INDEX

Acknowledgement

I extend my sincere gratitude to Mr. Christy for his invaluable guidance, unwavering support, and profound expertise throughout the development of the Proxy Maker project. His insightful advice and encouragement have been instrumental in shaping this project into a success.

His dedication to sharing knowledge, offering invaluable suggestions, and providing continuous motivation have been pivotal in navigating the challenges and achieving milestones in this project. His mentorship has not only enhanced the technical aspects but also contributed significantly to my growth as a professional.

I am profoundly thankful for his time, patience, and willingness to share expertise, which have been integral in shaping this project's outcomes. This project stands as a testament to his mentorship and support.

Thank you, Mr. Christy, for your unwavering guidance and mentorship throughout this endeavor.
Sincerely,
Sameer Sahu

Preface

Effective data management serves as the bedrock of modern operations, offering a spectrum of advantages across industries. It forms the backbone of informed decision-making, providing reliable insights crucial for strategic directions. This process streamlines operations, bolsters security measures, and enhances overall efficiency by ensuring swift access to accurate information, minimizing redundancy, and fostering productivity.

Moreover, proficient data management isn't just about organization; it's a catalyst for innovation and compliance. It enables businesses to personalize experiences, foresee market trends, and mitigate risks, propelling growth while ensuring adherence to stringent regulatory frameworks. Ultimately, mastering the art of data management isn't merely a functional requirement—it's a strategic imperative that empowers organizations to thrive in today's data-driven landscape.

Abstract

In an era defined by an unprecedented influx of information, the effective management of data stands as a cornerstone for organizational success. This abstract encapsulates the essence of a project devoted to exploring the multifaceted realm of data management within the context of contemporary operations.

This project delves into the pivotal role played by proficient data management practices across diverse industries. It examines how streamlined data organization, stringent security measures, and optimized accessibility foster informed decision-making, operational efficiency, and heightened productivity. Additionally, the abstract elucidates the transformative potential of data management in enabling personalized experiences for customers, fueling innovation, and ensuring compliance with regulatory frameworks.

By dissecting the intricacies of data management, this project aims to underscore its strategic significance as more than a functional necessity but as a catalyst for growth, risk mitigation, and informed decision-making. Through this exploration, it seeks to illuminate the imperative nature of mastering the art of data management in navigating the complex, data-driven landscape of modern enterprises.

Project Overview

The Proxy-Maker-system (PMS) project is a comprehensive endeavor designed to revolutionize the way data is handled, stored, and utilized within an organizational framework. Aimed at addressing the increasing complexity and volume of data, this project endeavors to create a robust and scalable system capable of facilitating seamless data operations.

At its core, the PMS focuses on three fundamental pillars: organization, and accessibility. By implementing meticulous data categorization and structuring methodologies, the system aims to streamline data storage and retrieval processes. Through this, it seeks to enhance operational efficiency while ensuring data accuracy and relevance.

Furthermore, the project emphasizes the democratization of data access, promoting ease of use and accessibility without compromising security measures. It aspires to offer intuitive interfaces and functionalities, enabling stakeholders to harness the power of data-driven insights for informed decision-making.

The PMS project, with its amalgamation of cutting-edge technology and strategic data management principles, seeks not only to optimize current data processes but also to lay a foundation for future scalability and adaptability. Through its implementation, it endeavors to empower organizations to harness the full potential of their data assets in driving innovation, enhancing operational efficiency, and fostering informed decision-making across diverse industries.

Features

- Day Handling: The code prompts for the current day and displays it throughout the program.

- Menu System: Utilizes a menu system to navigate through different functionalities and options.

- Teacher Management: Allows viewing all teachers, adding new teachers, deleting teachers, and displaying details of specific teachers.

- Proxy Generation: Generates proxies for absent teachers, showcasing their daily schedules and suggesting available teachers for substitution in different periods.

- Database Integration: Includes MySQL database integration for storing teacher information, managing their schedules, and retrieving data based on various queries.

- User Interaction: Offers a user-friendly interface with clear prompts, inputs, and visual representation of data through tabulated displays.

- Artifacts and Presentation: Incorporates ASCII art, centered text, and tables to improve visual appeal and readability of information within the console interface.

- Error Handling: Provides basic error handling for inputs and displays error messages for invalid inputs or unexpected errors.

- Modular Functions: Breaks down functionalities into smaller functions for specific tasks like menu rendering, database operations, and proxy creation, enhancing code readability and reusability.

- Credits Section: Includes a credits section showcasing project details and creator information.

This code aims to manage teacher scheduling, proxy creation, and database operations through a command-line interface, enabling users to efficiently handle teacher absences and schedule adjustments within an educational institution.

Objectives

- Develop an automated system for generating substitute teachers (proxies) based on absent teachers' schedules.

- Implement functionalities for adding, viewing, and deleting teacher details within a MySQL database.

- Display comprehensive daily schedules for absent teachers and available substitutes in a structured format.

- Create an intuitive interface with menus and prompts for user-friendly navigation.

- Implement robust error handling mechanisms to manage unexpected inputs.

- Utilize ASCII art, centered text, and tabulated displays to enhance visual presentation.

- Design the program with modularity and scalability for potential future enhancements.

- Enable administrators to make informed decisions by presenting clear information about teacher availability.

- Provide comprehensive project documentation outlining functionalities and database structure for reference.

User Interface

The user interface of this program is designed to provide a straightforward and interactive experience for administrators or users interacting with the system. It leverages a console-based interface that offers a series of menus, prompts, and visually enhanced displays to facilitate easy navigation and understanding of the program's functionalities.

Menu System: The interface employs a menu-driven system that guides users through various options and functionalities available within the program. This structured approach simplifies interaction, allowing users to select specific actions or tasks seamlessly.

Clear Prompts: Throughout the interaction, the program prompts users with clear and concise instructions, guiding them on how to input data, select options, or proceed with different operations. These prompts ensure users understand what actions are required at each stage.

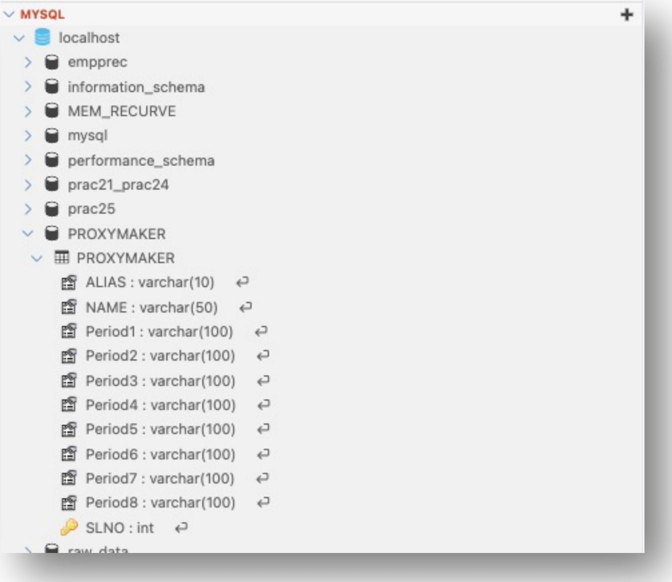
Tabulated Displays: To enhance readability and visualization of data, the interface presents information, such as teacher schedules or database details, in structured and tabulated formats. This presentation style improves comprehension and makes it easier for users to interpret complex data.

Visual Enhancements: Incorporating ASCII art and centered text, the interface adds visual appeal and aids in breaking the monotony of textual information. This aesthetic touch enhances the user experience and engagement with the program.

Error Handling: The interface includes mechanisms to handle unexpected user inputs or errors gracefully. It provides informative error messages or guidance to assist users in rectifying mistakes, ensuring a smoother user experience.
Modularity and Scalability: The interface is designed with modularity in mind, allowing for potential expansion or incorporation of additional features in the future. This scalable design ensures adaptability to evolving user needs without compromising usability.

Database

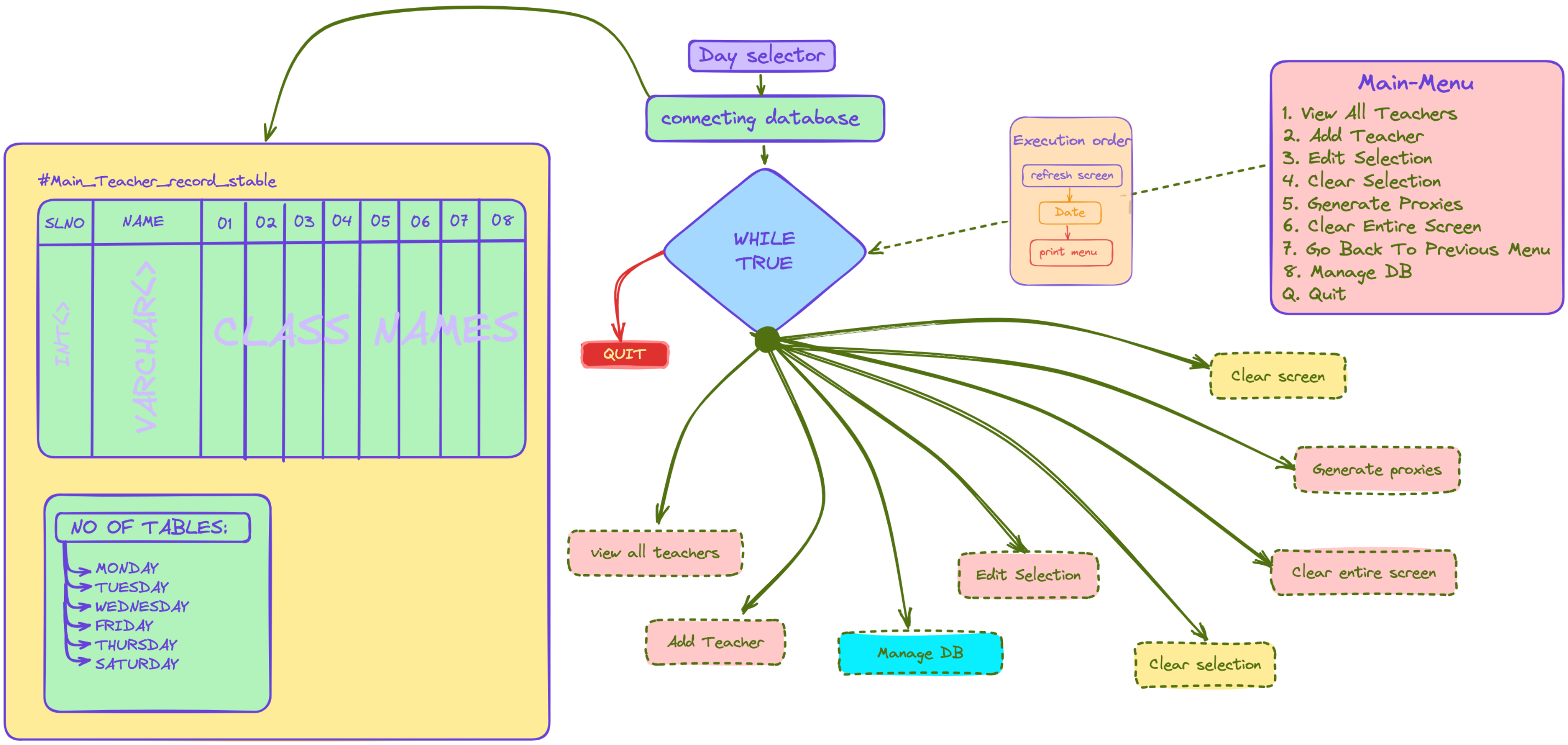
MySQL an open-source RDBMS, excels in managing data with its relational model, enabling efficient storage and retrieval through tables. Its scalability and speed cater to diverse needs, from small-scale applications to enterprise-level systems, ensuring seamless performance. MySQL's user-friendly interface, extensive community support, and robust security features make it a popular choice, offering reliability and adaptability for various projects and environments.



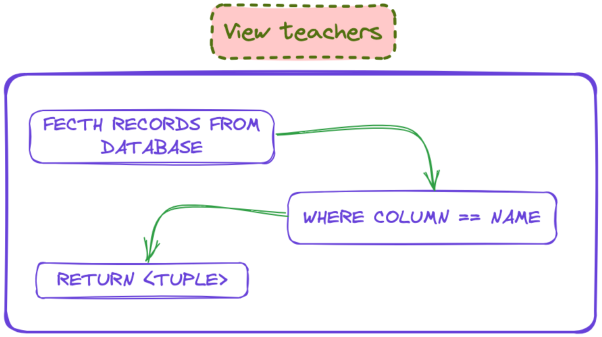
```
MYSQL                                          +
  localhost
    empprec
    information_schema
    MEM_RECURVE
    mysql
    performance_schema
    prac21_prac24
    prac25
    PROXYMAKER
      PROXYMAKER
        ALIAS : varchar(10)
        NAME : varchar(50)
        Period1 : varchar(100)
        Period2 : varchar(100)
        Period3 : varchar(100)
        Period4 : varchar(100)
        Period5 : varchar(100)
        Period6 : varchar(100)
        Period7 : varchar(100)
        Period8 : varchar(100)
        SLNO : int
    raw_data
```

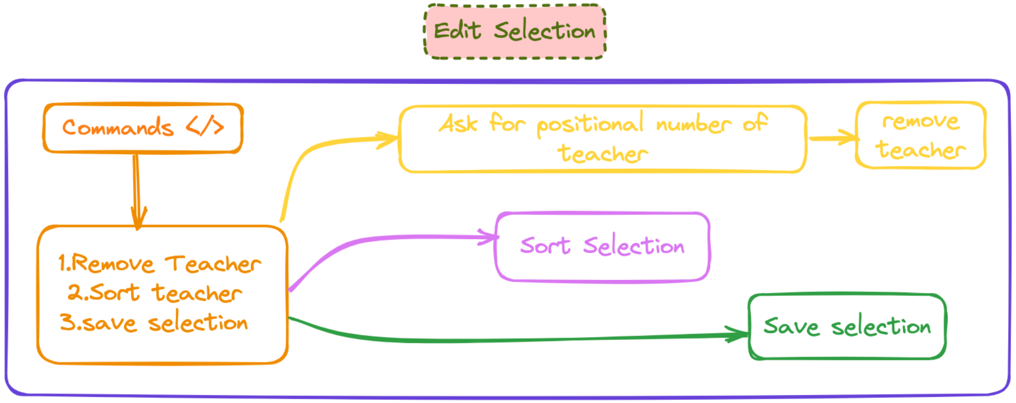
Tabulate is a Python library facilitating the creation of formatted tables from various data sources. It simplifies data visualization by generating tables in a readable, structured format, ideal for console output or documentation. With its intuitive API and support for different table formats, such as ASCII, Markdown, or HTML, Tabulate streamlines the presentation of complex data sets. This library offers flexibility in customizing table styles, headers, and data alignment, making it an excellent tool for enhancing the visual representation of information within Python applications.

