



**EASWARI ENGINEERING COLLEGE
(AUTONOMOUS)**



MINI PROJECT REPORT

Submitted by

S NO	Name of the student	Register number
1.	ACHYUTH NARAYAN	310622148002
2.	BHARATH K	310622148008
3.	KAVIYA R V	310622148021
4.	NAVEEN KARTHIK R	310622148030
5.	RANJANA G	310622148033
6.	FAHMITHA FARHANA S	310622148012

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (AIML)

EASWARI ENGINEERING COLLEGE, CHENNAI 600 089

ANNA UNIVERSITY: CHENNAI 600 025

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**TIMETABLE MANAGEMENT SYSTEM**”
is the bonafide work carried out under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT

SIGNATURE

SUPERVISOR

**COMPUTER SCIENCE AND
ENGINEERING (AIML)
EASWARI ENGINEERING
COLLEGE, RAMAPURAM,
CHENNAI 600 089**

INDEX

S.NO	TITLE
1.	AIM
2.	ALGORITHM
3.	PROGRAM
4.	OUTPUT
5.	RESULT

TIMETABLE MANAGEMENT SYSTEM

AIM:

To develop Timetable management system using MySQL as a back end and python tkinter as front end.

ALGORITHM:

1. Database Setup:

- Create Database:
 - Create a database named timetable_db.
- Create Tables:
 - Create a teachers table with fields for id, name, and dept.
 - Create a subjects table with fields for id, code, name, mne, teacher_id, periods_as_per_syllabus, and allotted_periods.
 - Create a timetable table with fields for id, day, time_slot, and subject_id.

2. Python Tkinter GUI Setup:

- Create Main Window:
 - Initialize the main Tkinter window.
- Create Tabs:
 - Create tabs for "Add Teacher", "Add Subject", "Add Timetable Entry", and "View Timetable".
- Design Forms:
 - Design input forms for adding teachers, subjects, and timetable entries.

3. Form Actions:

- Add Teacher:
 - Capture input values for teacher name and department.
 - Insert these values into the teachers table in the database.

- Add Subject:

- Capture input values for subject code, name, mne, teacher ID, periods as per syllabus, and allotted periods.

- Insert these values into the subjects table in the database.

- Add Timetable Entry:

- Capture input values for day, time slot, and subject ID.

- Insert these values into the timetable table in the database.

- View Timetable:

- Query the timetable and subjects tables to retrieve timetable data.

- Display the retrieved timetable data in a grid format within the GUI.

4. Data Handling:

- Database Connection:

- Establish and manage a connection to the MySQL database.

- Error Handling:

- Implement error handling for database connections and query executions to ensure robustness.

5. Clear Data (Optional):

- Clear All Data:

- Provide functionality to delete all entries from teachers, subjects, and timetable tables for resetting the database.

6. Populating the Timetable:

- Fetch Subjects and Teachers:

- Retrieve all subjects and teachers from the database.

- Assign Timetable Slots:

- Randomly assign subjects to different time slots for each day to ensure no empty spaces.

- Update Timetable Table:

- Insert the assigned timetable slots into the timetable table.

PROGRAM:

```
CREATE DATABASE timetable_db;
```

```
USE timetable_db;
```

```
CREATE TABLE teachers (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(100) NOT NULL,
```

```
    dept VARCHAR(50) NOT NULL
```

```
);
```

```
CREATE TABLE subjects (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    code VARCHAR(20) NOT NULL,
```

```
    name VARCHAR(100) NOT NULL,
```

```
    mne VARCHAR(10) NOT NULL,
```

```
    teacher_id INT,
```

```
    periods_as_per_syllabus INT NOT NULL,
```

```
    allotted_periods INT NOT NULL,
```

```
    FOREIGN KEY (teacher_id) REFERENCES teachers(id)
```

```
);
```

```
CREATE TABLE timetable (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    day VARCHAR(10) NOT NULL,
```

```
    time_slot VARCHAR(20) NOT NULL,
```

```
    subject_id INT,
```

```
    FOREIGN KEY (subject_id) REFERENCES subjects(id)
```

```

);

import tkinter as tk

from tkinter import ttk, messagebox

import mysql.connector

def connect_db():

    return mysql.connector.connect(

        host="localhost",

        user="your_username",

        password="your_password",

        database="timetable_db"

    )

class TimetableApp:

    def __init__(self, root):

        self.root = root

        self.root.title("Timetable Management System")

        self.root.geometry("1000x600")

        self.create_widgets()

    def create_widgets(self):

        # Tabs

        self.tab_control = ttk.Notebook(self.root)

        self.tab_control.pack(expand=1, fill='both')

        self.tab_add_teacher = ttk.Frame(self.tab_control)

        self.tab_add_subject = ttk.Frame(self.tab_control)

        self.tab_add_timetable = ttk.Frame(self.tab_control)

