KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638 060

DEPARTMENT OF COMPUTER APPLICATIONS

AN APPLICATION PROJECT REPORT

FOR

PROJECT MANAGEMENT SYSTEM

ADVANCED DATABASE TECHNOLOGIES

LABORATORY - 22MCT11

Submitted by

V.RAMYA(23MCR075)

M.SOUNDARYA(23MCR096)

L.SRIVARSHINI(23MCR108)

KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638 060

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Name	: **V.RAMYA (23MCR075)** **M.SOUNDARYA (23MCR096)** **L.SRIVARSHINI (23MCR108)**
Course Code	: **22MCT11**
Course Name	: **ADVANCED DATABASE TECHNOLOGIES LABORATORY**
Semester	: **I**

Certified that this is a bonafide record of work for application project done by the above students for **22MCT11 – ADVANCED DATABASE TECHNOLOGIES** during the academic year **2023-2024.**

Submitted for the Viva Voce Examination held on ______________________

Lab-in-Charge **Head of the Department**

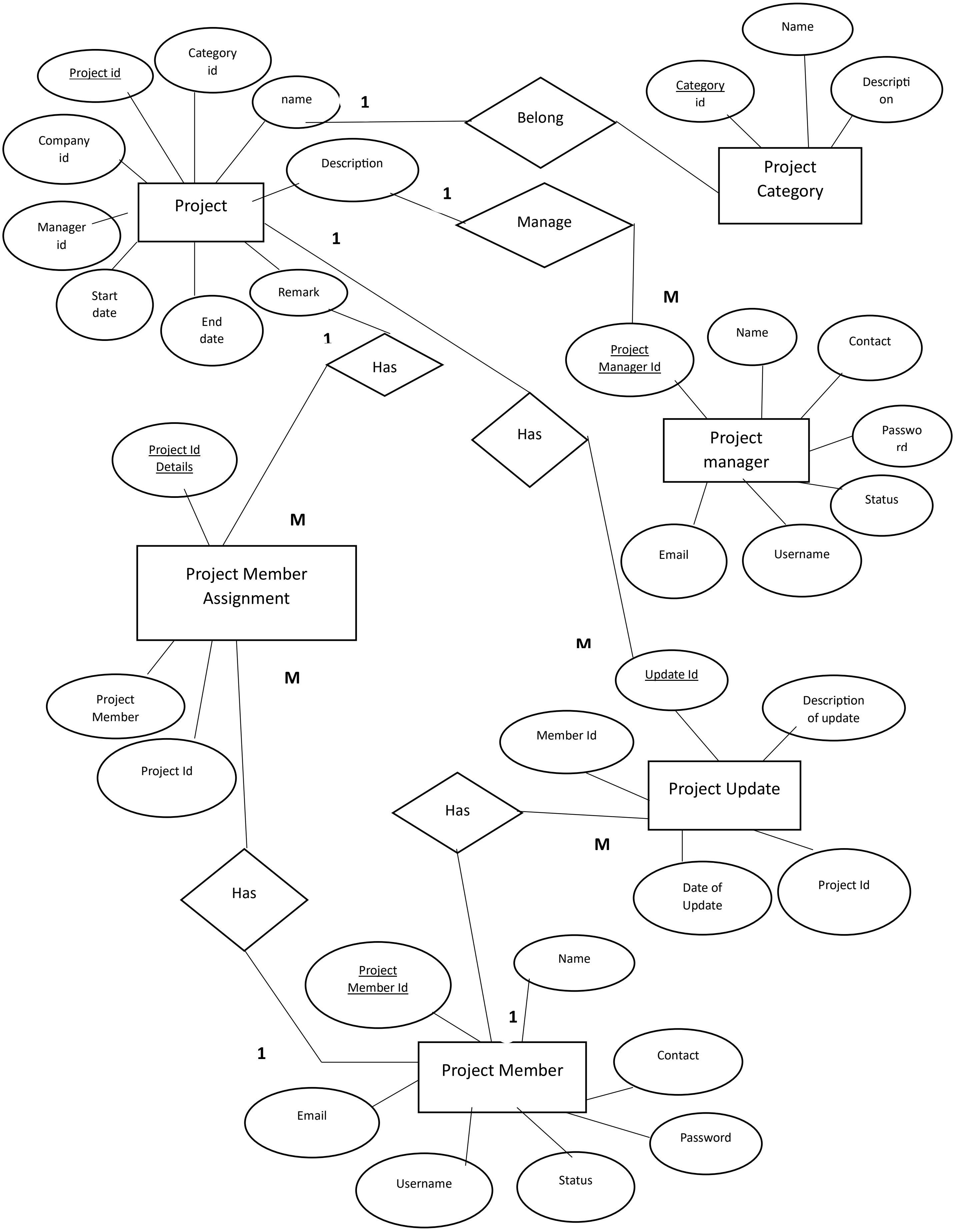
INDEX

ABSTRACT

The Project Management System is a comprehensive solution designed to optimize and streamline the task assignment and tracking processes within organizations. This project addresses the limitations of existing systems by introducing a user-friendly platform that empowers administrators to efficiently create, assign, and monitor tasks. Built on an Agile methodology, the system incorporates key features such as employee profiles, roles, and real-time task status tracking. With a focus on enhancing collaboration and transparency, the Project Management System aims to revolutionize task management, promoting organizational efficiency and accountability.

The project management system consists of entities such as company, project, project category, project manager, project update, project member and project member assignment. The tables consist of details of the employees, their passwords with hidden keys, authentication for the users, employee work status, and about the complete details of project. The data storage architecture for the Project Management System revolves around the effective utilization of a SQL database, offering a robust foundation for organizing, managing, and