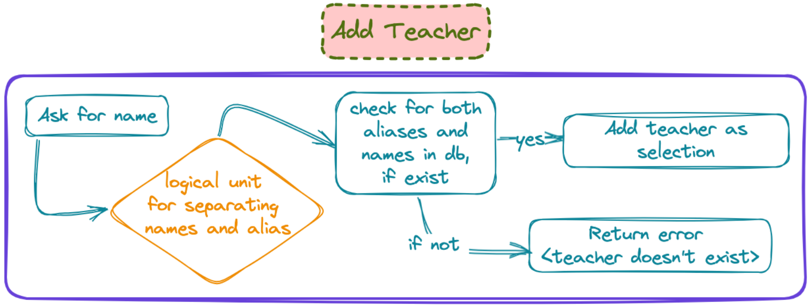
Algorithms

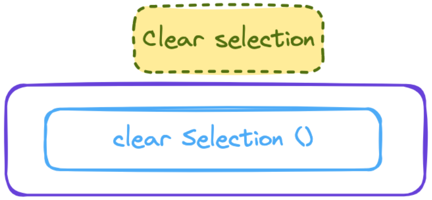


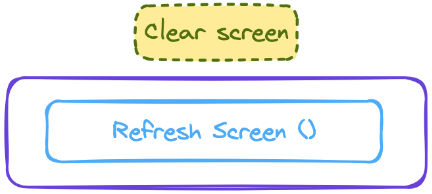
SLNO	NAME	01	02	03	04	05	06	07	08
INT<>	VARCHAR<>	CLASS NAMES							

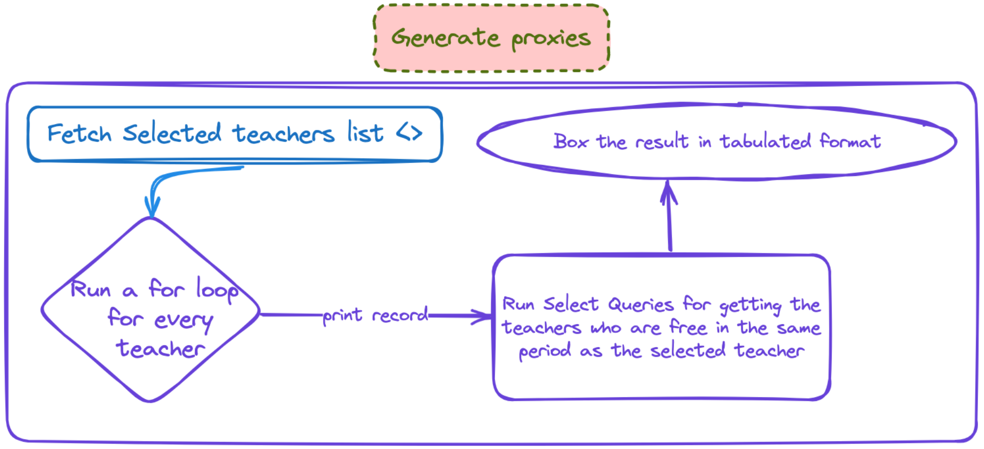

View teachers
FECTH RECORDS FROM DATABASE
WHERE COLUMN == NAME
RETURN <TUPLE>

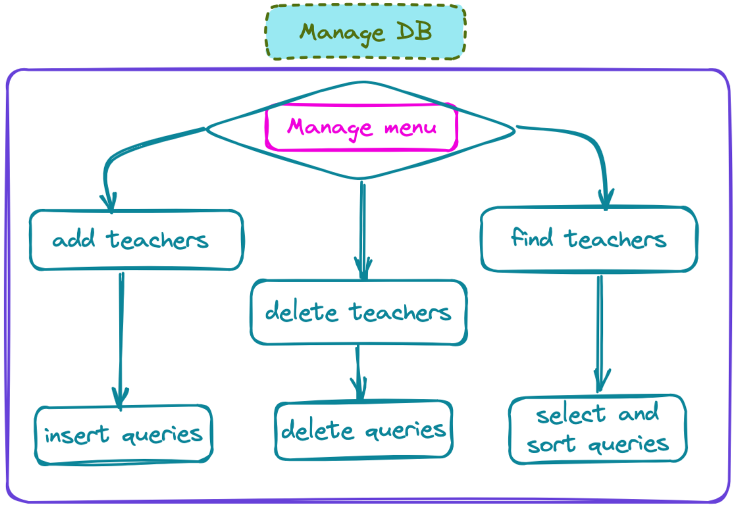

Edit Selection
Commands </>
1.Remove Teacher
2.Sort teacher
3.save selection
Ask for positional number of teacher
remove teacher
Sort Selection
Save selection


Add Teacher
Ask for name
logical unit for separating names and alias
check for both aliases and names in db, if exist
yes
Add teacher as selection
if not
Return error <teacher doesn't exist>


Clear selection
clear Selection ()

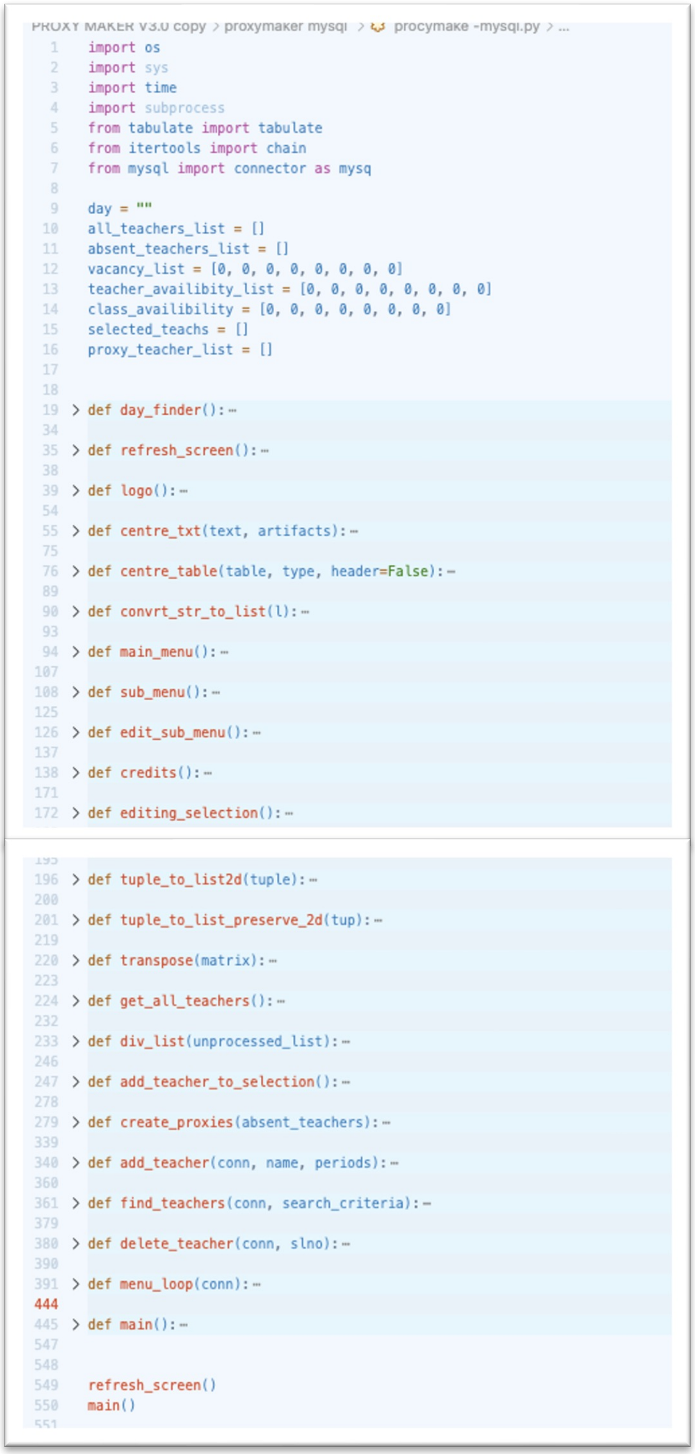

Clear screen
Refresh Screen ()


Generate proxies
Fetch Selected teachers list <>
Run a for loop for every teacher
print record
Run Select Queries for getting the teachers who are free in the same period as the selected teacher
Box the result in tabulated format


Manage DB
Manage menu
add teachers
delete teachers
find teachers
insert queries
delete queries
select and sort queries

Program

#Program function structure:

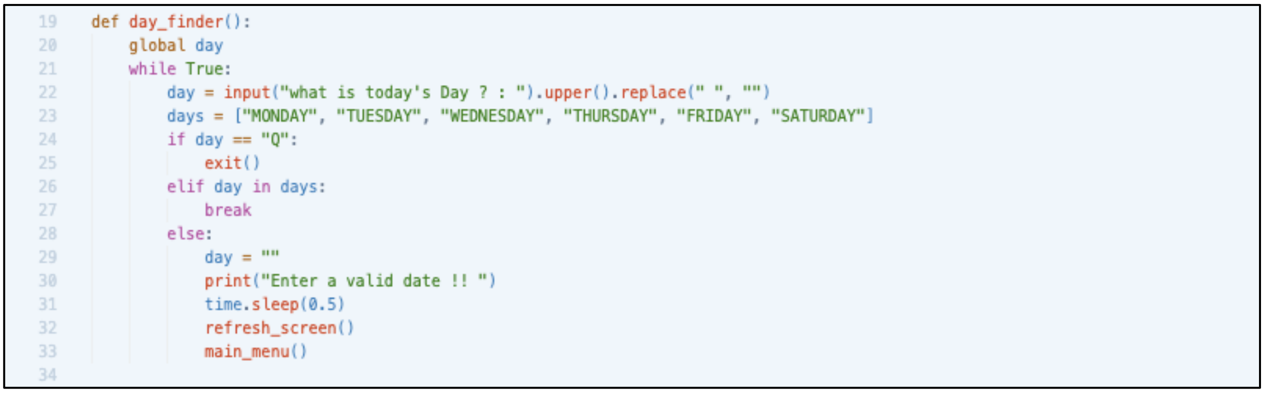


PROXY MAKER V3.0 copy > proxymaker mysql > procymake -mysql.py > ...

```
import os
import sys
import time
import subprocess
from tabulate import tabulate
from itertools import chain
from mysql import connector as mysq

day = ""
all_teachers_list = []
absent_teachers_list = []
vacancy_list = [0, 0, 0, 0, 0, 0, 0, 0]
teacher_availibity_list = [0, 0, 0, 0, 0, 0, 0, 0]
class_availibility = [0, 0, 0, 0, 0, 0, 0, 0]
selected_teachs = []
proxy_teacher_list = []


def day_finder():⋯

def refresh_screen():⋯

def logo():⋯

def centre_txt(text, artifacts):⋯

def centre_table(table, type, header=False):⋯

def convrt_str_to_list(l):⋯

def main_menu():⋯

def sub_menu():⋯

def edit_sub_menu():⋯

def credits():⋯

def editing_selection():⋯
```

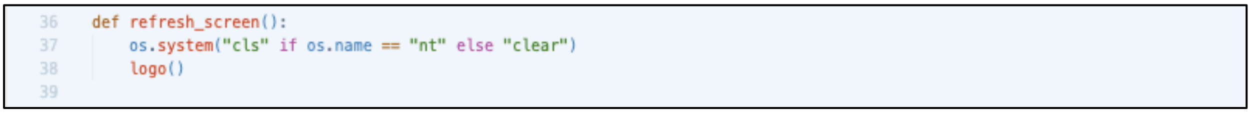
```
def tuple_to_list2d(tuple):⋯

def tuple_to_list_preserve_2d(tup):⋯

def transpose(matrix):⋯

def get_all_teachers():⋯

def div_list(unprocessed_list):⋯

def add_teacher_to_selection():⋯

def create_proxies(absent_teachers):⋯

def add_teacher(conn, name, periods):⋯

def find_teachers(conn, search_criteria):⋯

def delete_teacher(conn, slno):⋯

def menu_loop(conn):⋯

def main():⋯


refresh_screen()
main()
```

#Day-Finder:



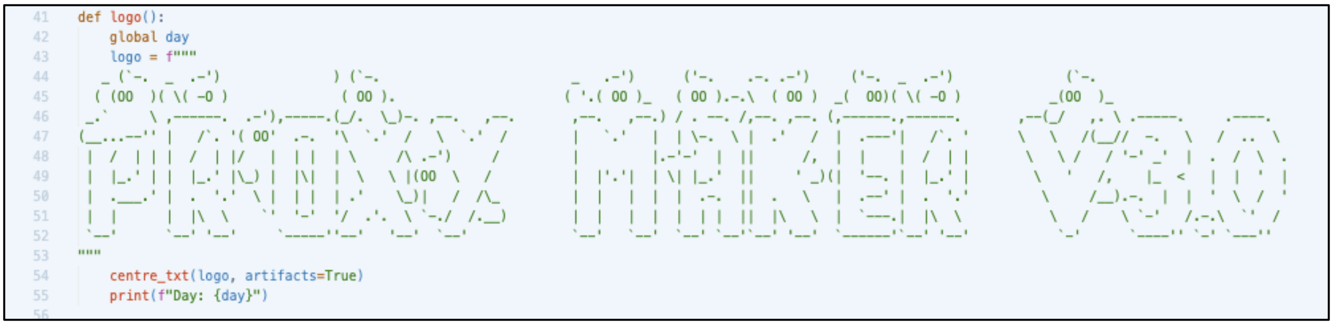
```python
def day_finder():
    global day
    while True:
        day = input("what is today's Day ? : ").upper().replace(" ", "")
        days = ["MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"]
        if day == "Q":
            exit()
        elif day in days:
            break
        else:
            day = ""
            print("Enter a valid date !! ")
            time.sleep(0.5)
            refresh_screen()
            main_menu()

```

#Refresh-screen :



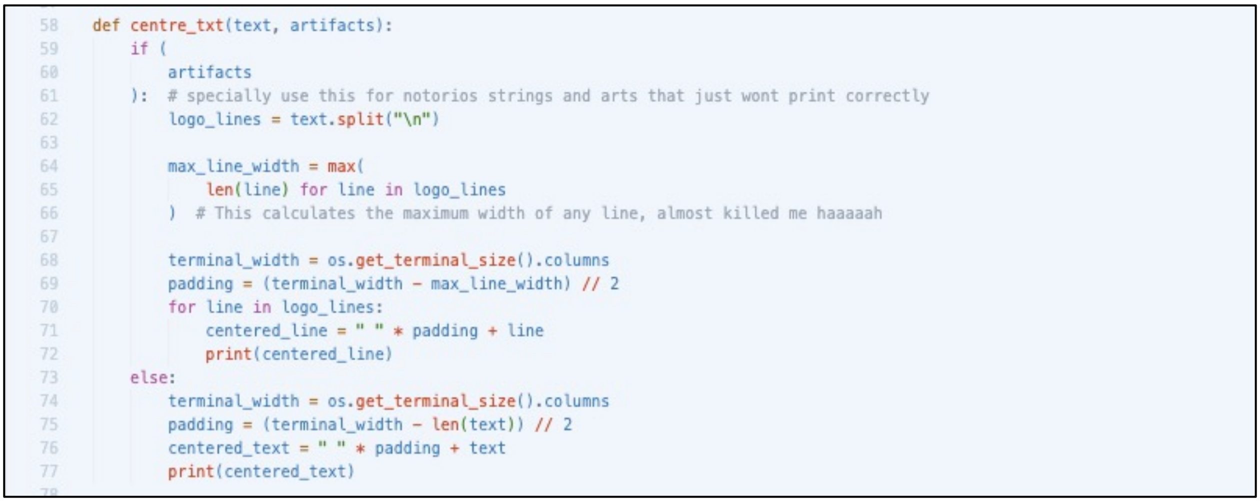
```python
def refresh_screen():
    os.system("cls" if os.name == "nt" else "clear")
    logo()

```

#Logo :



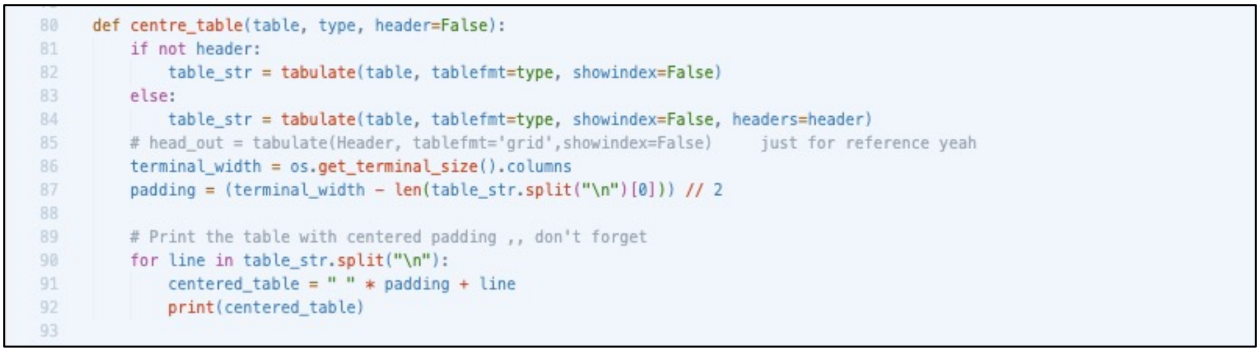
```python
def logo():
    global day
    logo = f"""
   _ (`-.  _  .-')             ) (`-.                      _   .-')       ('-.     .-. .-')     ('-.  _  .-')                (`-.
  ( (OO  )( \( -O )            ( OO ).                    ( '.( OO )_   ( OO ).-.\  ( OO )  _(  OO)( \( -O )            _(OO  )_
 _.`     \ ,------.  .-'),-----.(_/.  \_)-. ,--.   ,--.    ,--.   ,--.) / . --. /,--. ,--. (,------.,------.      ,--(_/   ,. \ .-----.      .----.
(__...--'' |   /`. '( OO'  .-.  '\  `.'  /   \  `.'  /     |   `.'   |  | \-.  \ |  .'   /  |  .---'|   /`. '     \   \   /(__/ /  -.   \    /  ..  \
 |  /  | | |  /  | |/   |  | |  | \     /\ .-')     /      |         |.-'-'  |  ||      /,  |  |    |  /  | |      \   \ /   / '-' _'  |  .  /  \  .
 |  |_.' | |  |_.' |\_) |  |\|  |  \   \ |(OO  \   /       |  |'.'|  | \| |_.'  ||     ' _)(|  '--. |  |_.' |       \   '   /,    |_  <   |  |  '  |
 |  .___.' |  .  '.'  \ |  | |  | .'    \_)|   /  /\_      |  |   |  |  |  .-.  ||  .   \  |  .--' |  .  '.'        \     /__).-.  |  |  '  \  /  '
 |  |      |  |\  \    `'  '-'  '/  .'.  \ `-./  /.__)     |  |   |  |  |  | |  ||  |\   \ |  `---.|  |\  \          \   /    \ `-'   /.-.\  `'  /
 `--'      `--' '--'     `-----''--'   '--'  `--'          `--'   `--'  `--' `--'`--' '--'  `------'`--' '--'          `-'      `----''  `-' `---''
"""
    centre_txt(logo, artifacts=True)
    print(f"Day: {day}")