```

```

self.tab_view_timetable = ttk.Frame(self.tab_control)

self.tab_control.add(self.tab_add_teacher, text='Add Teacher')

self.tab_control.add(self.tab_add_subject, text='Add Subject')

self.tab_control.add(self.tab_add_timetable, text='Add Timetable')

self.tab_control.add(self.tab_view_timetable, text='View Timetable')

self.create_add_teacher_tab()

self.create_add_subject_tab()

self.create_add_timetable_tab()

self.create_view_timetable_tab()


def create_add_teacher_tab(self):

    tk.Label(self.tab_add_teacher, text="Teacher Name:").grid(row=0, column=0,
padx=10, pady=10)

    self.teacher_name_entry = ttk.Entry(self.tab_add_teacher)

    self.teacher_name_entry.grid(row=0, column=1, padx=10, pady=10)


    tk.Label(self.tab_add_teacher, text="Department:").grid(row=1, column=0,
padx=10, pady=10)

    self.teacher_dept_entry = ttk.Entry(self.tab_add_teacher)

    self.teacher_dept_entry.grid(row=1, column=1, padx=10, pady=10)

    self.add_teacher_button = ttk.Button(self.tab_add_teacher, text="Add Teacher",
command=self.add_teacher)

    self.add_teacher_button.grid(row=2, column=0, columnspan=2, padx=10,
pady=10)


def create_add_subject_tab(self):

```



```
tk.Label(self.tab_add_subject, text="Subject Code:").grid(row=0, column=0,
padx=10, pady=10)
```

```
self.subject_code_entry = ttk.Entry(self.tab_add_subject)
```

```
self.subject_code_entry.grid(row=0, column=1, padx=10, pady=10)
```

```
tk.Label(self.tab_add_subject, text="Subject Name:").grid(row=1, column=0,
padx=10, pady=10)
```

```
self.subject_name_entry = ttk.Entry(self.tab_add_subject)
```

```
self.subject_name_entry.grid(row=1, column=1, padx=10, pady=10)
```

```
tk.Label(self.tab_add_subject, text="MNE:").grid(row=2, column=0, padx=10,
pady=10)
```

```
self.subject_mne_entry = ttk.Entry(self.tab_add_subject)
```

```
self.subject_mne_entry.grid(row=2, column=1, padx=10, pady=10)
```

```
tk.Label(self.tab_add_subject, text="Teacher ID:").grid(row=3, column=0,
padx=10, pady=10)
```

```
self.subject_teacher_id_entry = ttk.Entry(self.tab_add_subject)
```

```
self.subject_teacher_id_entry.grid(row=3, column=1, padx=10, pady=10)
```

```
tk.Label(self.tab_add_subject, text="Periods as per Syllabus:").grid(row=4,
column=0, padx=10, pady=10)
```

```
self.subject_periods_entry = ttk.Entry(self.tab_add_subject)
```

```
self.subject_periods_entry.grid(row=4, column=1, padx=10, pady=10)
```

```
tk.Label(self.tab_add_subject, text="Allotted Periods:").grid(row=5, column=0,
padx=10, pady=10)
```

```
self.subject_allotted_entry = ttk.Entry(self.tab_add_subject)
```

```

self.subject_allotted_entry.grid(row=5, column=1, padx=10, pady=10)

self.add_subject_button = ttk.Button(self.tab_add_subject, text="Add Subject",
command=self.add_subject)

self.add_subject_button.grid(row=6, column=0, columnspan=2, padx=10,
pady=10)

def create_add_timetable_tab(self):

    tk.Label(self.tab_add_timetable, text="Day:").grid(row=0, column=0, padx=10,
pady=10)

    self.day_entry = ttk.Entry(self.tab_add_timetable)

    self.day_entry.grid(row=0, column=1, padx=10, pady=10)

    tk.Label(self.tab_add_timetable, text="Time Slot:").grid(row=1, column=0,
padx=10, pady=10)

    self.time_slot_entry = ttk.Entry(self.tab_add_timetable)

    self.time_slot_entry.grid(row=1, column=1, padx=10, pady=10)

    tk.Label(self.tab_add_timetable, text="Subject ID:").grid(row=2, column=0,
padx=10, pady=10)

    self.subject_id_entry = ttk.Entry(self.tab_add_timetable)

    self.subject_id_entry.grid(row=2, column=1, padx=10, pady=10)

    self.add_timetable_button = ttk.Button(self.tab_add_timetable, text="Add
Timetable Entry", command=self.add_timetable_entry)

    self.add_timetable_button.grid(row=3, column=0, columnspan=2, padx=10,
pady=10)