retrieving project-related information. Employing a relational database model, the system encapsulates project data within well-defined tables, each representing a key entity such as tasks, projects, and users. Tables are structured with columns corresponding to specific attributes, ensuring a systematic and organized storage of data. Primary keys uniquely identify records, facilitating efficient retrieval and modification. Relationships between entities are established through foreign keys, fostering coherence and integrity within the database.

ER DIAGRAM


Project id
Category id
name
Company id
Description
Project
Manager id
Start date
End date
Remark
1
Belong
1
Manage
1
Name
Category id
Descripti on
Project Category
M
Project Manager Id
Name
Contact
Project manager
Passwo rd
Status
Email
Username
1
Has
Has
Project Id Details
M
Project Member Assignment
M
Project Member
Project Id
M
Update Id
Description of update
Member Id
Project Update
Has
M
Date of Update
Project Id
Has
Project Member Id
Name
1
1
Project Member
Contact
Email
Password
Username
Status

TABLES

USER AUTHENTICATION

S.No	FIELD NAME	TYPE	SIZE	CONSTRAINTS
1	id	int	10	Primary key
2	password	varchar	255	Not null
3	Last_login	datetime		Not null
4	Is_superuser	int	10	Not null
5	username	varchar	50	Not null
6	First_name	varchar	50	Not null
7	Last_name	varchar	50	Not null
8	email	varchar	50	Not null
9	Is_staff	int	10	Not null
10	Is_active	int	10	Not null
11	Date_joined	datetime		Not null

PROJECTS

S.no	Field name	type	size	constraints
1	Id	int	10	Foreign key
2	description	varchar	255	Not null
3	Date_created	datetime		Not null
4	name	varchar	50	Not null

TASKS

S.No	Field Name	TYPE	SIZE	CONSTRAINT
1	id	int	10	Foreign key
2	title	varchar	50	Not null
3	description	varchar	255	Not null
4	Date_created	datetime		Not null
5	Due_date	datetime		Not null
6	status	varchar	50	Not null
7	Assignee_id	int	10	Not null
8	Project_id	int	10	Not null

USER PROFILE

S.No	Feid name	type	size	constraint
1	id	int	10	Foreign key
2	name	varchar	50	Not null
3	email	varcahar	50	Not null
4	bio	varchar	50	Not null
5	User_id	int	10	Not null
6	photo	varchar	50	Not null