```

#Centering any text:



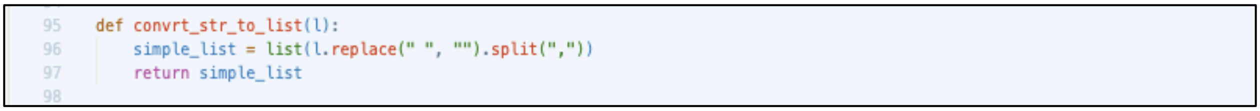
```python
def centre_txt(text, artifacts):
    if (
        artifacts
    ):  # specially use this for notorios strings and arts that just wont print correctly
        logo_lines = text.split("\n")

        max_line_width = max(
            len(line) for line in logo_lines
        )  # This calculates the maximum width of any line, almost killed me haaaaah

        terminal_width = os.get_terminal_size().columns
        padding = (terminal_width - max_line_width) // 2
        for line in logo_lines:
            centered_line = " " * padding + line
            print(centered_line)
    else:
        terminal_width = os.get_terminal_size().columns
        padding = (terminal_width - len(text)) // 2
        centered_text = " " * padding + text
        print(centered_text)
```

#Centering any table:



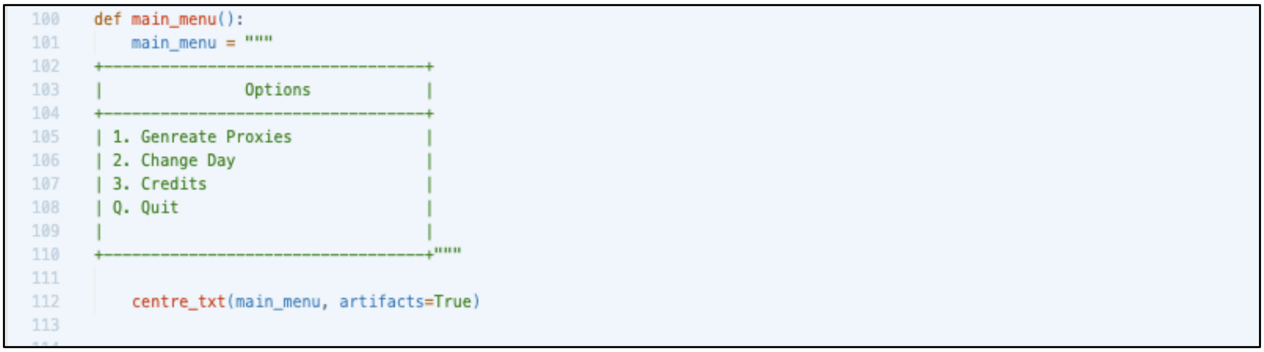
```python
def centre_table(table, type, header=False):
    if not header:
        table_str = tabulate(table, tablefmt=type, showindex=False)
    else:
        table_str = tabulate(table, tablefmt=type, showindex=False, headers=header)
    # head_out = tabulate(Header, tablefmt='grid',showindex=False)     just for reference yeah
    terminal_width = os.get_terminal_size().columns
    padding = (terminal_width - len(table_str.split("\n")[0])) // 2

    # Print the table with centered padding ,, don't forget
    for line in table_str.split("\n"):
        centered_table = " " * padding + line
        print(centered_table)
```

#Converting any str to list :



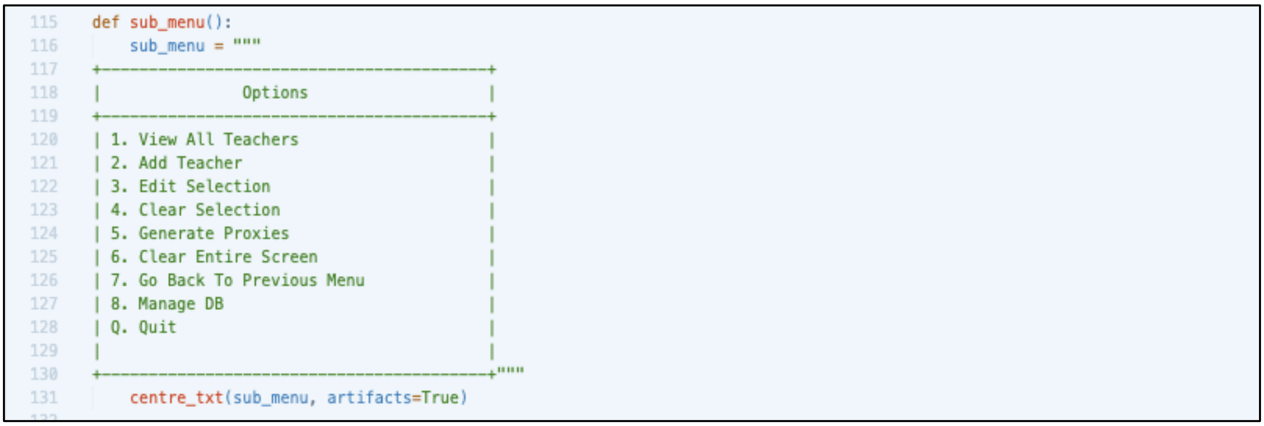
```python
def convrt_str_to_list(l):
    simple_list = list(l.replace(" ", "").split(","))
    return simple_list
```

#Main-menu:



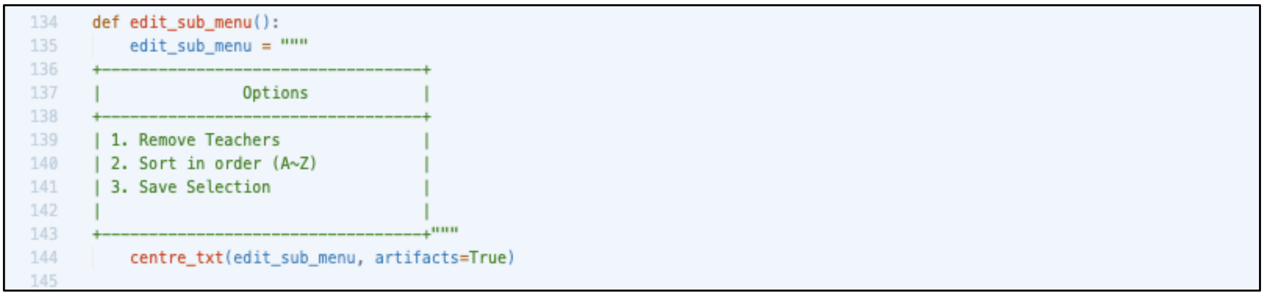
```python
def main_menu():
    main_menu = """
+----------------------------------+
|               Options            |
+----------------------------------+
| 1. Genreate Proxies              |
| 2. Change Day                    |
| 3. Credits                       |
| Q. Quit                          |
|                                  |
+----------------------------------+"""

    centre_txt(main_menu, artifacts=True)

```

#Sub-menu:



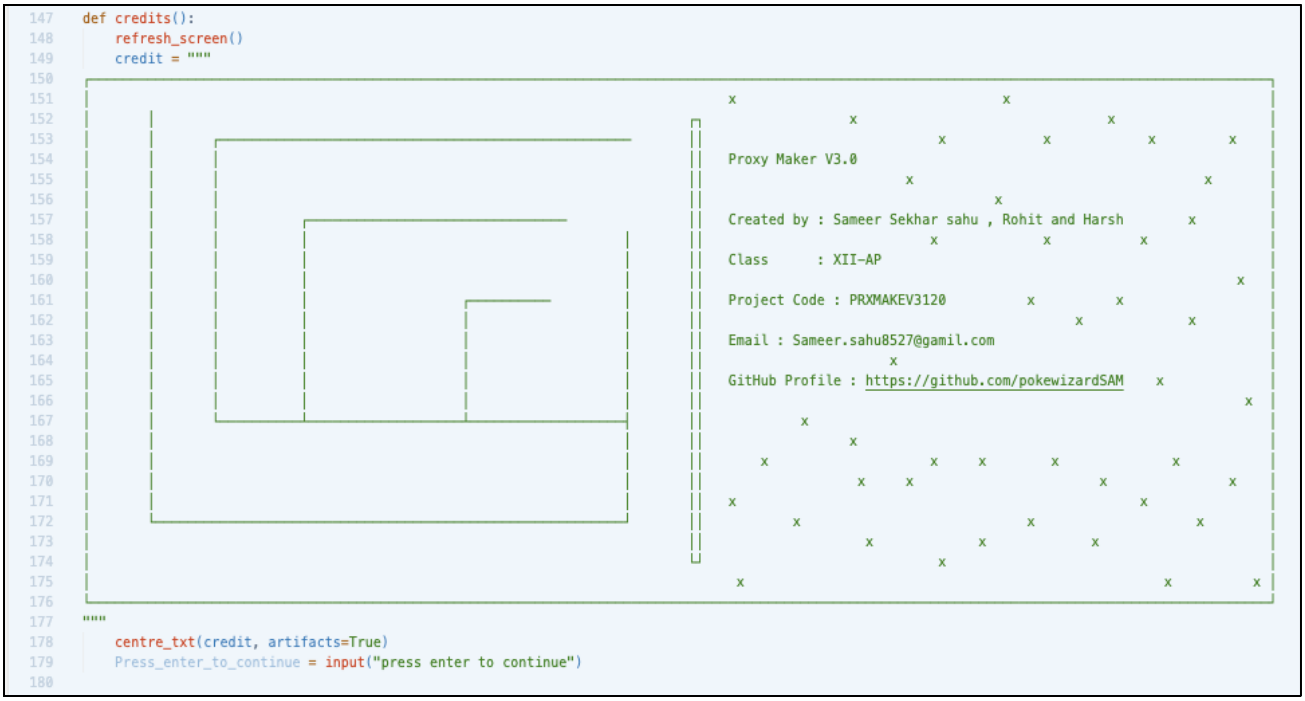
```python
def sub_menu():
    sub_menu = """
+-----------------------------------------+
|               Options                   |
+-----------------------------------------+
| 1. View All Teachers                    |
| 2. Add Teacher                          |
| 3. Edit Selection                       |
| 4. Clear Selection                      |
| 5. Generate Proxies                     |
| 6. Clear Entire Screen                  |
| 7. Go Back To Previous Menu             |
| 8. Manage DB                            |
| Q. Quit                                 |
|                                         |
+-----------------------------------------+"""
    centre_txt(sub_menu, artifacts=True)
```

#Editing-sub-menu:



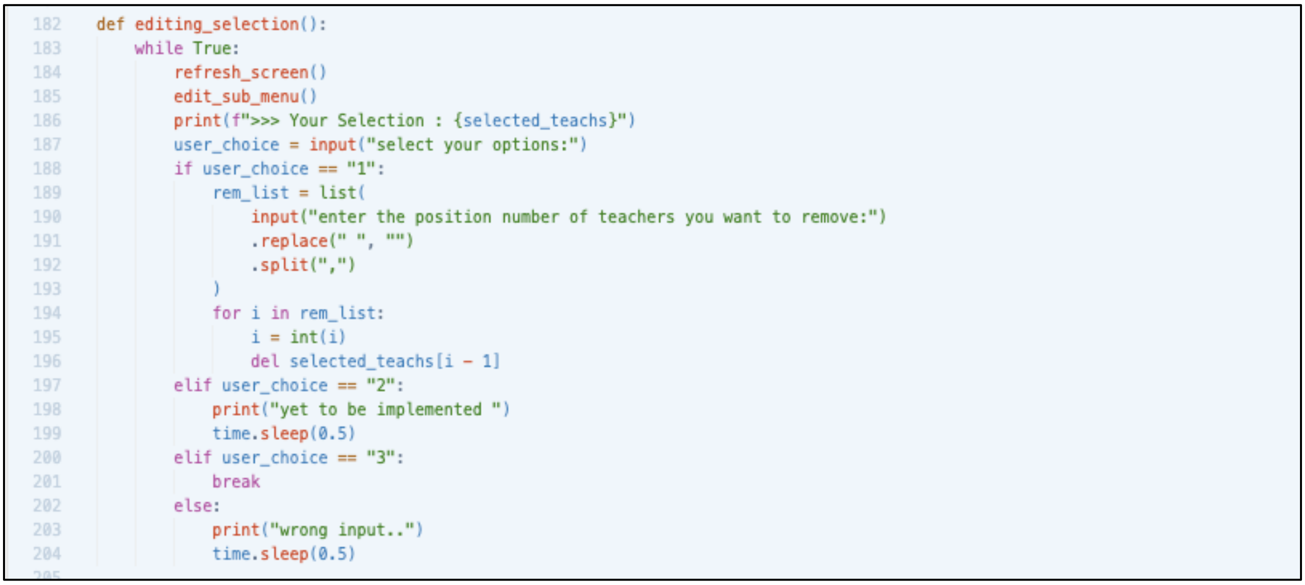
```python
def edit_sub_menu():
    edit_sub_menu = """
+----------------------------------+
|               Options            |
+----------------------------------+
| 1. Remove Teachers               |
| 2. Sort in order (A~Z)           |
| 3. Save Selection                |
|                                  |
+----------------------------------+"""
    centre_txt(edit_sub_menu, artifacts=True)

```

#Credits:



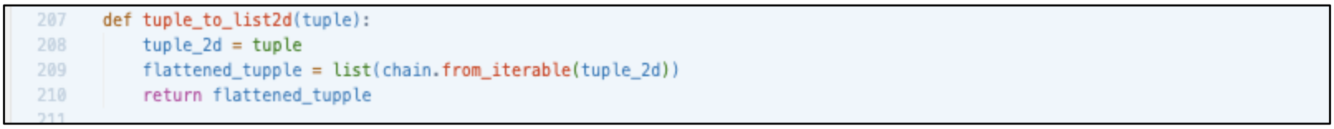
```
def credits():
    refresh_screen()
    credit = """
┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│                                                                               x                                 x                                │
│       │                                                                  ┌┐                  x                               x                   │
│       │       ┌────────────────────────────────────────────────────      ││                             x            x            x         x    │
│       │       │                                                          ││   Proxy Maker V3.0                                                   │
│       │       │                                                          ││                         x                                    x       │
│       │       │                                                          ││                                  x                                 │
│       │       │          ┌─────────────────────────────────              ││   Created by : Sameer Sekhar sahu , Rohit and Harsh        x         │
│       │       │          │                                       │       ││                            x             x           x               │
│       │       │          │                                       │       ││   Class      : XII-AP                                                │
│       │       │          │                                       │       ││                                                                x   │
│       │       │          │                   ┌───────────        │       ││   Project Code : PRXMAKEV3120          x          x                  │
│       │       │          │                   │                   │       ││                                              x             x         │
│       │       │          │                   │                   │       ││   Email : Sameer.sahu8527@gamil.com                                  │
│       │       │          │                   │                   │       ││                       x                                              │
│       │       │          │                   │                   │       ││   GitHub Profile : https://github.com/pokewizardSAM    x             │
│       │       │          │                   │                   │       ││                                                                 x  │
│       │       └──────────┴───────────────────┴───────────────────┤       ││            x                                                         │
│       │                                                          │       ││                  x                                                   │
│       │                                                          │       ││       x                    x     x        x              x           │
│       │                                                          │       ││                   x     x                       x               x    │
│       │                                                          │       ││   x                                                  x               │
│       └──────────────────────────────────────────────────────────┘       ││           x                            x                    x        │
│                                                                          ││                    x             x             x                     │
│                                                                          └┘                             x                                        │
│                                                                                x                                                    x          x │
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘
"""
    centre_txt(credit, artifacts=True)
    Press_enter_to_continue = input("press enter to continue")
```

#Editing-selected-teachers:



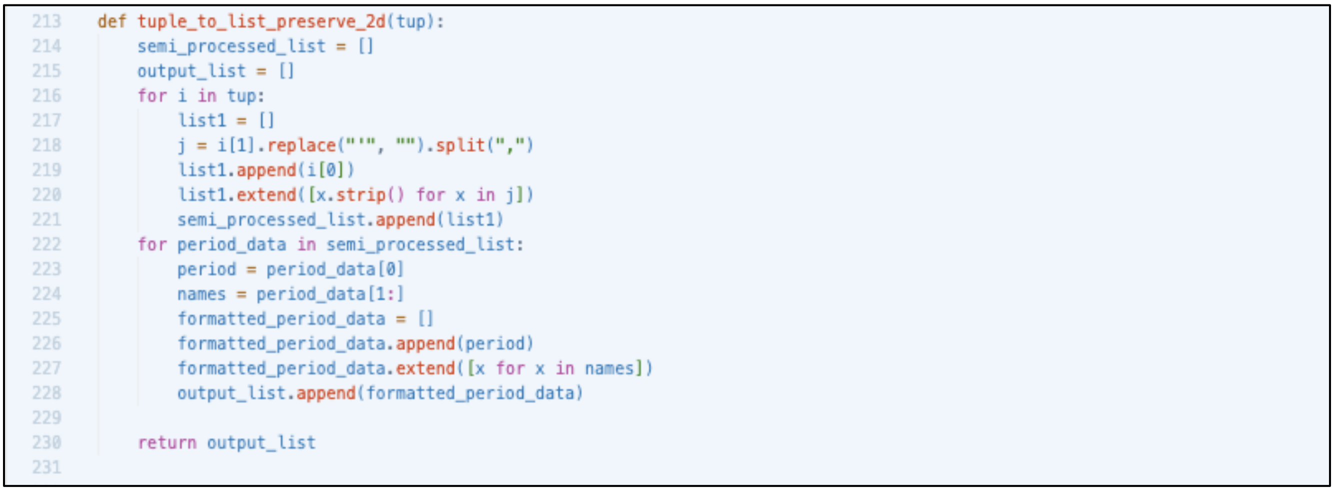
```
def editing_selection():
    while True:
        refresh_screen()
        edit_sub_menu()
        print(f">>> Your Selection : {selected_teachs}")
        user_choice = input("select your options:")
        if user_choice == "1":
            rem_list = list(
                input("enter the position number of teachers you want to remove:")
                .replace(" ", "")
                .split(",")
            )
            for i in rem_list:
                i = int(i)
                del selected_teachs[i - 1]
        elif user_choice == "2":
            print("yet to be implemented ")
            time.sleep(0.5)
        elif user_choice == "3":
            break
        else:
            print("wrong input..")
            time.sleep(0.5)
```