```

```

def create_view_timetable_tab(self):

    self.timetable_grid = ttk.Frame(self.tab_view_timetable)

    self.timetable_grid.pack(expand=1, fill='both')

    self.view_button = ttk.Button(self.tab_view_timetable, text="Refresh
Timetable", command=self.view_timetable)

    self.view_button.pack(pady=10)

    self.clear_button = ttk.Button(self.tab_view_timetable, text="Clear All Data",
command=self.clear_data)

    self.clear_button.pack(pady=10)


def add_teacher(self):

    name = self.teacher_name_entry.get()

    dept = self.teacher_dept_entry.get()

    db = connect_db()

    cursor = db.cursor()

    try:

        cursor.execute("INSERT INTO teachers (name, dept) VALUES (%s, %s)",
(name, dept))

        db.commit()

        messagebox.showinfo("Success", "Teacher added successfully")

    except mysql.connector.Error as err:

        messagebox.showerror("Error", f"Error: {err}")

    finally:

        db.close()


def add_subject(self):

    code = self.subject_code_entry.get()

    name = self.subject_name_entry.get()

```

```

mne = self.subject_mne_entry.get()
teacher_id = self.subject_teacher_id_entry.get()
periods = self.subject_periods_entry.get()
allotted = self.subject_allotted_entry.get()

db = connect_db()
cursor = db.cursor()

try:
    cursor.execute("INSERT INTO subjects (code, name, mne, teacher_id,
periods_as_per_syllabus, allotted_periods) VALUES (%s, %s, %s, %s, %s, %s)",
                    (code, name, mne, teacher_id, periods, allotted))

    db.commit()

    messagebox.showinfo("Success", "Subject added successfully")
except mysql.connector.Error as err:
    messagebox.showerror("Error", f"Error: {err}")
finally:
    db.close()

def add_timetable_entry(self):
    day = self.day_entry.get()
    time_slot = self.time_slot_entry.get()
    subject_id = self.subject_id_entry.get()

    db = connect_db()
    cursor = db.cursor()

    try:
        cursor.execute("INSERT INTO timetable (day, time_slot, subject_id)
VALUES (%s, %s, %s)",

```

```

        (day, time_slot, subject_id))

    db.commit()

    messagebox.showinfo("Success", "Timetable entry added successfully")

except mysql.connector.Error as err:

    messagebox.showerror("Error", f"Error: {err}")

finally:

    db.close()


def view_timetable(self):

    for widget in self.timetable_grid.winfo_children():

        widget.destroy()


    db = connect_db()

    cursor = db.cursor()

    try:

        cursor.execute("""

            SELECT timetable.day, timetable.time_slot, subjects.mne

            FROM timetable

            JOIN subjects ON timetable.subject_id = subjects.id

            ORDER BY

            CASE

                WHEN timetable.day = 'MON' THEN 1

                WHEN timetable.day = 'TUE' THEN 2

                WHEN timetable.day = 'WED' THEN 3

                WHEN timetable.day = 'THU' THEN 4

                WHEN timetable.day = 'FRI' THEN 5

```

```

        ELSE 6

        END, timetable.time_slot

        """)

    rows = cursor.fetchall()

    days = ['MON', 'TUE', 'WED', 'THU', 'FRI']

    times = ["8.15-9.05 AM", "9.05-9.55 AM", "10.10-11.00 AM", "11.00-11.50
AM", "11.50-12.40 PM",

            "1.30-2.15 PM", "2.15-3.00 PM", "3.00-3.45 PM", "4.00-5.00 PM"]

    timetable = {day: {time: "" for time in times} for day in days}

    for row in rows:

        day, time_slot, mne = row

        timetable[day][time_slot] = mne

    for i, day in enumerate(days):

        tk.Label(self.timetable_grid, text=day, borderwidth=1,
relief="solid").grid(row=i+1, column=0, sticky="nsew")

    for j, time in enumerate(times):

        tk.Label(self.timetable_grid, text=time, borderwidth=1,
relief="solid").grid(row=0, column=j+1, sticky="nsew")

    for i, day in enumerate(days):

        for j, time in enumerate(times):