SOURCE CODE

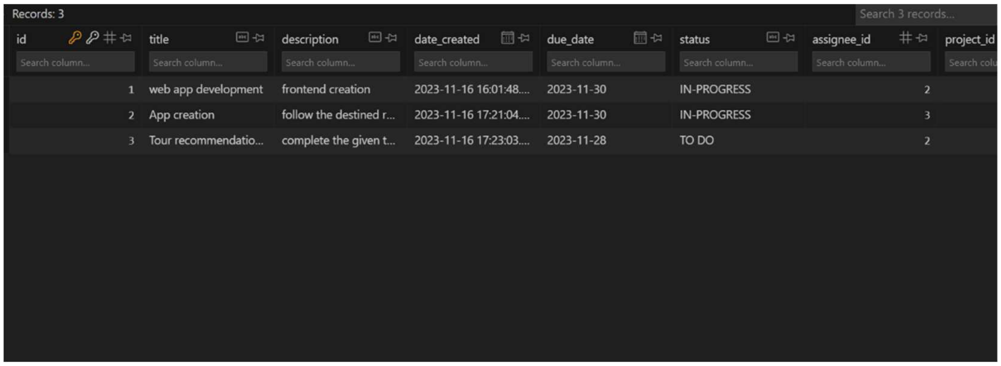
```
"""Parse SQL statements."""

# Setup namespace
from sqlparse import sql
from sqlparse import cli
from sqlparse import engine
from sqlparse import tokens
from sqlparse import filters
from sqlparse import formatter


__version__ = '0.4.4'
__all__ = ['engine', 'filters', 'formatter', 'sql', 'tokens', 'cli']


def parse(sql, encoding=None):
    """Parse sql and return a list of statements.

    :param sql: A string containing one or more SQL statements.
    :param encoding: The encoding of the statement (optional).
    :returns: A tuple of :class:`~sqlparse.sql.Statement` instances.
    """
    return tuple(parsestream(sql, encoding))
```

```
def parsestream(stream, encoding=None):
    """Parses sql statements from file-like object.

    :param stream: A file-like object.
    :param encoding: The encoding of the stream contents (optional).
    :returns: A generator of :class:`~sqlparse.sql.Statement` instances.
    """
    stack = engine.FilterStack()
    stack.enable_grouping()
    return stack.run(stream, encoding)


def format(sql, encoding=None, **options):
    """Format *sql* according to *options*.

    Available options are documented in :ref:`formatting`.

    In addition to the formatting options this function accepts the
    keyword "encoding" which determines the encoding of the statement.

    :returns: The formatted SQL statement as string.
    """
    stack = engine.FilterStack()
    options = formatter.validate_options(options)
    stack = formatter.build_filter_stack(stack, options)
    stack.postprocess.append(filters.SerializerUnicode())
    return ".join(stack.run(sql, encoding))
```

```
def split(sql, encoding=None):
    """Split *sql* into single statements.

    :param sql: A string containing one or more SQL statements.
    :param encoding: The encoding of the statement (optional).
    :returns: A list of strings.
    """
    stack = engine.FilterStack()
    return [str(stmt).strip() for stmt in stack.run(sql, encoding)]


import sys

from sqlparse.cli import main

if __name__ == '__main__':
    sys.exit(main())
```

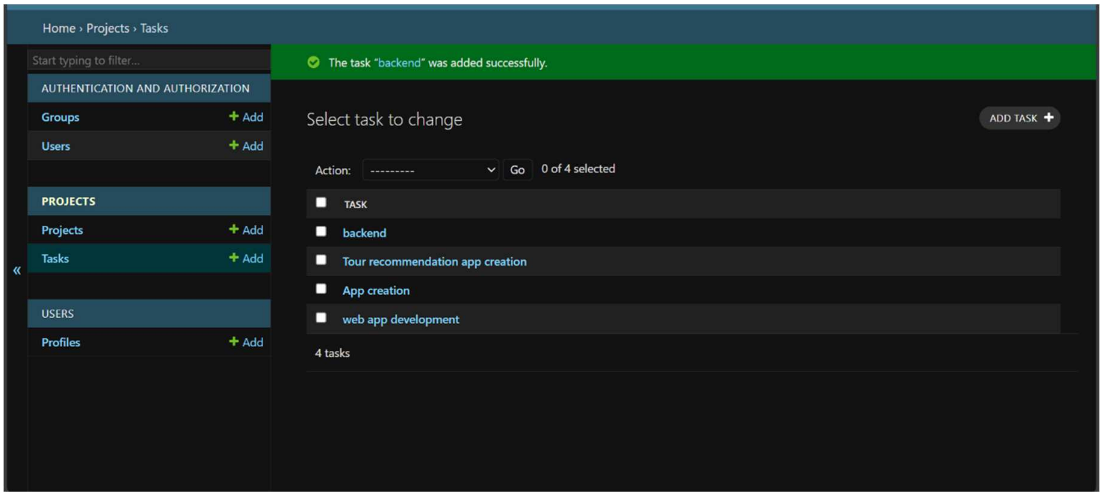
SCREENSHOTS

INSERTION OF RECORD



Records: 3

id	title	description	date_created	due_date	status	assignee_id	project_id
Search column...	Search column...	Search column...	Search column...	Search column...	Search column...	Search column...	Search colu
1	web app development	frontend creation	2023-11-16 16:01:48....	2023-11-30	IN-PROGRESS	2	
2	App creation	follow the destined r...	2023-11-16 17:21:04....	2023-11-30	IN-PROGRESS	3	
3	Tour recommendatio...	complete the given t...	2023-11-16 17:23:03....	2023-11-28	TO DO	2	



