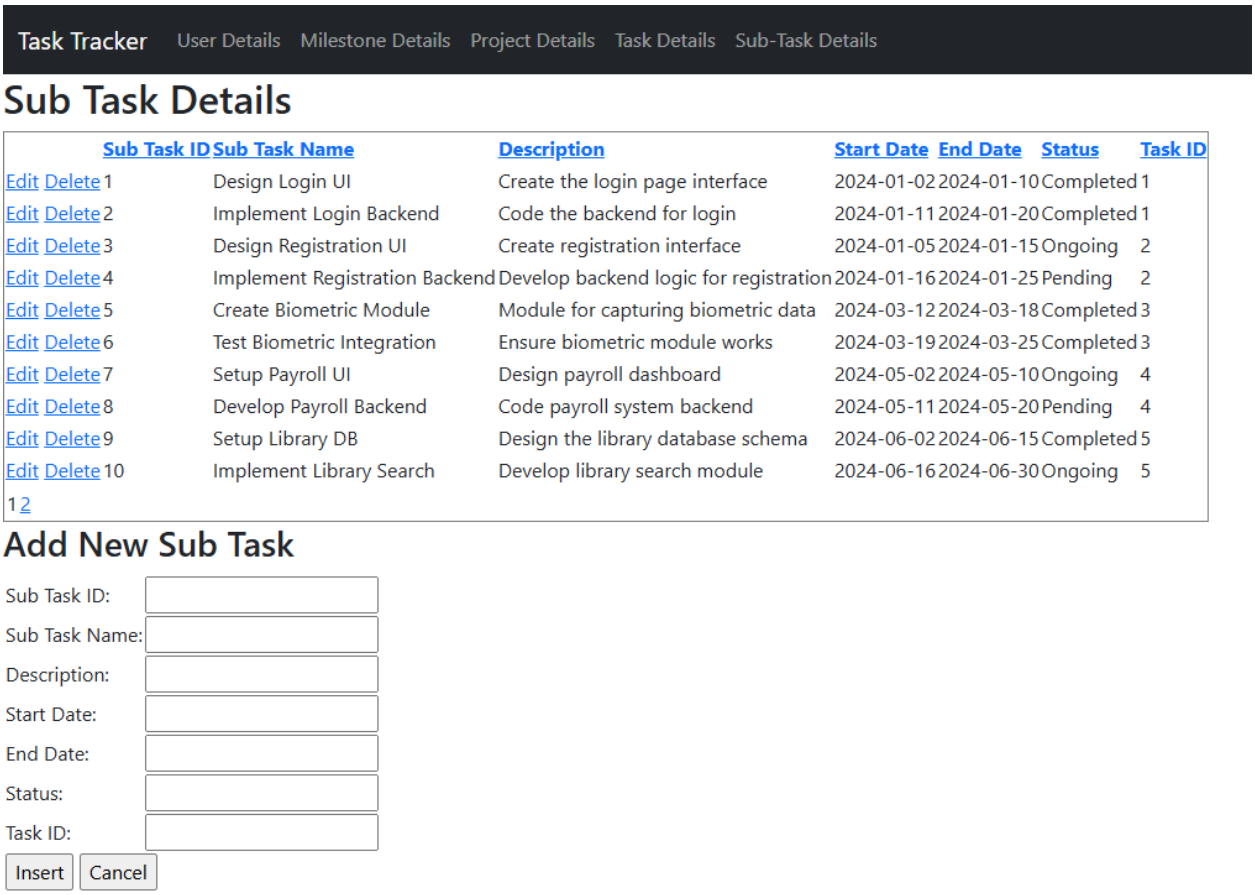


Figure 39: Sub_task_details Web Form

6.6 Milestone_details Web Form

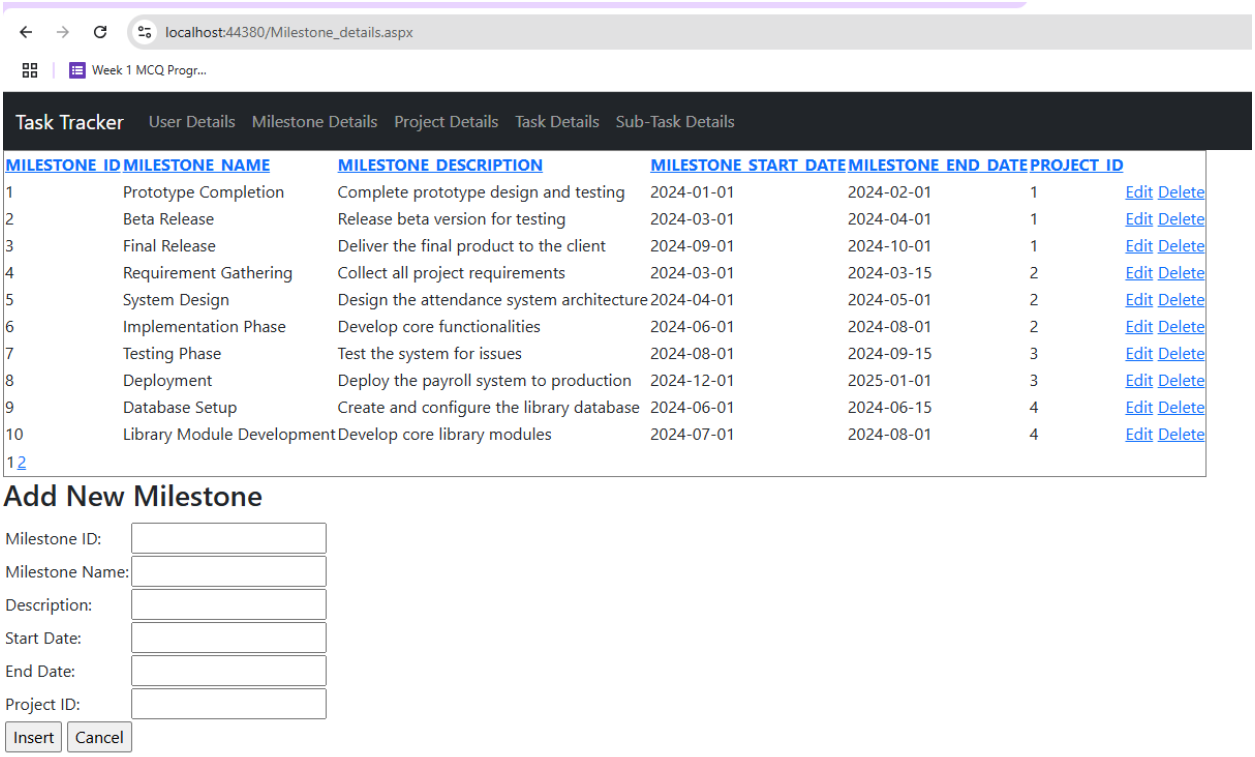


Figure 40: Milestone_Details Web Form

7 Complex Web Form

7.1 Updated Dashboard

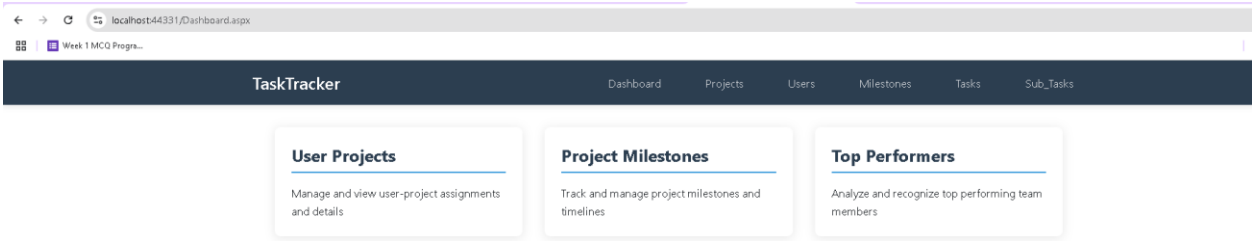


Figure 41: Updated Dashboard

7.2 User Project (for any user, show details of the user and the details of all the projects that he/she has been working or has worked on.