```

```
tk.Label(self.timetable_grid, text=timetable[day][time], borderwidth=1,
relief="solid").grid(row=i+1, column=j+1, sticky="nsew")
```

```
except mysql.connector.Error as err:
```

```
    messagebox.showerror("Error", f"Error: {err}")
```

```
finally:
```

```
    db.close()
```

```
def clear_data(self):
```

```
    db = connect_db()
```

```
    cursor = db.cursor()
```

```
    try:
```

```
        cursor.execute("DELETE FROM timetable")
```

```
        cursor.execute("DELETE FROM teachers")
```

```
        cursor.execute("DELETE FROM subjects")
```

```
        db.commit()
```

```
        messagebox.showinfo("Success", "All data cleared successfully")
```

```
except mysql.connector.Error as err:
```

```
    messagebox.showerror("Error", f"Error: {err}")
```

```
finally:
```

```
    db.close()
```

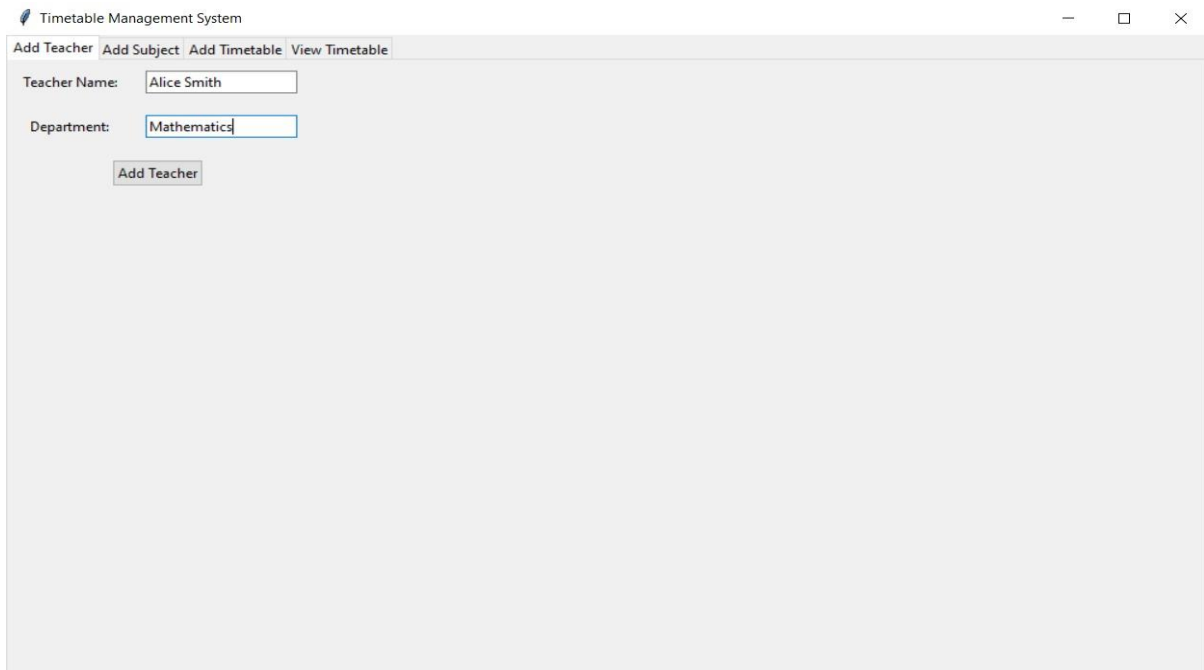
```
if __name__ == "__main__":
```

```
    root = tk.Tk()
```

```
    app = TimetableApp(root)
```


```
    root.mainloop()
```

OUTPUT:



The screenshot shows a window titled "Timetable Management System" with standard Windows window controls (minimize, maximize, close). The window has four tabs: "Add Teacher", "Add Subject", "Add Timetable", and "View Timetable". The "Add Teacher" tab is active. It contains two text input fields: "Teacher Name:" with the value "Alice Smith" and "Department:" with the value "Mathematics". Below these fields is a button labeled "Add Teacher".