Home › Projects › Tasks

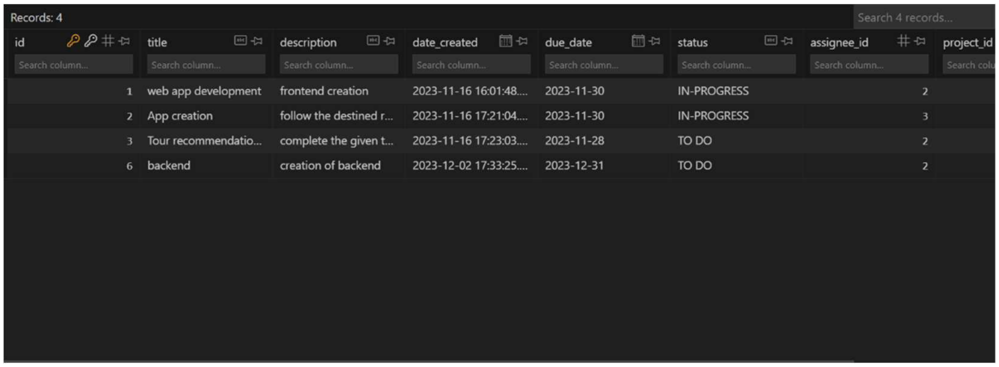
The task "backend" was added successfully.

Select task to change

Action: --------- Go 0 of 4 selected

- □ TASK
- □ backend
- □ Tour recommendation app creation
- □ App creation
- □ web app development

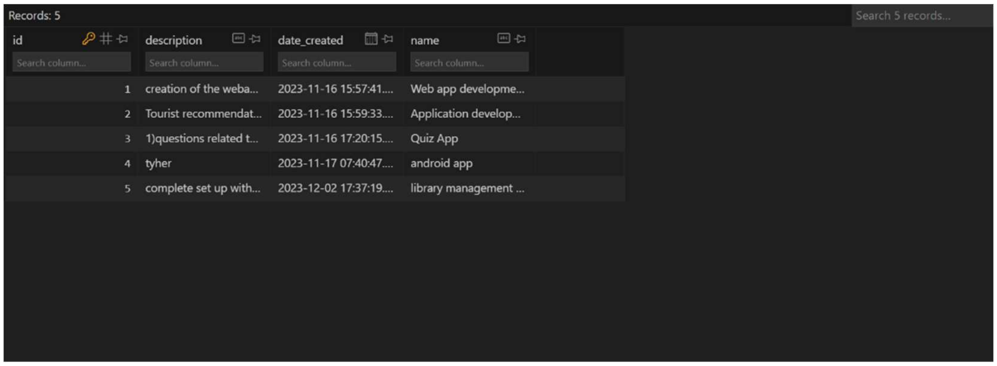
4 tasks



Records: 4

id	title	description	date_created	due_date	status	assignee_id	project_id
Search column...	Search column...	Search column...	Search column...	Search column...	Search column...	Search column...	Search colu
1	web app development	frontend creation	2023-11-16 16:01:48....	2023-11-30	IN-PROGRESS	2	
2	App creation	follow the destined r...	2023-11-16 17:21:04....	2023-11-30	IN-PROGRESS	3	
3	Tour recommendatio...	complete the given t...	2023-11-16 17:23:03....	2023-11-28	TO DO	2	
6	backend	creation of backend	2023-12-02 17:33:25....	2023-12-31	TO DO	2	