#Tuple-to-list-destroy-2d:



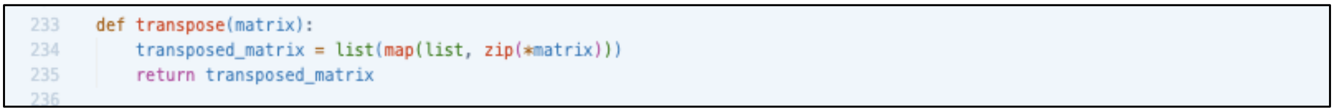
```python
def tuple_to_list2d(tuple):
    tuple_2d = tuple
    flattened_tupple = list(chain.from_iterable(tuple_2d))
    return flattened_tupple
```

#Tuple-to-list-preserve-2d:



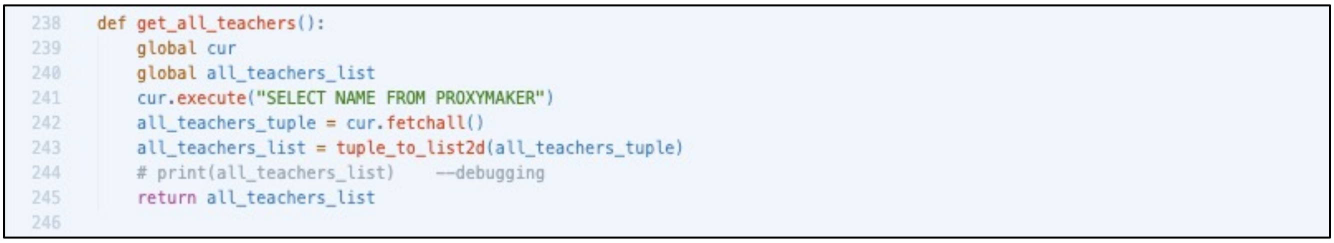
```python
def tuple_to_list_preserve_2d(tup):
    semi_processed_list = []
    output_list = []
    for i in tup:
        list1 = []
        j = i[1].replace("'", "").split(",")
        list1.append(i[0])
        list1.extend([x.strip() for x in j])
        semi_processed_list.append(list1)
    for period_data in semi_processed_list:
        period = period_data[0]
        names = period_data[1:]
        formatted_period_data = []
        formatted_period_data.append(period)
        formatted_period_data.extend([x for x in names])
        output_list.append(formatted_period_data)

    return output_list
```

#Transposing-an-array:



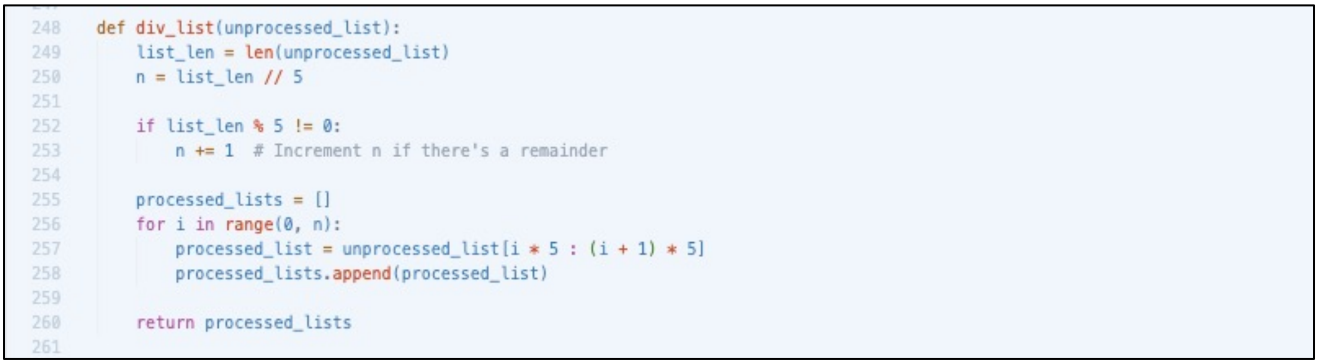
```python
def transpose(matrix):
    transposed_matrix = list(map(list, zip(*matrix)))
    return transposed_matrix
```

#Fetching-all-teachers-from-DB:



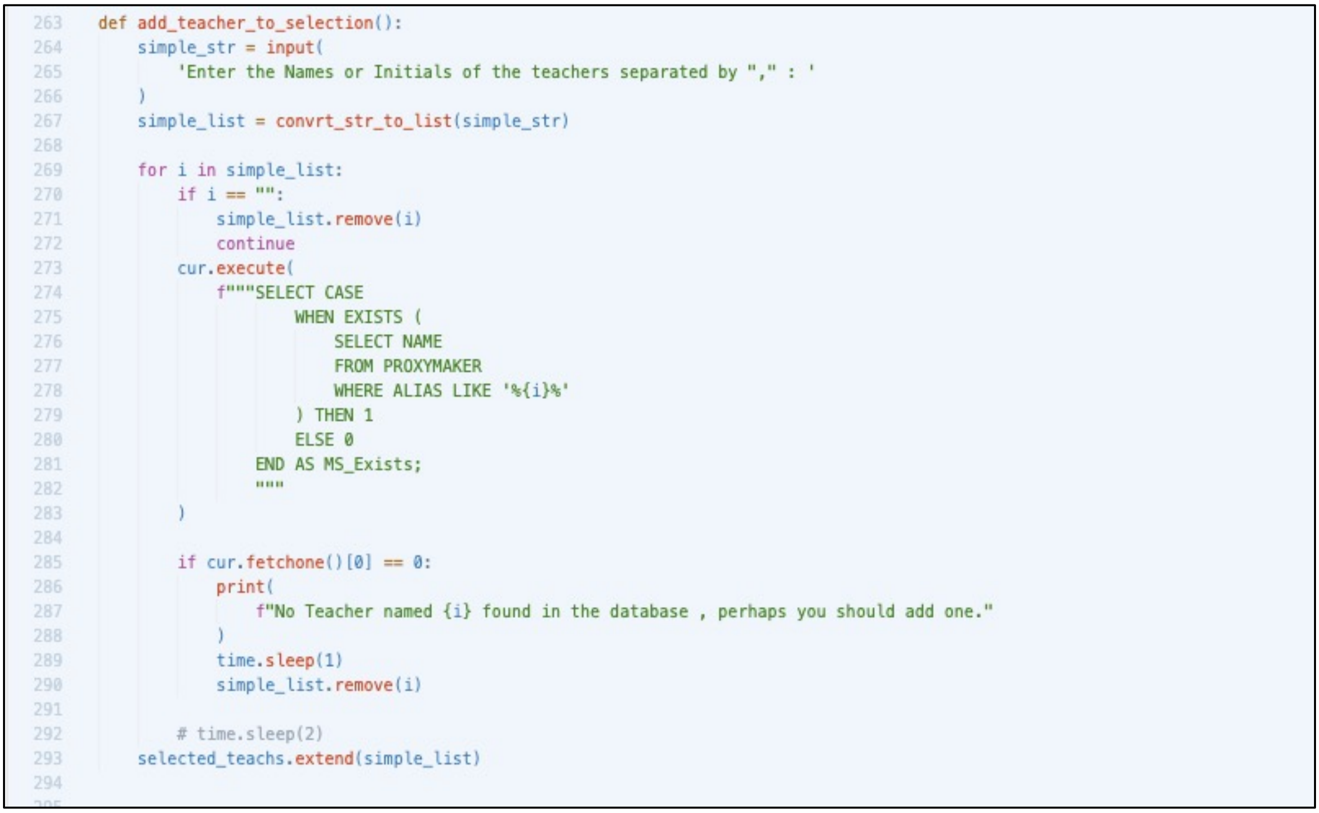
```python
def get_all_teachers():
    global cur
    global all_teachers_list
    cur.execute("SELECT NAME FROM PROXYMAKER")
    all_teachers_tuple = cur.fetchall()
    all_teachers_list = tuple_to_list2d(all_teachers_tuple)
    # print(all_teachers_list)    --debugging
    return all_teachers_list
```

#Divide-list-into-usable-rows-of-length(5):



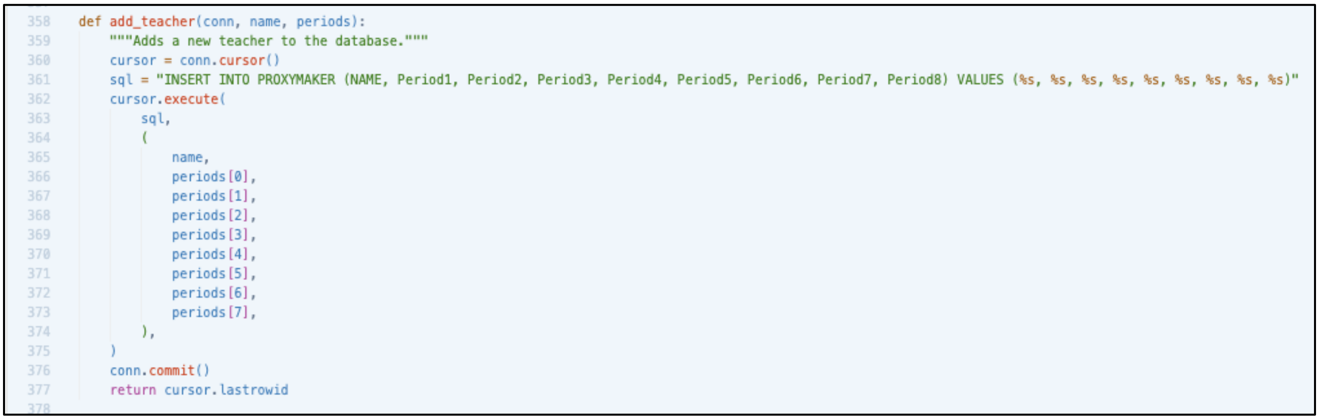
```python
def div_list(unprocessed_list):
    list_len = len(unprocessed_list)
    n = list_len // 5

    if list_len % 5 != 0:
        n += 1  # Increment n if there's a remainder

    processed_lists = []
    for i in range(0, n):
        processed_list = unprocessed_list[i * 5 : (i + 1) * 5]
        processed_lists.append(processed_list)

    return processed_lists
```

#Add-teacher-into-current-selection:



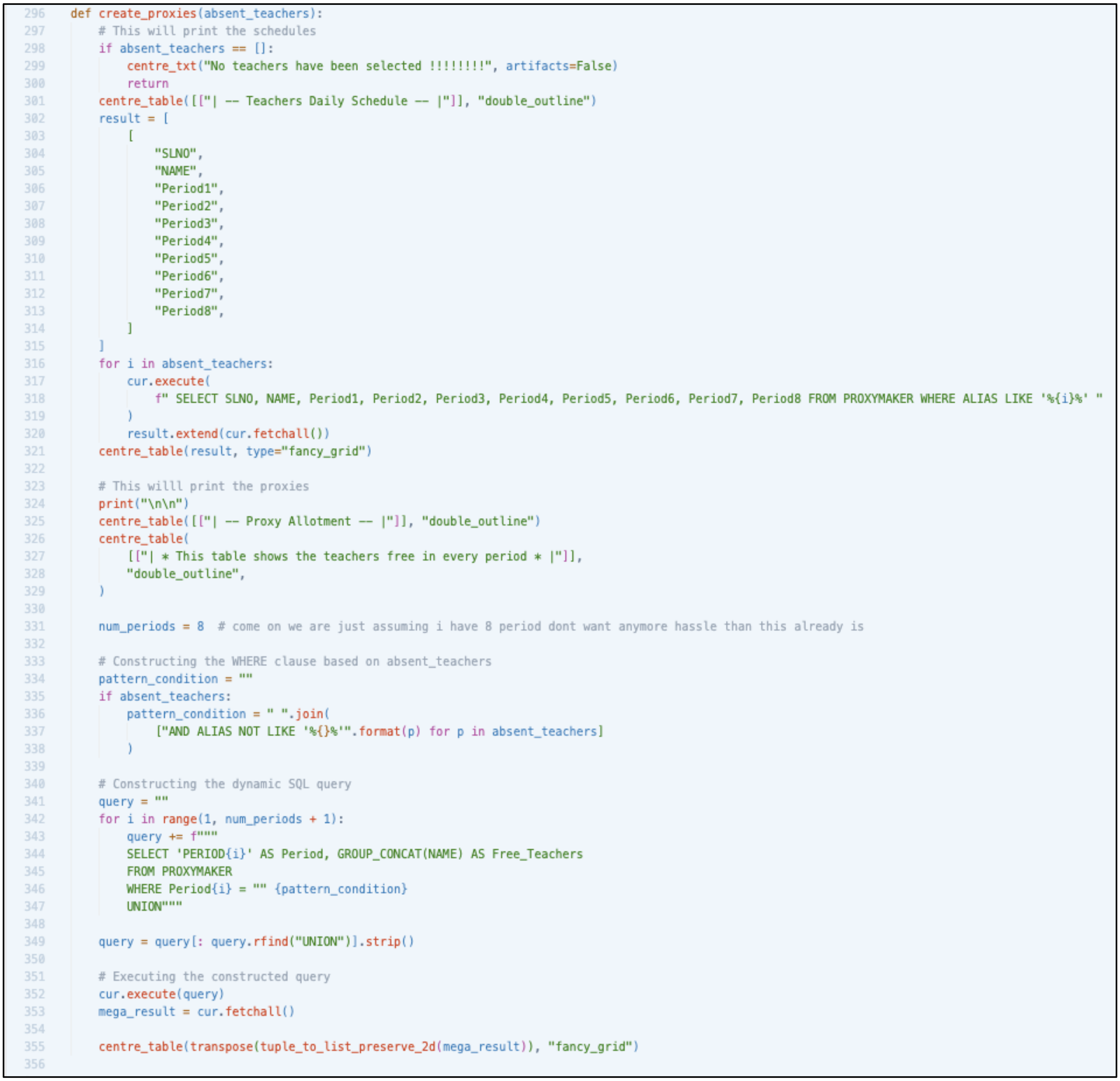
```python
def add_teacher_to_selection():
    simple_str = input(
        'Enter the Names or Initials of the teachers separated by "," : '
    )
    simple_list = convrt_str_to_list(simple_str)

    for i in simple_list:
        if i == "":
            simple_list.remove(i)
            continue
        cur.execute(
            f"""SELECT CASE
                    WHEN EXISTS (
                        SELECT NAME
                        FROM PROXYMAKER
                        WHERE ALIAS LIKE '%{i}%'
                    ) THEN 1
                    ELSE 0
                END AS MS_Exists;
                """
        )

        if cur.fetchone()[0] == 0:
            print(
                f"No Teacher named {i} found in the database , perhaps you should add one."
            )
            time.sleep(1)
            simple_list.remove(i)

        # time.sleep(2)
    selected_teachs.extend(simple_list)
```

#Adding-new-teacher-to-DB:



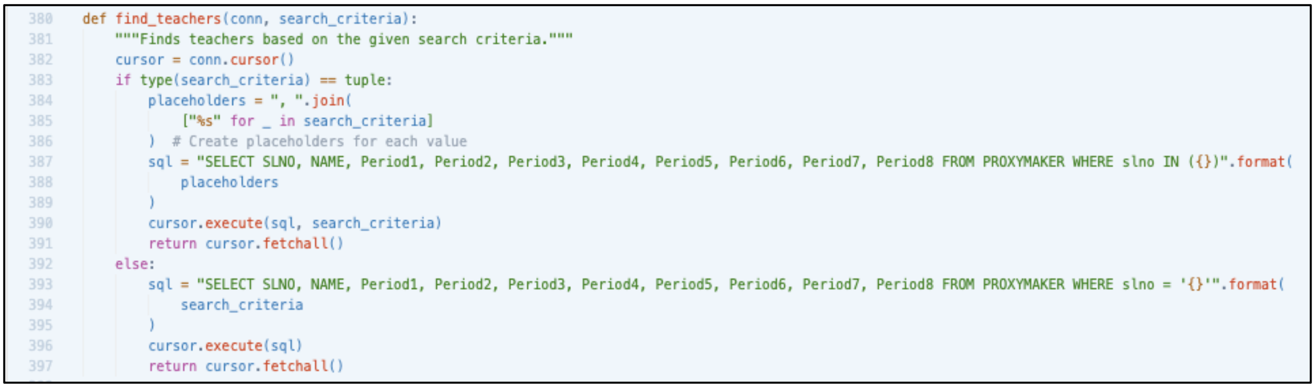
```python
def add_teacher(conn, name, periods):
    """Adds a new teacher to the database."""
    cursor = conn.cursor()
    sql = "INSERT INTO PROXYMAKER (NAME, Period1, Period2, Period3, Period4, Period5, Period6, Period7, Period8) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
    cursor.execute(
        sql,
        (
            name,
            periods[0],
            periods[1],
            periods[2],
            periods[3],
            periods[4],
            periods[5],
            periods[6],
            periods[7],
        ),
    )
    conn.commit()
    return cursor.lastrowid
```

#Generating-proxies:



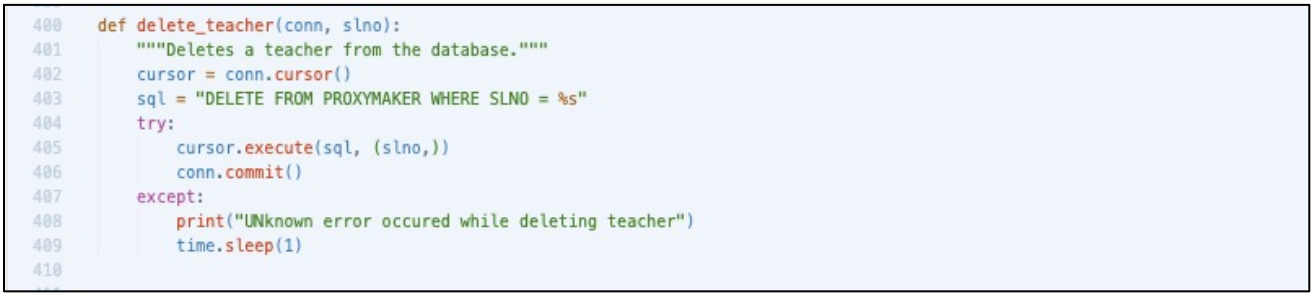
```python
def create_proxies(absent_teachers):
    # This will print the schedules
    if absent_teachers == []:
        centre_txt("No teachers have been selected !!!!!!!!", artifacts=False)
        return
    centre_table([["| -- Teachers Daily Schedule -- |"]], "double_outline")
    result = [
        [
            "SLNO",
            "NAME",
            "Period1",
            "Period2",
            "Period3",
            "Period4",
            "Period5",
            "Period6",
            "Period7",
            "Period8",
        ]
    ]
    for i in absent_teachers:
        cur.execute(
            f" SELECT SLNO, NAME, Period1, Period2, Period3, Period4, Period5, Period6, Period7, Period8 FROM PROXYMAKER WHERE ALIAS LIKE '%{i}%' "
        )
        result.extend(cur.fetchall())
    centre_table(result, type="fancy_grid")

    # This willl print the proxies
    print("\n\n")
    centre_table([["| -- Proxy Allotment -- |"]], "double_outline")
    centre_table(
        [["| * This table shows the teachers free in every period * |"]],
        "double_outline",
    )

    num_periods = 8  # come on we are just assuming i have 8 period dont want anymore hassle than this already is

    # Constructing the WHERE clause based on absent_teachers
    pattern_condition = ""
    if absent_teachers:
        pattern_condition = " ".join(
            ["AND ALIAS NOT LIKE '%{}%'".format(p) for p in absent_teachers]
        )

    # Constructing the dynamic SQL query
    query = ""
    for i in range(1, num_periods + 1):
        query += f"""
        SELECT 'PERIOD{i}' AS Period, GROUP_CONCAT(NAME) AS Free_Teachers
        FROM PROXYMAKER
        WHERE Period{i} = "" {pattern_condition}
        UNION"""

    query = query[: query.rfind("UNION")].strip()

    # Executing the constructed query
    cur.execute(query)
    mega_result = cur.fetchall()

    centre_table(transpose(tuple_to_list_preserve_2d(mega_result)), "fancy_grid")
```

#Finding-teacher-in-DB:



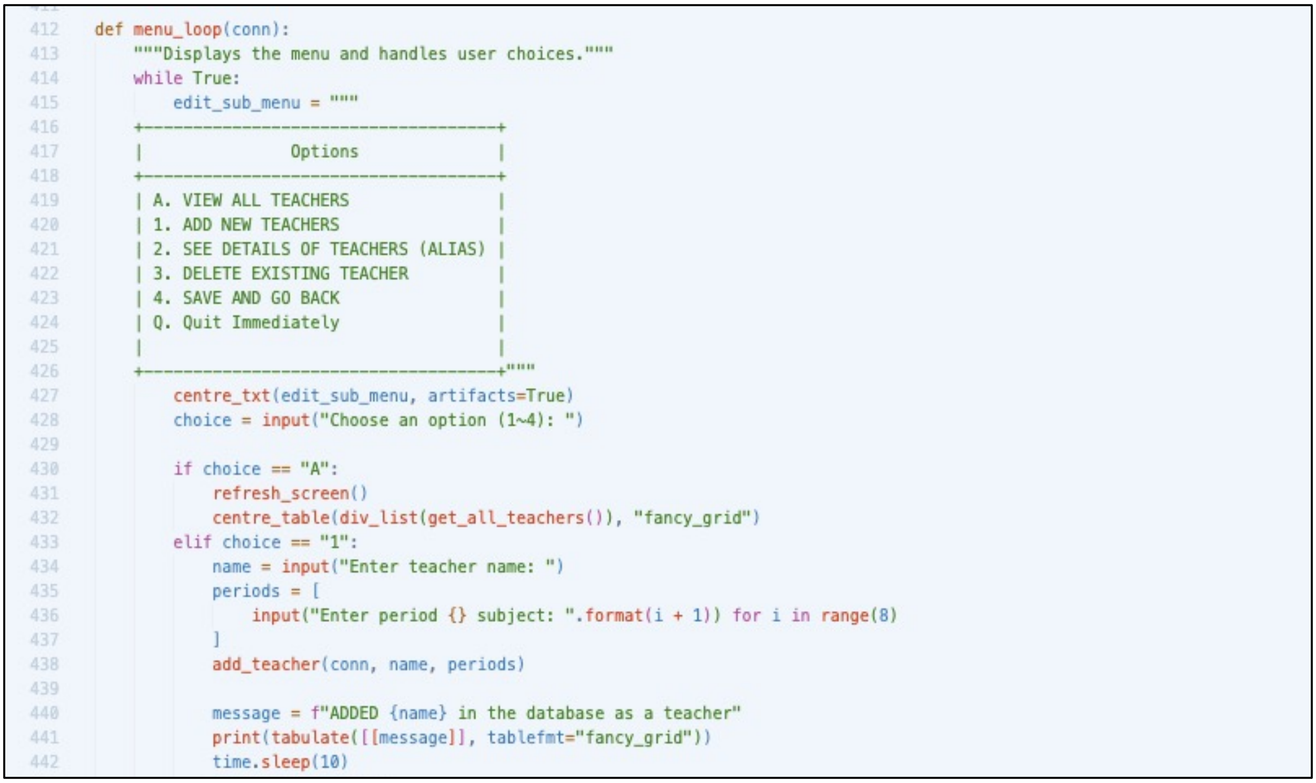
```python
def find_teachers(conn, search_criteria):
    """Finds teachers based on the given search criteria."""
    cursor = conn.cursor()
    if type(search_criteria) == tuple:
        placeholders = ", ".join(
            ["%s" for _ in search_criteria]
        )  # Create placeholders for each value
        sql = "SELECT SLNO, NAME, Period1, Period2, Period3, Period4, Period5, Period6, Period7, Period8 FROM PROXYMAKER WHERE slno IN ({})".format(
            placeholders
        )
        cursor.execute(sql, search_criteria)
        return cursor.fetchall()
    else:
        sql = "SELECT SLNO, NAME, Period1, Period2, Period3, Period4, Period5, Period6, Period7, Period8 FROM PROXYMAKER WHERE slno = '{}'".format(
            search_criteria
        )
        cursor.execute(sql)
        return cursor.fetchall()
```

#Deleting-teacher-in-DB:

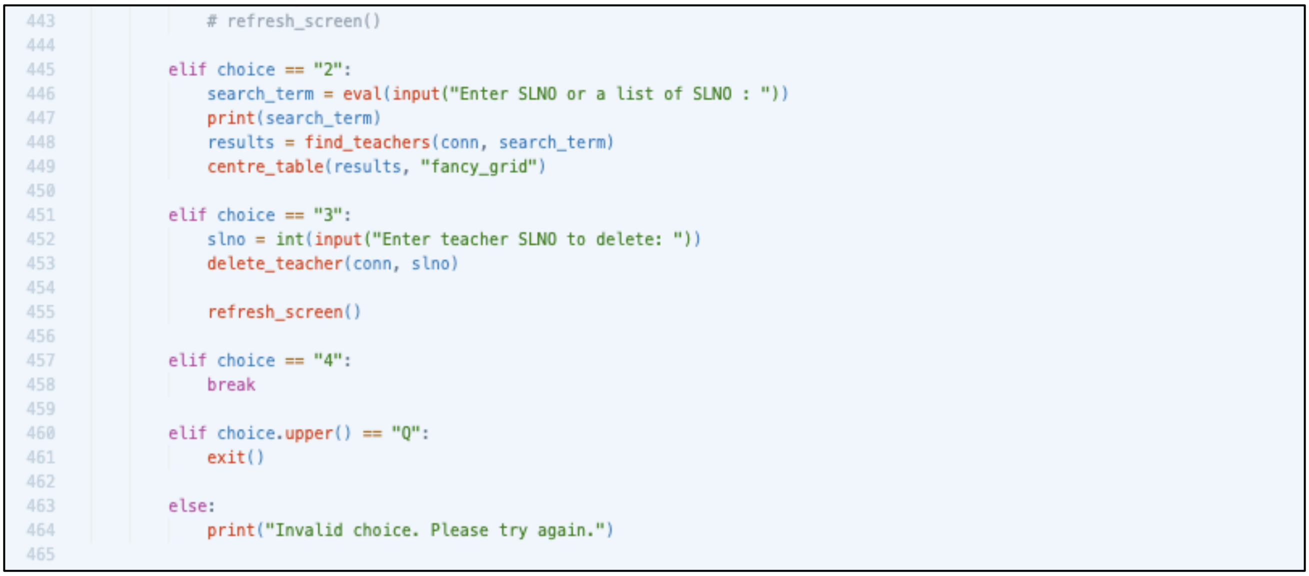


```python
def delete_teacher(conn, slno):
    """Deletes a teacher from the database."""
    cursor = conn.cursor()
    sql = "DELETE FROM PROXYMAKER WHERE SLNO = %s"
    try:
        cursor.execute(sql, (slno,))
        conn.commit()
    except:
        print("UNknown error occured while deleting teacher")
        time.sleep(1)

```

#DB-edit-menu-loop:

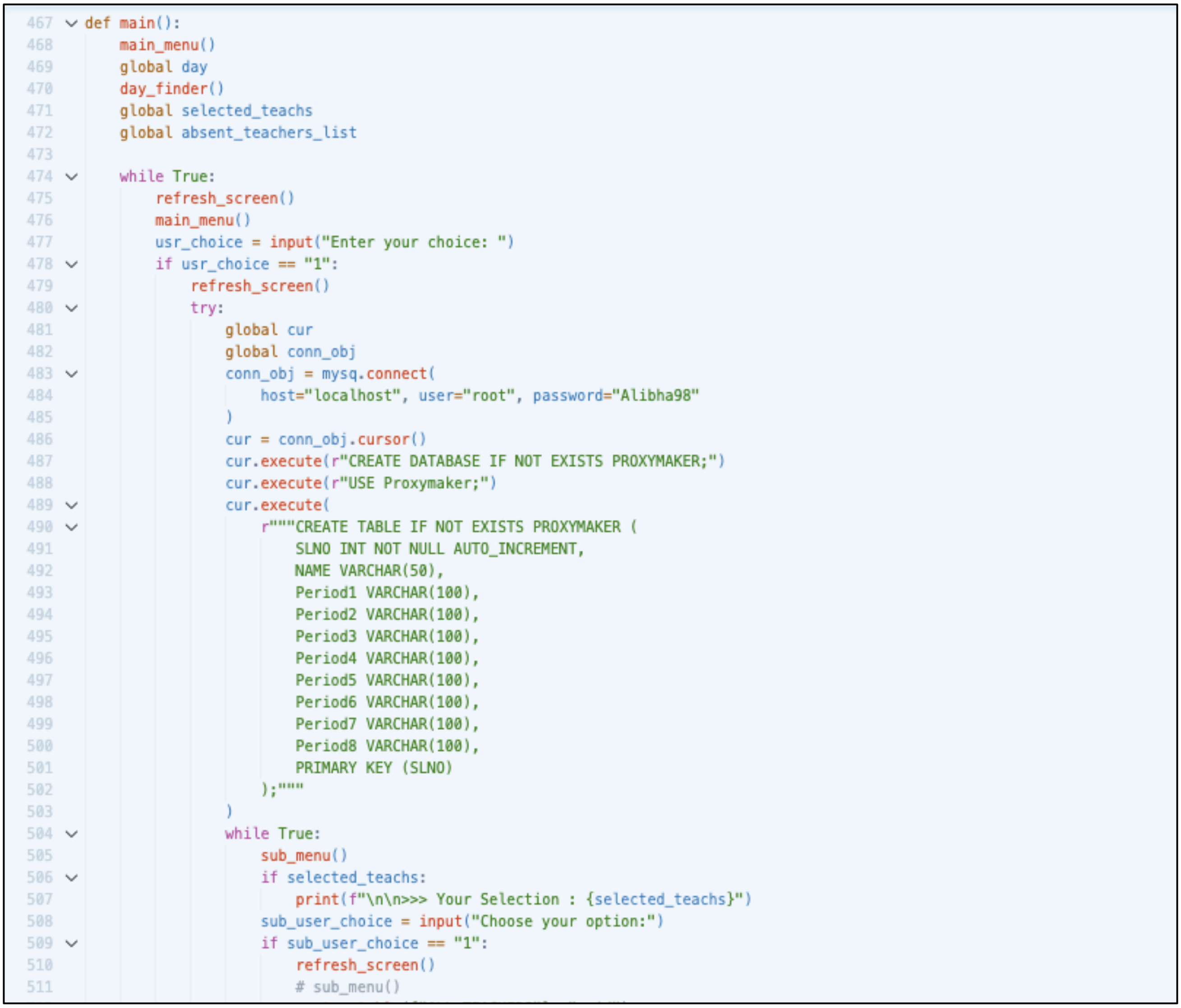


```python
def menu_loop(conn):
    """Displays the menu and handles user choices."""
    while True:
        edit_sub_menu = """
    +------------------------------------+
    |               Options              |
    +------------------------------------+
    | A. VIEW ALL TEACHERS               |
    | 1. ADD NEW TEACHERS                |
    | 2. SEE DETAILS OF TEACHERS (ALIAS) |
    | 3. DELETE EXISTING TEACHER         |
    | 4. SAVE AND GO BACK                |
    | Q. Quit Immediately                |
    |                                    |
    +------------------------------------+"""
        centre_txt(edit_sub_menu, artifacts=True)
        choice = input("Choose an option (1~4): ")

        if choice == "A":
            refresh_screen()
            centre_table(div_list(get_all_teachers()), "fancy_grid")
        elif choice == "1":
            name = input("Enter teacher name: ")
            periods = [
                input("Enter period {} subject: ".format(i + 1)) for i in range(8)
            ]
            add_teacher(conn, name, periods)

            message = f"ADDED {name} in the database as a teacher"
            print(tabulate([[message]], tablefmt="fancy_grid"))
            time.sleep(10)
```