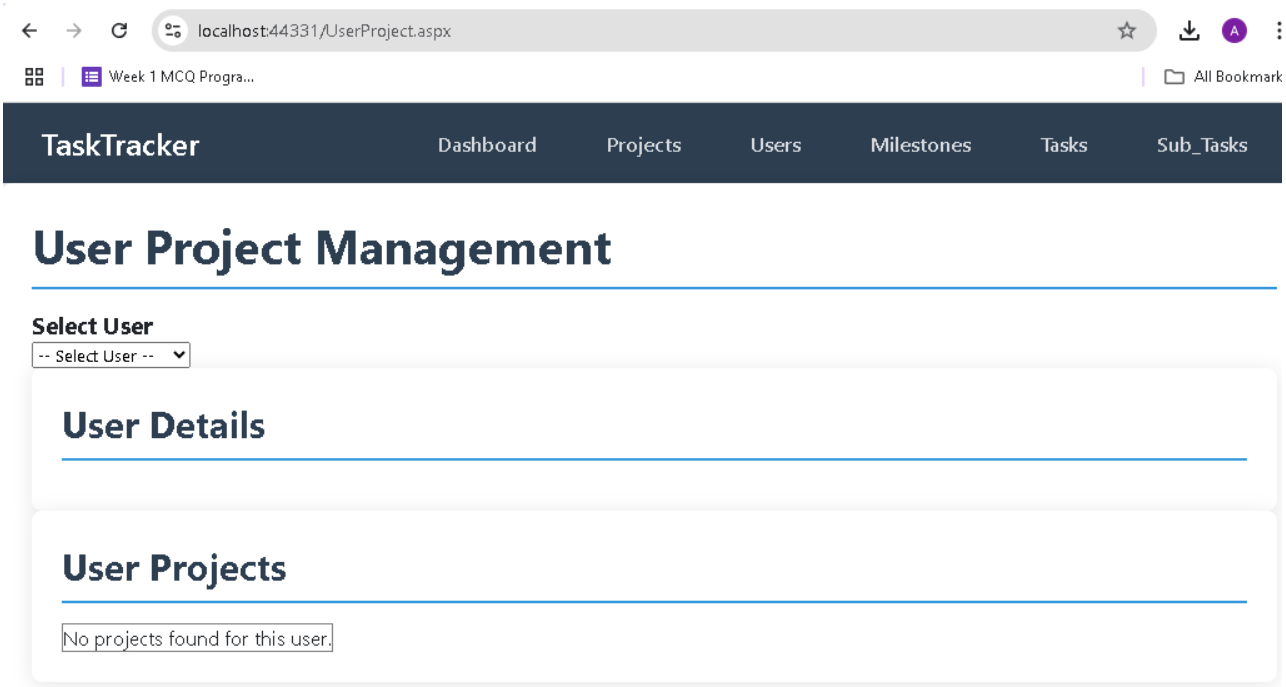


Figure 42: Before showing details of user and their associated Project

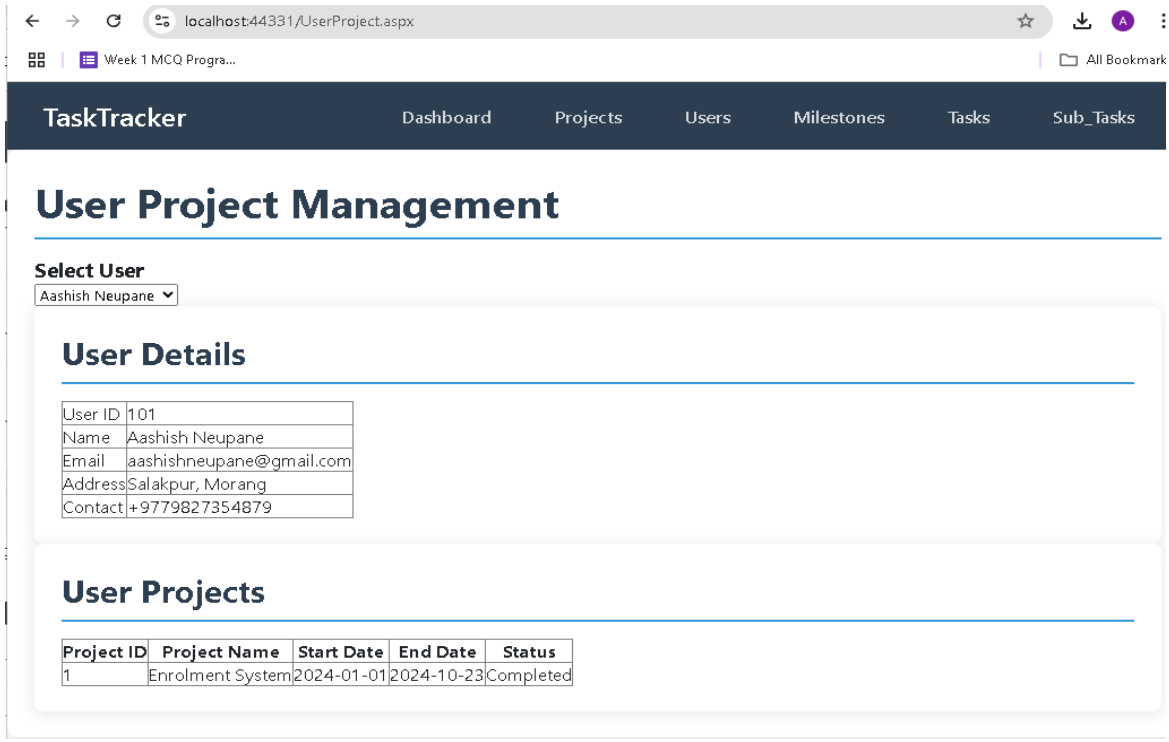


Figure 43: After showing the details of User and their project