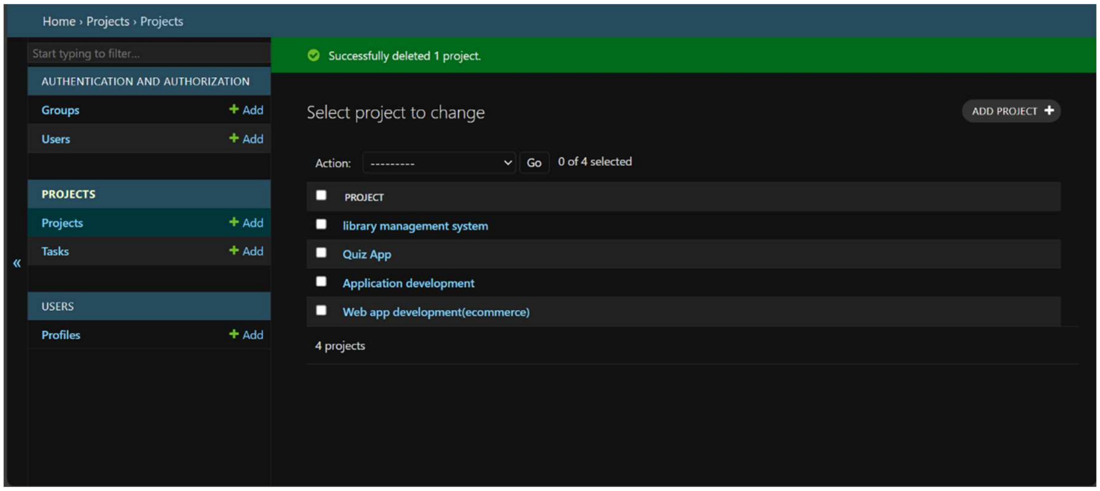
DELETION OF RECORDS



Records: 5

Search 5 records...

id	description	date_created	name
Search column...	Search column...	Search column...	Search column...
1	creation of the weba...	2023-11-16 15:57:41....	Web app developme...
2	Tourist recommendat...	2023-11-16 15:59:33....	Application develop...
3	1)questions related t...	2023-11-16 17:20:15....	Quiz App
4	tyher	2023-11-17 07:40:47....	android app
5	complete set up with...	2023-12-02 17:37:19....	library management ...



Home › Projects › Projects

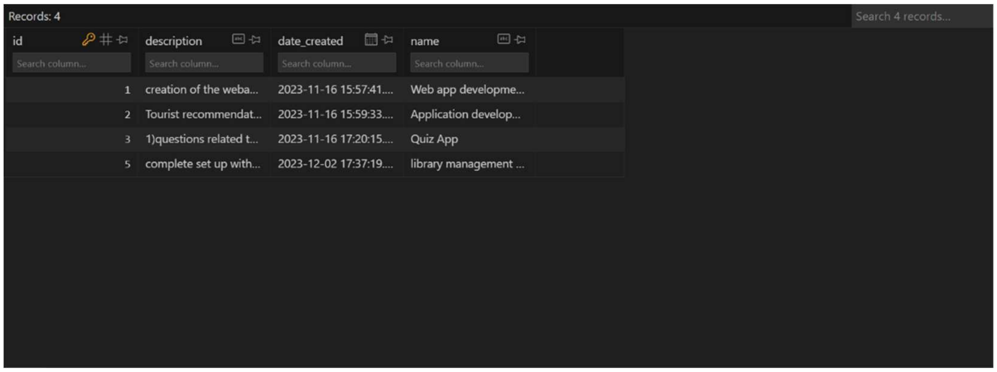
Successfully deleted 1 project.

Select project to change

Action: --------- Go 0 of 4 selected

□	PROJECT
□	library management system
□	Quiz App
□	Application development
□	Web app development(ecommerce)

4 projects



Records: 4

Search 4 records...

id	description	date_created	name
Search column...	Search column...	Search column...	Search column...
1	creation of the weba...	2023-11-16 15:57:41....	Web app developme...
2	Tourist recommendat...	2023-11-16 15:59:33....	Application develop...
3	1)questions related t...	2023-11-16 17:20:15....	Quiz App
5	complete set up with...	2023-12-02 17:37:19....	library management ...

REFERENCES

https://docs.djangoproject.com/en/4.2/howto/deployment/asgi/

The web framework for perfectionists with deadlines | Django (djangoproject.com)

URL dispatcher | Django documentation | Django (djangoproject.com)

CONCLUSION

In conclusion, the database architecture for our Project Management System serves as a robust foundation, providing a structured and organized environment for the storage, retrieval, and management of project-related data. The relational model, implemented through SQL, facilitates efficient organization of entities such as Users, Projects, and Tasks, ensuring data integrity and coherence.