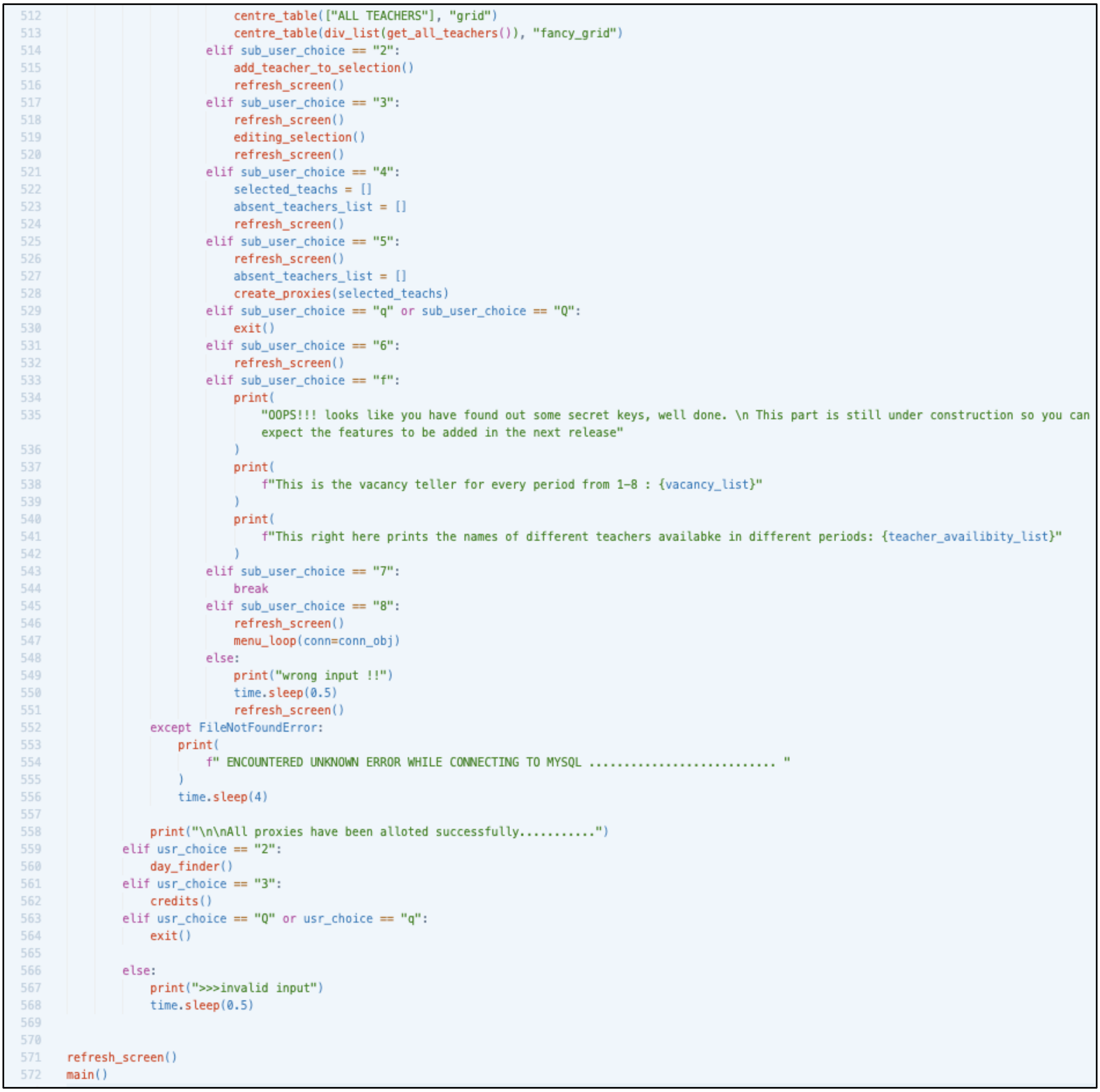


```python
            # refresh_screen()

        elif choice == "2":
            search_term = eval(input("Enter SLNO or a list of SLNO : "))
            print(search_term)
            results = find_teachers(conn, search_term)
            centre_table(results, "fancy_grid")

        elif choice == "3":
            slno = int(input("Enter teacher SLNO to delete: "))
            delete_teacher(conn, slno)

            refresh_screen()

        elif choice == "4":
            break

        elif choice.upper() == "Q":
            exit()

        else:
            print("Invalid choice. Please try again.")

```

#Main-loop:



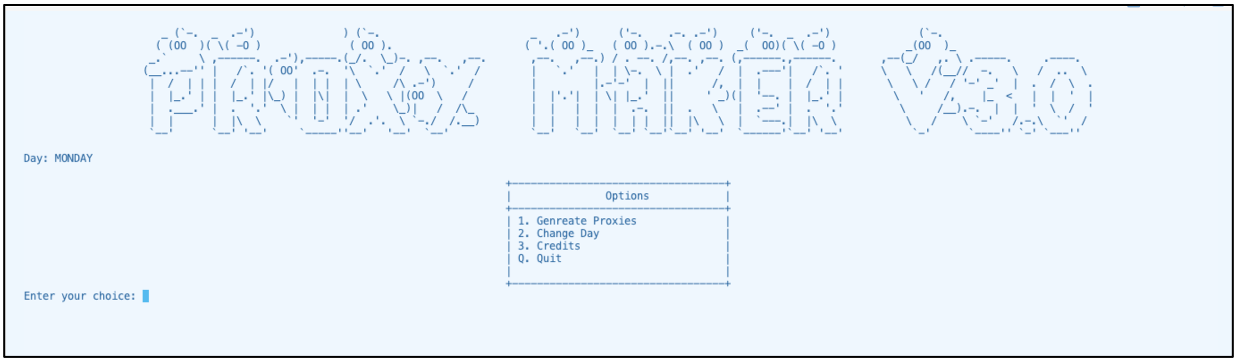
```python
def main():
    main_menu()
    global day
    day_finder()
    global selected_teachs
    global absent_teachers_list

    while True:
        refresh_screen()
        main_menu()
        usr_choice = input("Enter your choice: ")
        if usr_choice == "1":
            refresh_screen()
            try:
                global cur
                global conn_obj
                conn_obj = mysq.connect(
                    host="localhost", user="root", password="Alibha98"
                )
                cur = conn_obj.cursor()
                cur.execute(r"CREATE DATABASE IF NOT EXISTS PROXYMAKER;")
                cur.execute(r"USE Proxymaker;")
                cur.execute(
                    r"""CREATE TABLE IF NOT EXISTS PROXYMAKER (
                        SLNO INT NOT NULL AUTO_INCREMENT,
                        NAME VARCHAR(50),
                        Period1 VARCHAR(100),
                        Period2 VARCHAR(100),
                        Period3 VARCHAR(100),
                        Period4 VARCHAR(100),
                        Period5 VARCHAR(100),
                        Period6 VARCHAR(100),
                        Period7 VARCHAR(100),
                        Period8 VARCHAR(100),
                        PRIMARY KEY (SLNO)
                    );"""
                )
                while True:
                    sub_menu()
                    if selected_teachs:
                        print(f"\n\n>> Your Selection : {selected_teachs}")
                    sub_user_choice = input("Choose your option:")
                    if sub_user_choice == "1":
                        refresh_screen()
                        # sub_menu()
```



```
                        centre_table(["ALL TEACHERS"], "grid")
                        centre_table(div_list(get_all_teachers()), "fancy_grid")
                    elif sub_user_choice == "2":
                        add_teacher_to_selection()
                        refresh_screen()
                    elif sub_user_choice == "3":
                        refresh_screen()
                        editing_selection()
                        refresh_screen()
                    elif sub_user_choice == "4":
                        selected_teachs = []
                        absent_teachers_list = []
                        refresh_screen()
                    elif sub_user_choice == "5":
                        refresh_screen()
                        absent_teachers_list = []
                        create_proxies(selected_teachs)
                    elif sub_user_choice == "q" or sub_user_choice == "Q":
                        exit()
                    elif sub_user_choice == "6":
                        refresh_screen()
                    elif sub_user_choice == "f":
                        print(
                            "OOPS!!! looks like you have found out some secret keys, well done. \n This part is still under construction so you can expect the features to be added in the next release"
                        )
                        print(
                            f"This is the vacancy teller for every period from 1-8 : {vacancy_list}"
                        )
                        print(
                            f"This right here prints the names of different teachers availabke in different periods: {teacher_availibity_list}"
                        )
                    elif sub_user_choice == "7":
                        break
                    elif sub_user_choice == "8":
                        refresh_screen()
                        menu_loop(conn=conn_obj)
                    else:
                        print("wrong input !!")
                        time.sleep(0.5)
                        refresh_screen()
            except FileNotFoundError:
                print(
                    f" ENCOUNTERED UNKNOWN ERROR WHILE CONNECTING TO MYSQL ........................... "
                )
                time.sleep(4)

            print("\n\nAll proxies have been alloted successfully...........")
        elif usr_choice == "2":
            day_finder()
        elif usr_choice == "3":
            credits()
        elif usr_choice == "Q" or usr_choice == "q":
            exit()

        else:
            print(">>>invalid input")
            time.sleep(0.5)


refresh_screen()
main()
```

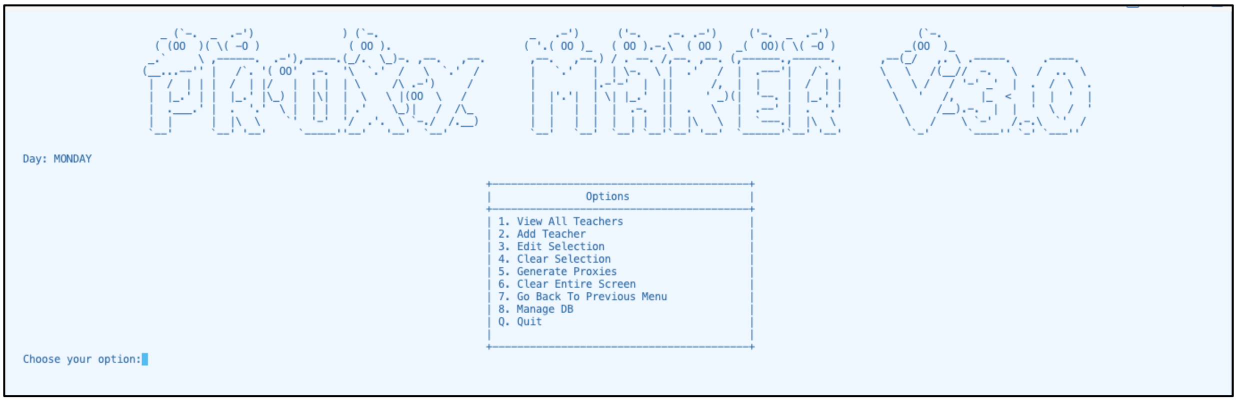
Test Cases

#Main-Menu:



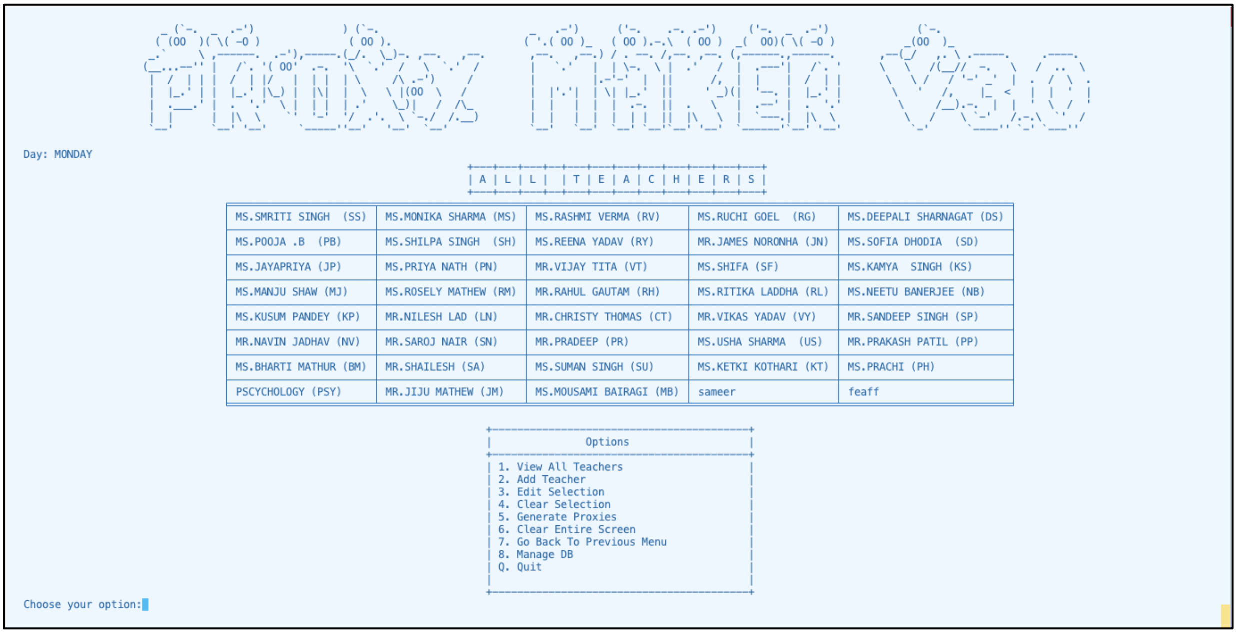
```
Day: MONDAY

                        +--------------------------------+
                        |            Options             |
                        +--------------------------------+
                        | 1. Genreate Proxies            |
                        | 2. Change Day                  |
                        | 3. Credits                     |
                        | Q. Quit                        |
                        |                                |
                        +--------------------------------+

Enter your choice: 
```

#Sub-Menu:



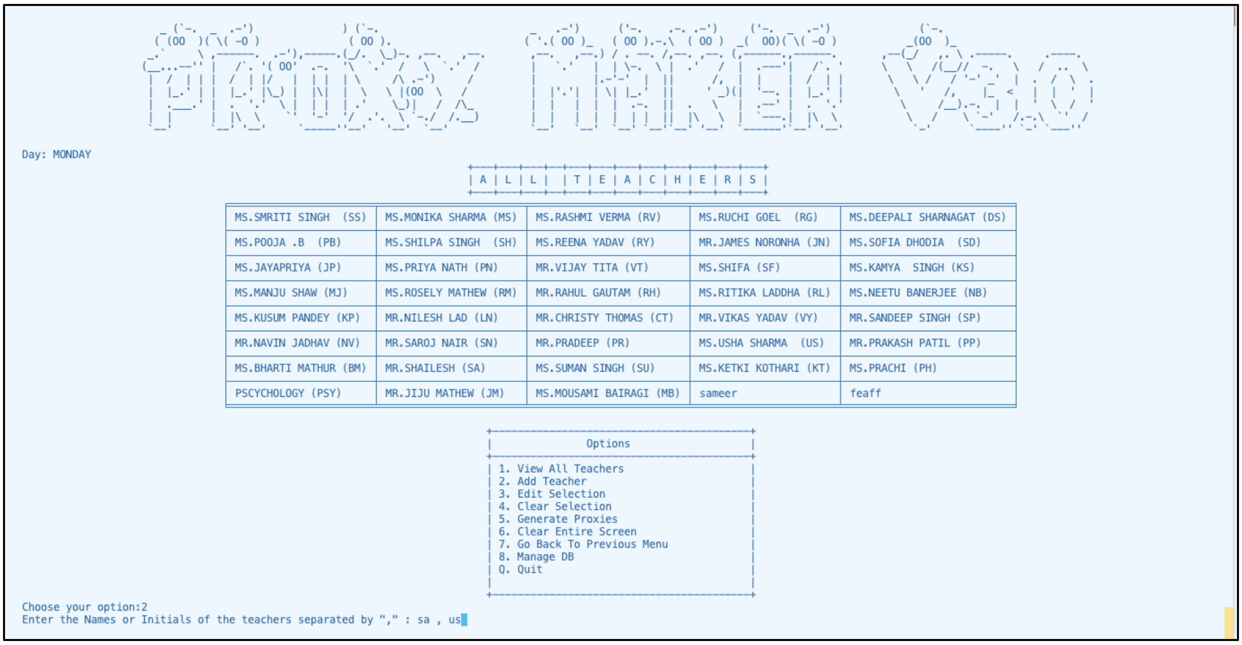
```
Day: MONDAY

                     +--------------------------------------+
                     |               Options                |
                     +--------------------------------------+
                     | 1. View All Teachers                 |
                     | 2. Add Teacher                       |
                     | 3. Edit Selection                    |
                     | 4. Clear Selection                   |
                     | 5. Generate Proxies                  |
                     | 6. Clear Entire Screen               |
                     | 7. Go Back To Previous Menu          |
                     | 8. Manage DB                         |
                     | Q. Quit                              |
                     |                                      |
                     +--------------------------------------+

Choose your option:
```

#get-all-teachers:



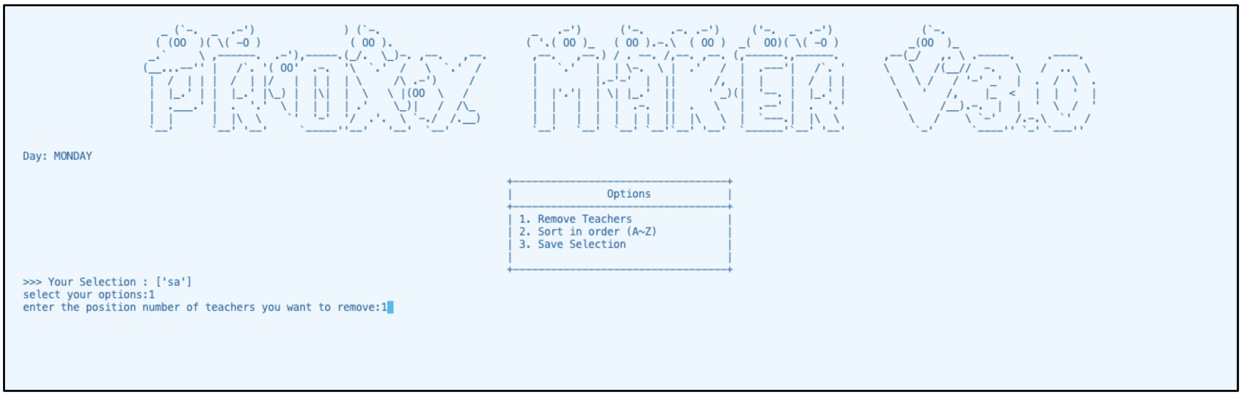
```
Day: MONDAY

                     +---+---+---+---+---+---+---+---+---+---+---+---+
                     | A | L | L |   | T | E | A | C | H | E | R | S |
                     +---+---+---+---+---+---+---+---+---+---+---+---+

| MS.SMRITI SINGH  (SS) | MS.MONIKA SHARMA (MS) | MS.RASHMI VERMA (RV)    | MS.RUCHI GOEL  (RG)   | MS.DEEPALI SHARNAGAT (DS) |
| MS.POOJA .B  (PB)     | MS.SHILPA SINGH  (SH) | MS.REENA YADAV (RY)     | MR.JAMES NORONHA (JN) | MS.SOFIA DHODIA  (SD)     |
| MS.JAYAPRIYA (JP)     | MS.PRIYA NATH (PN)    | MR.VIJAY TITA (VT)      | MS.SHIFA (SF)         | MS.KAMYA  SINGH (KS)      |
| MS.MANJU SHAW (MJ)    | MS.ROSELY MATHEW (RM) | MR.RAHUL GAUTAM (RH)    | MS.RITIKA LADDHA (RL) | MS.NEETU BANERJEE (NB)    |
| MS.KUSUM PANDEY (KP)  | MR.NILESH LAD (LN)    | MR.CHRISTY THOMAS (CT)  | MR.VIKAS YADAV (VY)   | MR.SANDEEP SINGH (SP)     |
| MR.NAVIN JADHAV (NV)  | MR.SAROJ NAIR (SN)    | MR.PRADEEP (PR)         | MS.USHA SHARMA  (US)  | MR.PRAKASH PATIL (PP)     |
| MS.BHARTI MATHUR (BM) | MR.SHAILESH (SA)      | MS.SUMAN SINGH (SU)     | MS.KETKI KOTHARI (KT) | MS.PRACHI (PH)            |
| PSCYCHOLOGY (PSY)     | MR.JIJU MATHEW (JM)   | MS.MOUSAMI BAIRAGI (MB) | sameer                | feaff                     |

                     +--------------------------------------+
                     |               Options                |
                     +--------------------------------------+
                     | 1. View All Teachers                 |
                     | 2. Add Teacher                       |
                     | 3. Edit Selection                    |
                     | 4. Clear Selection                   |
                     | 5. Generate Proxies                  |
                     | 6. Clear Entire Screen               |
                     | 7. Go Back To Previous Menu          |
                     | 8. Manage DB                         |
                     | Q. Quit                              |
                     |                                      |
                     +--------------------------------------+

Choose your option:
```

#Adding-Teachers-to-selection:



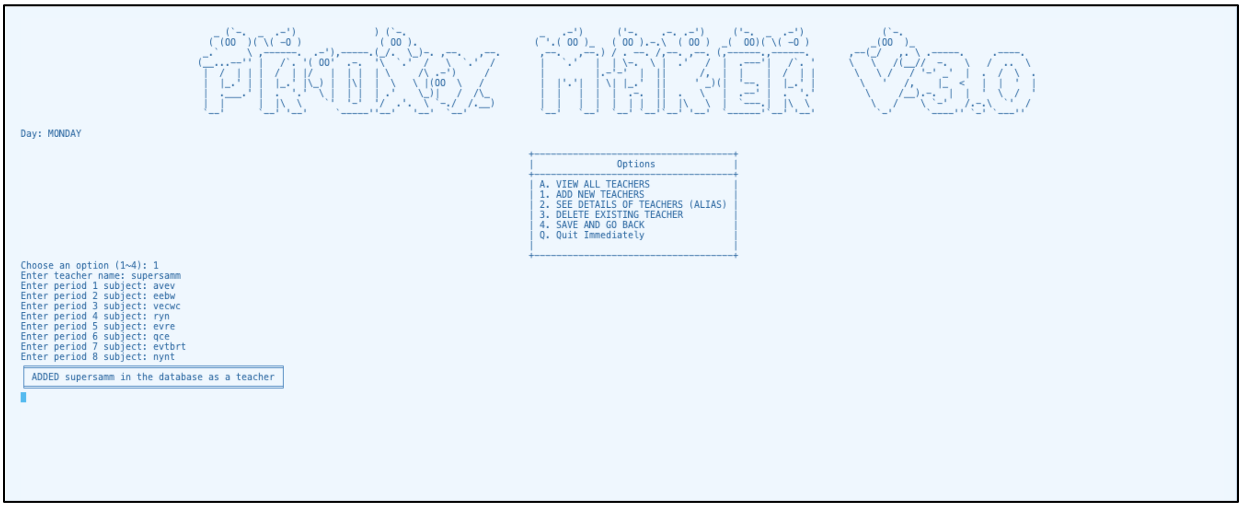
```
Day: MONDAY

                    +---+---+---+---+---+---+---+---+---+---+---+---+
                    | A | L | L |   | T | E | A | C | H | E | R | S |
                    +---+---+---+---+---+---+---+---+---+---+---+---+

| MS.SMRITI SINGH  (SS) | MS.MONIKA SHARMA (MS) | MS.RASHMI VERMA (RV)     | MS.RUCHI GOEL  (RG)    | MS.DEEPALI SHARNAGAT (DS) |
| MS.POOJA .B  (PB)     | MS.SHILPA SINGH  (SH) | MS.REENA YADAV (RY)      | MR.JAMES NORONHA (JN)  | MS.SOFIA DHODIA  (SD)     |
| MS.JAYAPRIYA (JP)     | MS.PRIYA NATH (PN)    | MR.VIJAY TITA (VT)       | MS.SHIFA (SF)          | MS.KAMYA  SINGH (KS)      |
| MS.MANJU SHAW (MJ)    | MS.ROSELY MATHEW (RM) | MR.RAHUL GAUTAM (RH)     | MS.RITIKA LADDHA (RL)  | MS.NEETU BANERJEE (NB)    |
| MS.KUSUM PANDEY (KP)  | MR.NILESH LAD (LN)    | MR.CHRISTY THOMAS (CT)   | MR.VIKAS YADAV (VY)    | MR.SANDEEP SINGH (SP)     |
| MR.NAVIN JADHAV (NV)  | MR.SAROJ NAIR (SN)    | MR.PRADEEP (PR)          | MS.USHA SHARMA  (US)   | MR.PRAKASH PATIL (PP)     |
| MS.BHARTI MATHUR (BM) | MR.SHAILESH (SA)      | MS.SUMAN SINGH (SU)      | MS.KETKI KOTHARI (KT)  | MS.PRACHI (PH)            |
| PSCYCHOLOGY (PSY)     | MR.JIJU MATHEW (JM)   | MS.MOUSAMI BAIRAGI (MB)  | sameer                 | feaff                     |

                    +-----------------------------------------+
                    |                 Options                 |
                    +-----------------------------------------+
                    | 1. View All Teachers                    |
                    | 2. Add Teacher                          |
                    | 3. Edit Selection                       |
                    | 4. Clear Selection                      |
                    | 5. Generate Proxies                     |
                    | 6. Clear Entire Screen                  |
                    | 7. Go Back To Previous Menu             |
                    | 8. Manage DB                            |
                    | Q. Quit                                 |
                    |                                         |
                    +-----------------------------------------+

Choose your option:2
Enter the Names or Initials of the teachers separated by "," : sa , us
```

#Editing-selected-teachers:



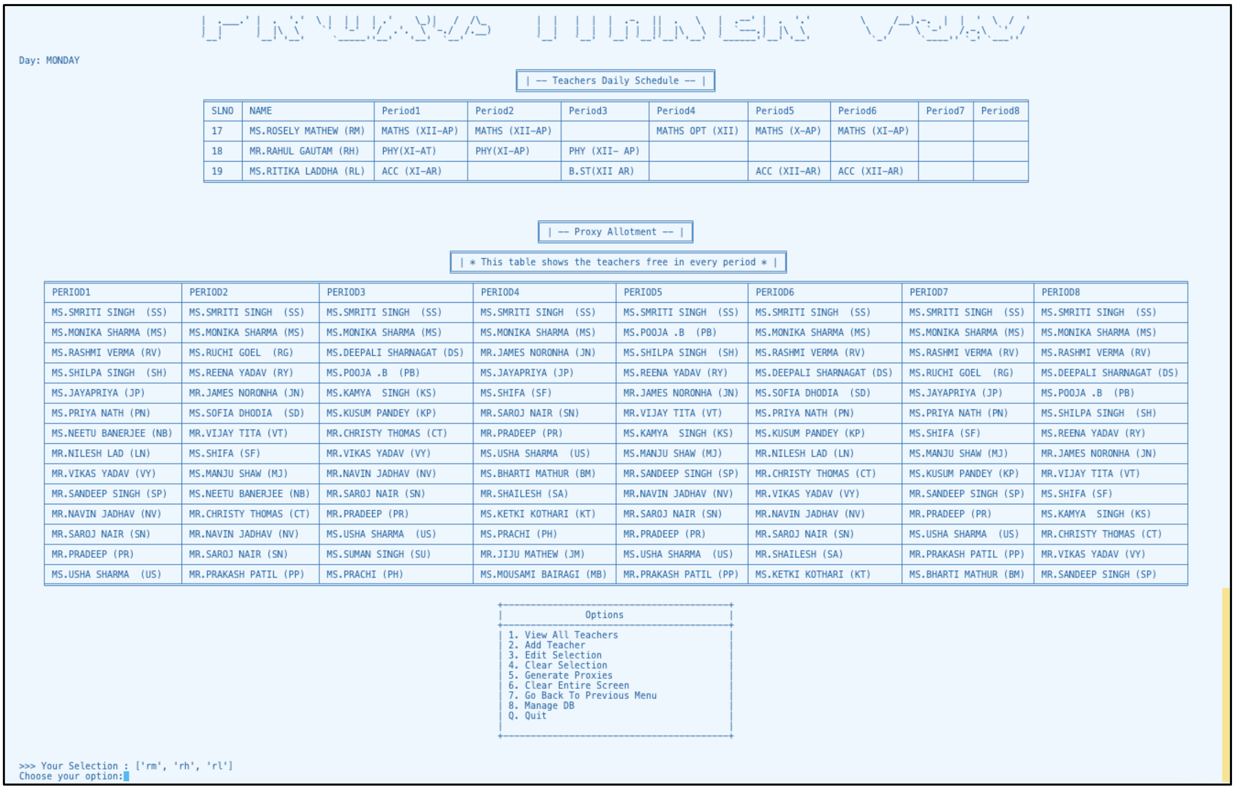
```
Day: MONDAY

                    +----------------------------------+
                    |             Options              |
                    +----------------------------------+
                    | 1. Remove Teachers               |
                    | 2. Sort in order (A~Z)           |
                    | 3. Save Selection                |
                    |                                  |
                    +----------------------------------+

>>> Your Selection : ['sa']
select your options:1
enter the position number of teachers you want to remove:1
```

#Adding-new-teachers-to-DB:



```
Day: MONDAY

                    +---------------------------------------+
                    |                Options                |
                    +---------------------------------------+
                    | A. VIEW ALL TEACHERS                  |
                    | 1. ADD NEW TEACHERS                   |
                    | 2. SEE DETAILS OF TEACHERS (ALIAS)    |
                    | 3. DELETE EXISTING TEACHER            |
                    | 4. SAVE AND GO BACK                   |
                    | Q. Quit Immediately                   |
                    |                                       |
                    +---------------------------------------+

Choose an option (1~4): 1
Enter teacher name: supersamm
Enter period 1 subject: avev
Enter period 2 subject: eebw
Enter period 3 subject: vecwc
Enter period 4 subject: ryn
Enter period 5 subject: evre
Enter period 6 subject: qce
Enter period 7 subject: evtbrt
Enter period 8 subject: nynt
 +-----------------------------------------+
 | ADDED supersamm in the database as a teacher |
 +-----------------------------------------+
```

#Generating-proxies:



Day: MONDAY

| -- Teachers Daily Schedule -- |

SLNO	NAME	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8
17	MS.ROSELY MATHEW (RM)	MATHS (XII-AP)	MATHS (XII-AP)		MATHS OPT (XII)	MATHS (X-AP)	MATHS (XI-AP)		
18	MR.RAHUL GAUTAM (RH)	PHY(XI-AT)	PHY(XI-AP)	PHY (XII- AP)					
19	MS.RITIKA LADDHA (RL)	ACC (XI-AR)		B.ST(XII AR)		ACC (XII-AR)	ACC (XII-AR)		

| -- Proxy Allotment -- |

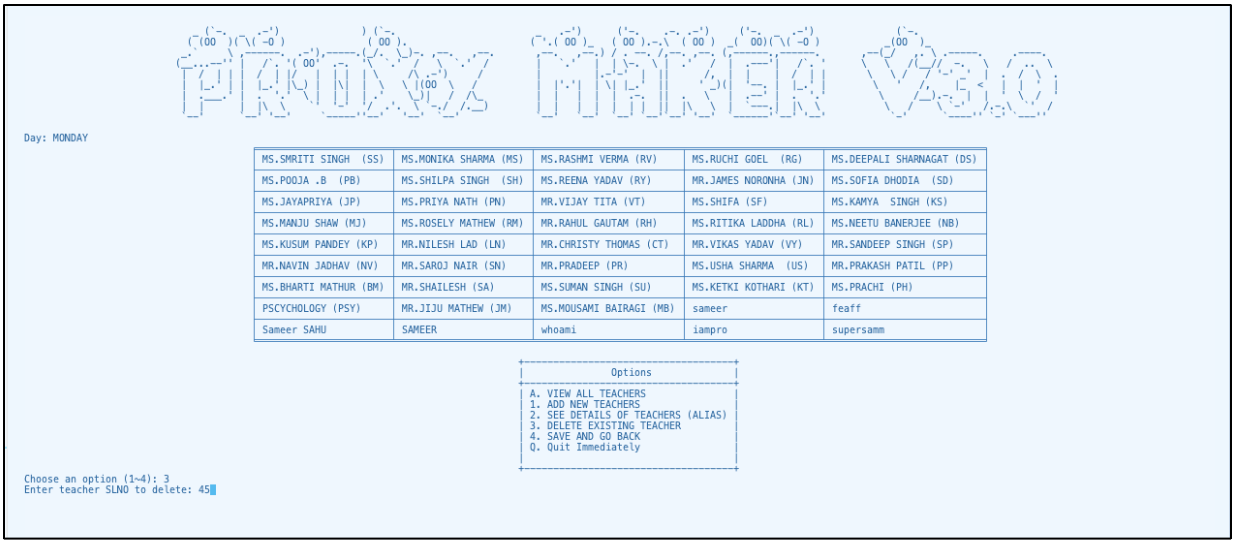
| * This table shows the teachers free in every period * |