7.3 Project Milestone (For any Project, show the detail of the project with all the milestone of the project)

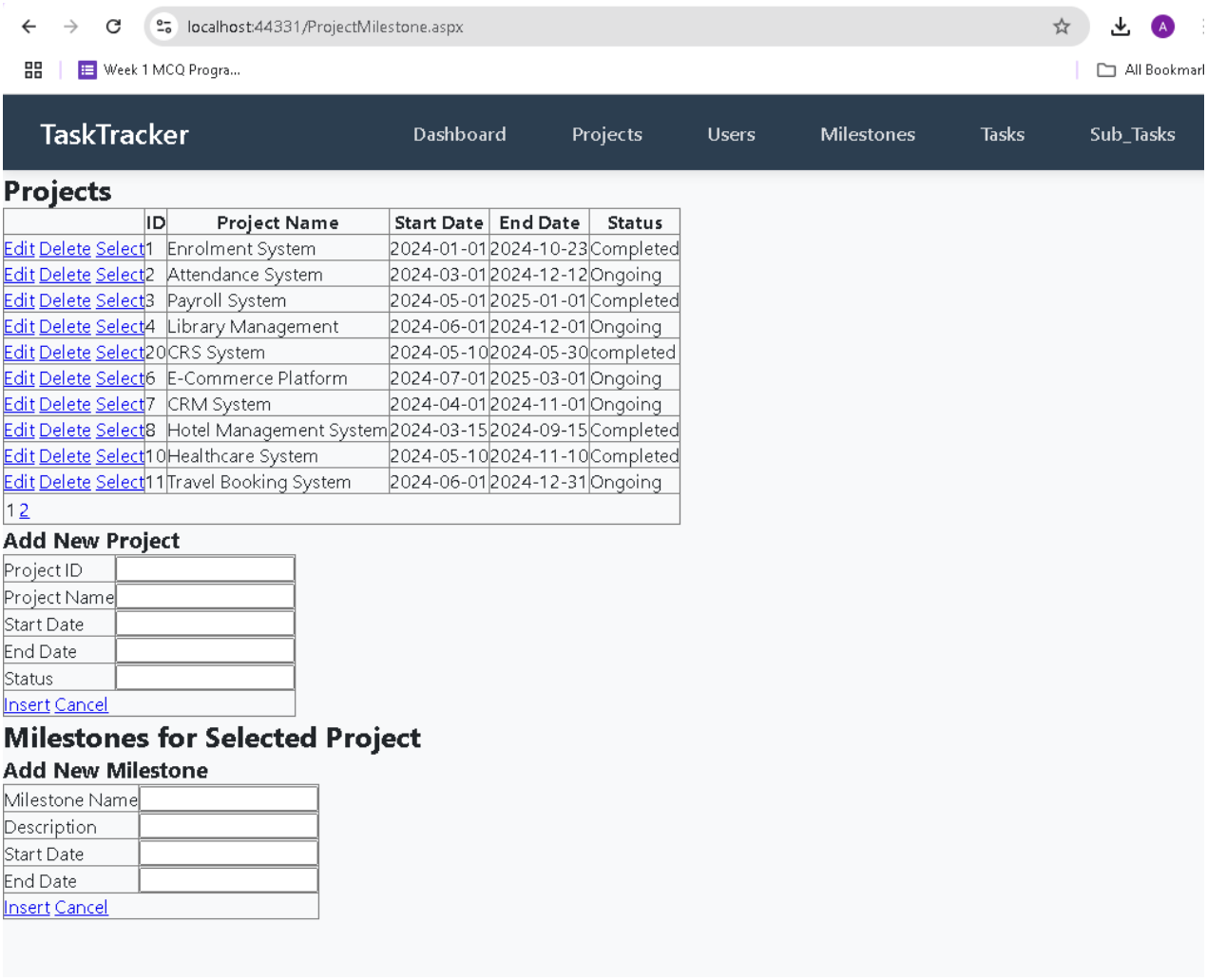


Figure 44: Before showing the projects details and all their milestones

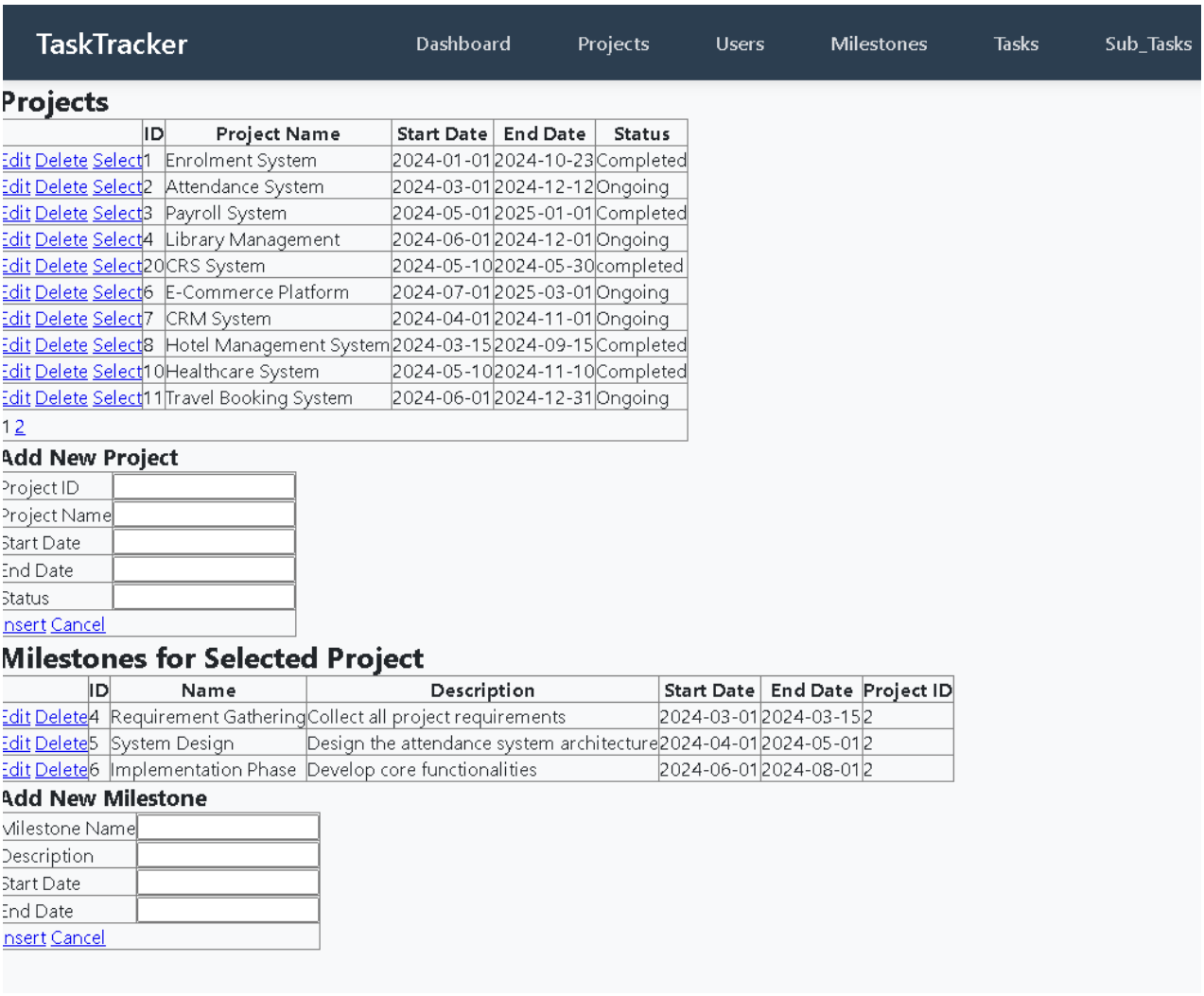


Figure 45: After choosing the Project and output as milestone

7.4 Top Performer (for any project, show the details of top 3 user who has done the most task).

NOTE: Any pending task won't be counted as task done.