 Success 

 Teacher added successfully

OK

Timetable Management System

— □ ×

Add Teacher Add Subject Add Timetable View Timetable

Subject Code: MATH101

Subject Name: Calculus

MNE: CAL

Teacher ID: 1

Periods as per Syllabus: 4

Allotted Periods: 4

Add Subject

Success

Subject added successfully

OK

Timetable Management System

— □ ×

Add Teacher Add Subject Add Timetable View Timetable

Day: MON

Time Slot: 8.15-9.05 AM

Subject ID: 1

Add Timetable Entry

Success

Timetable entry added successfully

OK

Timetable Management System

Add Teacher
Add Subject
Add Timetable
View Timetable

	8.15-9.05 AM	9.05-9.55 AM	10.10-11.00 AM	11.00-11.50 AM	11.50-12.40 PM	1.30-2.15 PM	2.15-3.00 PM	3.00-3.45 PM	4.00-5.00 PM
MON	CAL	MEC	ORG	BIO	LIT	CAL	MEC	ORG	BIO
TUE	LIT	CAL	MEC	ORG	BIO	LIT	CAL	MEC	ORG
WED	BIO	LIT	CAL	MEC	ORG	BIO	LIT	CAL	MEC
THU	ORG	BIO	LIT	CAL	MEC	ORG	BIO	LIT	CAL
FRI	MEC	ORG	BIO	LIT	CAL	MEC	ORG	BIO	LIT

Refresh Timetable

Clear All Data

```
mysql> use timetable_db
Database changed
mysql> show tables;
+-----+
| Tables_in_timetable_db |
+-----+
| subjects                |
| teachers                |
| timetable               |
+-----+
3 rows in set (0.10 sec)

mysql> select * from subjects;
+----+-----+-----+-----+-----+-----+-----+
| id | code | name          | mne | teacher_id | periods_as_per_syllabus | allotted_periods |
+----+-----+-----+-----+-----+-----+-----+
| 1  | MATH101 | Calculus      | CAL | 1          | 4                       | 4               |
| 2  | PHYS101 | Mechanics     | MEC | 2          | 4                       | 4               |
| 3  | CHEM101 | Organic Chemistry | ORG | 3          | 4                       | 4               |
| 4  | BIO101  | General Biology | BIO | 4          | 4                       | 4               |
| 5  | ENG101  | English Literature | LIT | 5          | 4                       | 4               |
+----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.06 sec)

mysql> select * from teachers;
+----+-----+-----+
| id | name      | dept      |
+----+-----+-----+
| 1  | Alice Smith | Mathematics |
| 2  | Bob Johnson  | Physics    |
| 3  | Charlie Brown | Chemistry  |
| 4  | David Wilson | Biology    |
| 5  | Eve Davis    | English    |
+----+-----+-----+
5 rows in set (0.05 sec)
```

```
mysql> select * from timetable;
```

id	day	time_slot	subject_id
1	MON	8.15-9.05 AM	1
2	MON	9.05-9.55 AM	2
3	MON	10.10-11.00 AM	3
4	MON	11.00-11.50 AM	4
5	MON	11.50-12.40 PM	5
6	MON	1.30-2.15 PM	1
7	MON	2.15-3.00 PM	2
8	MON	3.00-3.45 PM	3
9	MON	4.00-5.00 PM	4
10	TUE	8.15-9.05 AM	5
11	TUE	9.05-9.55 AM	1
12	TUE	10.10-11.00 AM	2
13	TUE	11.00-11.50 AM	3
14	TUE	11.50-12.40 PM	4
15	TUE	1.30-2.15 PM	5
16	TUE	2.15-3.00 PM	1
17	TUE	3.00-3.45 PM	2
18	TUE	4.00-5.00 PM	3
19	WED	8.15-9.05 AM	4
20	WED	9.05-9.55 AM	5
21	WED	10.10-11.00 AM	1
22	WED	11.00-11.50 AM	2
23	WED	11.50-12.40 PM	3
24	WED	1.30-2.15 PM	4
25	WED	2.15-3.00 PM	5
26	WED	3.00-3.45 PM	1
27	WED	4.00-5.00 PM	2
28	THU	8.15-9.05 AM	3
29	THU	9.05-9.55 AM	4
30	THU	10.10-11.00 AM	5
31	THU	11.00-11.50 AM	1
32	THU	11.50-12.40 PM	2
33	THU	1.30-2.15 PM	3
34	THU	2.15-3.00 PM	4
35	THU	3.00-3.45 PM	5
36	THU	4.00-5.00 PM	1
37	FRI	8.15-9.05 AM	2
38	FRI	9.05-9.55 AM	3
39	FRI	10.10-11.00 AM	4
40	FRI	11.00-11.50 AM	5
41	FRI	11.50-12.40 PM	1
42	FRI	1.30-2.15 PM	2
43	FRI	2.15-3.00 PM	3
44	FRI	3.00-3.45 PM	4
45	FRI	4.00-5.00 PM	5

```
45 rows in set (0.05 sec)
```

RESULT:

Thus, a program for hospital management system has been implemented and executed successfully.