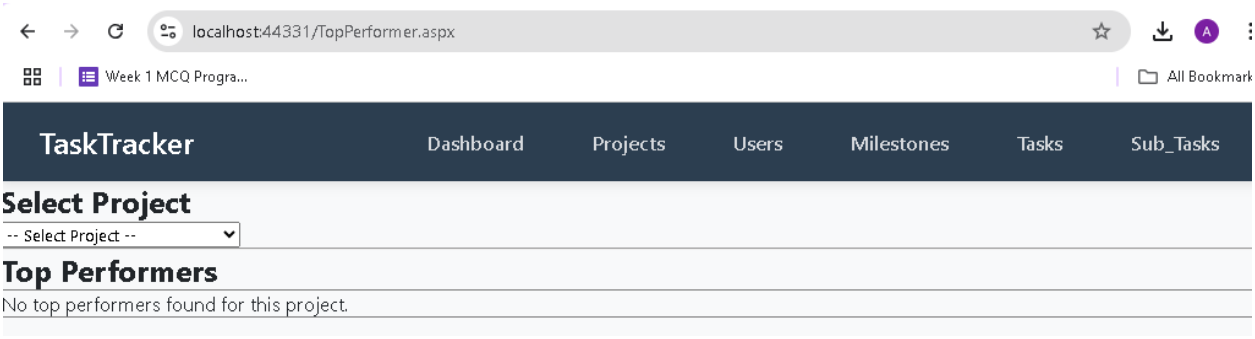


Figure 46: Before Selecting the Project with top performer

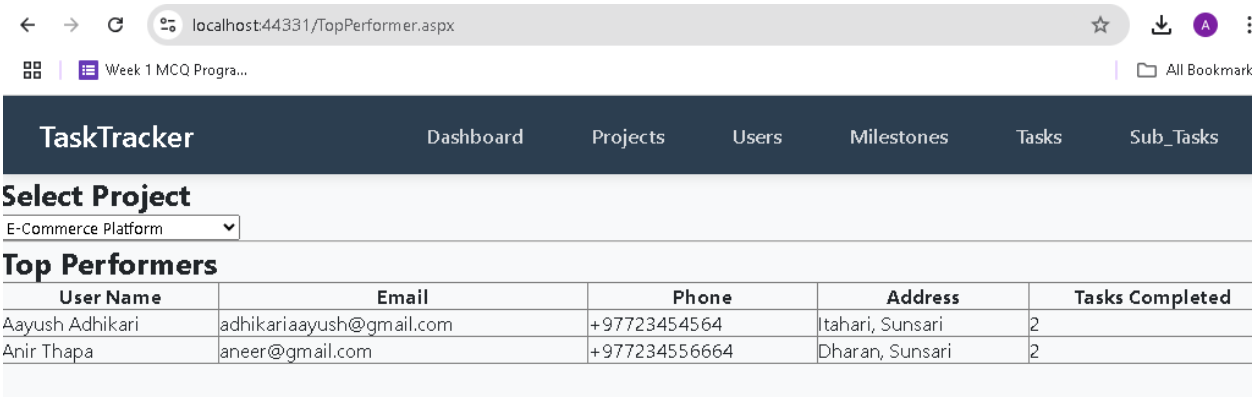


Figure 47: After selecting the Project with top performer

8 Conclusion

The Basic and Complex Web form required for the project is successfully finished. Also, the Dashboard required for the Task Tracking System is also done as shown in figure above in the dashboard section. So, the remaining UI components designs will be finished and will be finalized and included the final submission

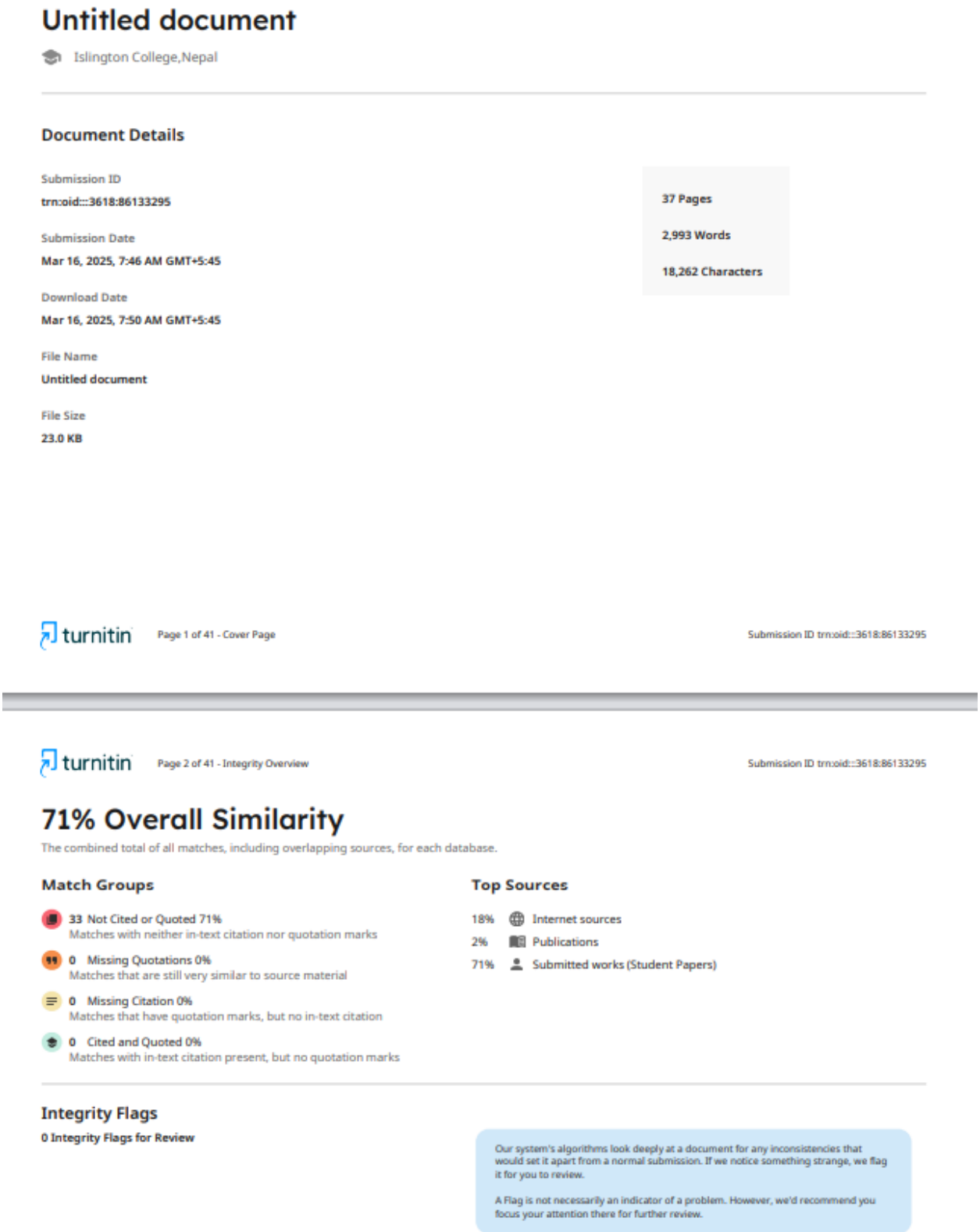
9 References

Margaret Rouse, J. V. (2024, december 22). *techtarget*. Retrieved from techtarget.com: https://www.techtarget.com/searchdatamanagement/definition/normalization

Mondal, T. (2014). Entity search : A New Search Paradigm. *Tanmay Mondal,MSLIS Student,DRTC*, 4.

Soni, D. (2024, december 21). *scaler*. Retrieved from scaler.com: https://www.scaler.com/topics/types-of-relationship-in-dbms/

Appendix



Untitled document

Islington College,Nepal

Document Details

Submission ID
trn:oid:::3618:86133295

Submission Date
Mar 16, 2025, 7:46 AM GMT+5:45

Download Date
Mar 16, 2025, 7:50 AM GMT+5:45

File Name
Untitled document

File Size
23.0 KB

37 Pages

2,993 Words

18,262 Characters

turnitin Page 1 of 41 - Cover Page Submission ID trn:oid:::3618:86133295

turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::3618:86133295

71% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

33 Not Cited or Quoted 71%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

18% Internet sources
2% Publications
71% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Figure 48: Similarity Report