PERIOD1	PERIOD2	PERIOD3	PERIOD4	PERIOD5	PERIOD6	PERIOD7	PERIOD8
MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)	MS.SMRITI SINGH (SS)
MS.MONIKA SHARMA (MS)	MS.MONIKA SHARMA (MS)	MS.MONIKA SHARMA (MS)	MS.MONIKA SHARMA (MS)	MS.POOJA .B (PB)	MS.MONIKA SHARMA (MS)	MS.MONIKA SHARMA (MS)	MS.MONIKA SHARMA (MS)
MS.RASHMI VERMA (RV)	MS.RUCHI GOEL (RG)	MS.DEEPALI SHARNAGAT (DS)	MR.JAMES NORONHA (JN)	MS.SHILPA SINGH (SH)	MS.RASHMI VERMA (RV)	MS.RASHMI VERMA (RV)	MS.RASHMI VERMA (RV)
MS.SHILPA SINGH (SH)	MS.REENA YADAV (RY)	MS.POOJA .B (PB)	MS.JAYAPRIYA (JP)	MS.REENA YADAV (RY)	MS.DEEPALI SHARNAGAT (DS)	MS.RUCHI GOEL (RG)	MS.DEEPALI SHARNAGAT (DS)
MS.JAYAPRIYA (JP)	MR.JAMES NORONHA (JN)	MS.KAMYA SINGH (KS)	MS.SHIFA (SF)	MR.JAMES NORONHA (JN)	MS.SOFIA DHODIA (SD)	MS.JAYAPRIYA (JP)	MS.POOJA .B (PB)
MS.PRIYA NATH (PN)	MS.SOFIA DHODIA (SD)	MS.KUSUM PANDEY (KP)	MR.SAROJ NAIR (SN)	MR.VIJAY TITA (VT)	MS.PRIYA NATH (PN)	MS.PRIYA NATH (PN)	MS.SHILPA SINGH (SH)
MS.NEETU BANERJEE (NB)	MR.VIJAY TITA (VT)	MR.CHRISTY THOMAS (CT)	MR.PRADEEP (PR)	MS.KAMYA SINGH (KS)	MS.KUSUM PANDEY (KP)	MS.SHIFA (SF)	MS.REENA YADAV (RY)
MR.NILESH LAD (LN)	MS.SHIFA (SF)	MR.VIKAS YADAV (VY)	MS.USHA SHARMA (US)	MS.MANJU SHAW (MJ)	MR.NILESH LAD (LN)	MS.MANJU SHAW (MJ)	MR.JAMES NORONHA (JN)
MR.VIKAS YADAV (VY)	MS.MANJU SHAW (MJ)	MR.NAVIN JADHAV (NV)	MS.BHARTI MATHUR (BM)	MR.SANDEEP SINGH (SP)	MR.CHRISTY THOMAS (CT)	MS.KUSUM PANDEY (KP)	MR.VIJAY TITA (VT)
MR.SANDEEP SINGH (SP)	MS.NEETU BANERJEE (NB)	MR.SAROJ NAIR (SN)	MR.SHAILESH (SA)	MR.NAVIN JADHAV (NV)	MR.VIKAS YADAV (VY)	MR.SANDEEP SINGH (SP)	MS.SHIFA (SF)
MR.NAVIN JADHAV (NV)	MR.CHRISTY THOMAS (CT)	MR.PRADEEP (PR)	MS.KETKI KOTHARI (KT)	MR.SAROJ NAIR (SN)	MR.NAVIN JADHAV (NV)	MR.PRADEEP (PR)	MS.KAMYA SINGH (KS)
MR.SAROJ NAIR (SN)	MR.NAVIN JADHAV (NV)	MS.USHA SHARMA (US)	MS.PRACHI (PH)	MR.PRADEEP (PR)	MR.SAROJ NAIR (SN)	MS.USHA SHARMA (US)	MR.CHRISTY THOMAS (CT)
MR.PRADEEP (PR)	MR.SAROJ NAIR (SN)	MS.SUMAN SINGH (SU)	MR.JIJU MATHEW (JM)	MS.USHA SHARMA (US)	MR.SHAILESH (SA)	MR.PRAKASH PATIL (PP)	MR.VIKAS YADAV (VY)
MS.USHA SHARMA (US)	MR.PRAKASH PATIL (PP)	MS.PRACHI (PH)	MS.MOUSAMI BAIRAGI (MB)	MR.PRAKASH PATIL (PP)	MS.KETKI KOTHARI (KT)	MS.BHARTI MATHUR (BM)	MR.SANDEEP SINGH (SP)

```
+----------------------------------+
|              Options             |
+----------------------------------+
| 1. View All Teachers             |
| 2. Add Teacher                   |
| 3. Edit Selection                |
| 4. Clear Selection               |
| 5. Generate Proxies              |
| 6. Clear Entire Screen           |
| 7. Go Back To Previous Menu      |
| 8. Manage DB                     |
| Q. Quit                          |
|                                  |
+----------------------------------+
```

```
>>> Your Selection : ['rm', 'rh', 'rl']
Choose your option:
```

#Deleting-teachers-in-DB:

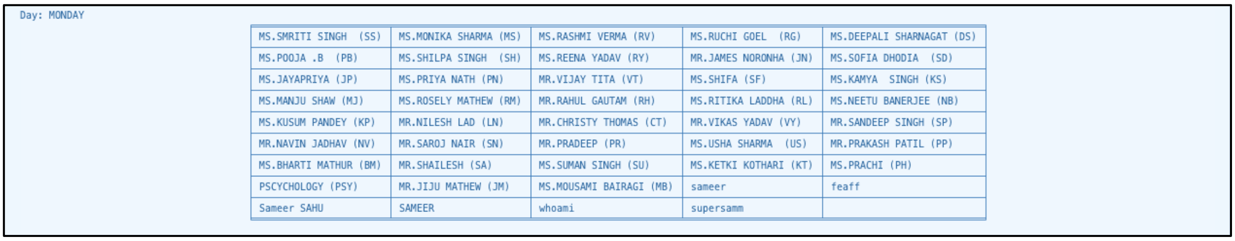


Day: MONDAY

MS.SMRITI SINGH (SS)	MS.MONIKA SHARMA (MS)	MS.RASHMI VERMA (RV)	MS.RUCHI GOEL (RG)	MS.DEEPALI SHARNAGAT (DS)
MS.POOJA .B (PB)	MS.SHILPA SINGH (SH)	MS.REENA YADAV (RY)	MR.JAMES NORONHA (JN)	MS.SOFIA DHODIA (SD)
MS.JAYAPRIYA (JP)	MS.PRIYA NATH (PN)	MR.VIJAY TITA (VT)	MS.SHIFA (SF)	MS.KAMYA SINGH (KS)
MS.MANJU SHAW (MJ)	MS.ROSELY MATHEW (RM)	MR.RAHUL GAUTAM (RH)	MS.RITIKA LADDHA (RL)	MS.NEETU BANERJEE (NB)
MS.KUSUM PANDEY (KP)	MR.NILESH LAD (LN)	MR.CHRISTY THOMAS (CT)	MR.VIKAS YADAV (VY)	MR.SANDEEP SINGH (SP)
MR.NAVIN JADHAV (NV)	MR.SAROJ NAIR (SN)	MR.PRADEEP (PR)	MS.USHA SHARMA (US)	MR.PRAKASH PATIL (PP)
MS.BHARTI MATHUR (BM)	MR.SHAILESH (SA)	MS.SUMAN SINGH (SU)	MS.KETKI KOTHARI (KT)	MS.PRACHI (PH)
PSCYCHOLOGY (PSY)	MR.JIJU MATHEW (JM)	MS.MOUSAMI BAIRAGI (MB)	sameer	feaff
Sameer SAHU	SAMEER	whoami	iampro	supersamm

```
+--------------------------------------+
|                Options               |
+--------------------------------------+
| A. VIEW ALL TEACHERS                 |
| 1. ADD NEW TEACHERS                  |
| 2. SEE DETAILS OF TEACHERS (ALIAS)   |
| 3. DELETE EXISTING TEACHER           |
| 4. SAVE AND GO BACK                  |
| Q. Quit Immediately                  |
|                                      |
+--------------------------------------+
```

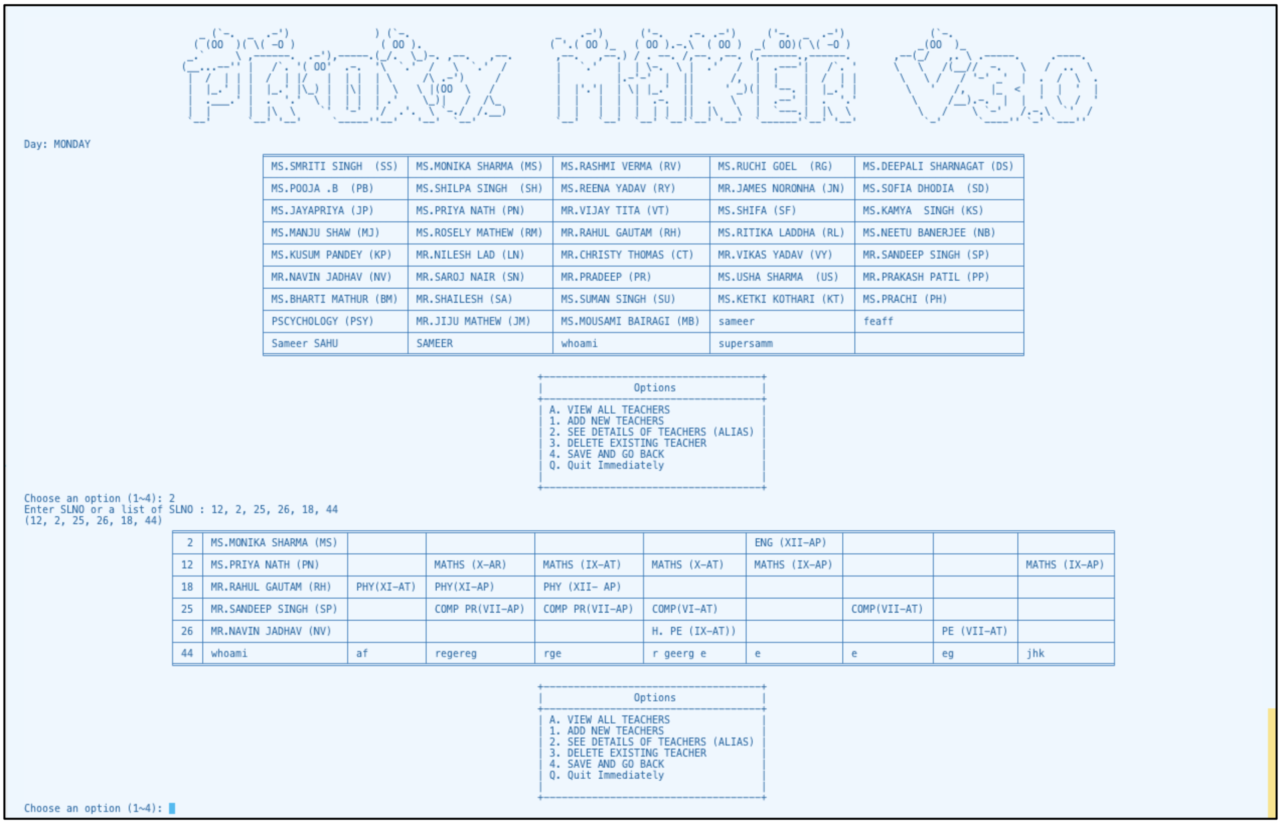
```
Choose an option (1~4): 3
Enter teacher SLNO to delete: 45
```



Day: MONDAY

MS.SMRITI SINGH (SS)	MS.MONIKA SHARMA (MS)	MS.RASHMI VERMA (RV)	MS.RUCHI GOEL (RG)	MS.DEEPALI SHARNAGAT (DS)
MS.POOJA .B (PB)	MS.SHILPA SINGH (SH)	MS.REENA YADAV (RY)	MR.JAMES NORONHA (JN)	MS.SOFIA DHODIA (SD)
MS.JAYAPRIYA (JP)	MS.PRIYA NATH (PN)	MR.VIJAY TITA (VT)	MS.SHIFA (SF)	MS.KAMYA SINGH (KS)
MS.MANJU SHAW (MJ)	MS.ROSELY MATHEW (RM)	MR.RAHUL GAUTAM (RH)	MS.RITIKA LADDHA (RL)	MS.NEETU BANERJEE (NB)
MS.KUSUM PANDEY (KP)	MR.NILESH LAD (LN)	MR.CHRISTY THOMAS (CT)	MR.VIKAS YADAV (VY)	MR.SANDEEP SINGH (SP)
MR.NAVIN JADHAV (NV)	MR.SAROJ NAIR (SN)	MR.PRADEEP (PR)	MS.USHA SHARMA (US)	MR.PRAKASH PATIL (PP)
MS.BHARTI MATHUR (BM)	MR.SHAILESH (SA)	MS.SUMAN SINGH (SU)	MS.KETKI KOTHARI (KT)	MS.PRACHI (PH)
PSCYCHOLOGY (PSY)	MR.JIJU MATHEW (JM)	MS.MOUSAMI BAIRAGI (MB)	sameer	feaff
Sameer SAHU	SAMEER	whoami	supersamm	

#Fetching-details-of-teachers-via-SLNO:



```
Day: MONDAY
```

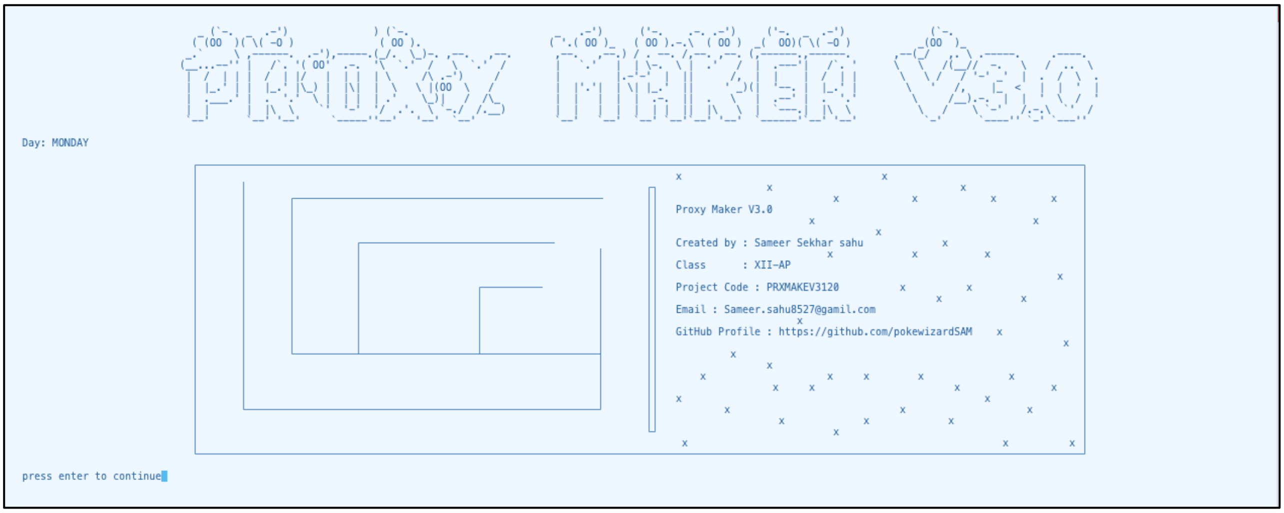
MS.SMRITI SINGH (SS)	MS.MONIKA SHARMA (MS)	MS.RASHMI VERMA (RV)	MS.RUCHI GOEL (RG)	MS.DEEPALI SHARNAGAT (DS)
MS.POOJA .B (PB)	MS.SHILPA SINGH (SH)	MS.REENA YADAV (RY)	MR.JAMES NORONHA (JN)	MS.SOFIA DHODIA (SD)
MS.JAYAPRIYA (JP)	MS.PRIYA NATH (PN)	MR.VIJAY TITA (VT)	MS.SHIFA (SF)	MS.KAMYA SINGH (KS)
MS.MANJU SHAW (MJ)	MS.ROSELY MATHEW (RM)	MR.RAHUL GAUTAM (RH)	MS.RITIKA LADDHA (RL)	MS.NEETU BANERJEE (NB)
MS.KUSUM PANDEY (KP)	MR.NILESH LAD (LN)	MR.CHRISTY THOMAS (CT)	MR.VIKAS YADAV (VY)	MR.SANDEEP SINGH (SP)
MR.NAVIN JADHAV (NV)	MR.SAROJ NAIR (SN)	MR.PRADEEP (PR)	MS.USHA SHARMA (US)	MR.PRAKASH PATIL (PP)
MS.BHARTI MATHUR (BM)	MR.SHAILESH (SA)	MS.SUMAN SINGH (SU)	MS.KETKI KOTHARI (KT)	MS.PRACHI (PH)
PSCYCHOLOGY (PSY)	MR.JIJU MATHEW (JM)	MS.MOUSAMI BAIRAGI (MB)	sameer	feaff
Sameer SAHU	SAMEER	whoami	supersamm	

```
+------------------------------------+
|               Options              |
+------------------------------------+
| A. VIEW ALL TEACHERS               |
| 1. ADD NEW TEACHERS                |
| 2. SEE DETAILS OF TEACHERS (ALIAS) |
| 3. DELETE EXISTING TEACHER         |
| 4. SAVE AND GO BACK                |
| Q. Quit Immediately                |
|                                    |
+------------------------------------+

Choose an option (1~4): 2
Enter SLNO or a list of SLNO : 12, 2, 25, 26, 18, 44
(12, 2, 25, 26, 18, 44)
```

2	MS.MONIKA SHARMA (MS)					ENG (XII-AP)			
12	MS.PRIYA NATH (PN)		MATHS (X-AR)	MATHS (IX-AT)	MATHS (X-AT)	MATHS (IX-AP)			MATHS (IX-AP)
18	MR.RAHUL GAUTAM (RH)	PHY(XI-AT)	PHY(XI-AP)	PHY (XII- AP)					
25	MR.SANDEEP SINGH (SP)		COMP PR(VII-AP)	COMP PR(VII-AP)	COMP(VI-AT)		COMP(VII-AT)		
26	MR.NAVIN JADHAV (NV)				H. PE (IX-AT))			PE (VII-AT)	
44	whoami	af	regereg	rge	r geerg e	e	e	eg	jhk

```
+------------------------------------+
|               Options              |
+------------------------------------+
| A. VIEW ALL TEACHERS               |
| 1. ADD NEW TEACHERS                |
| 2. SEE DETAILS OF TEACHERS (ALIAS) |
| 3. DELETE EXISTING TEACHER         |
| 4. SAVE AND GO BACK                |
| Q. Quit Immediately                |
|                                    |
+------------------------------------+

Choose an option (1~4):
```

#Credit-screen-test:



```
Day: MONDAY

Proxy Maker V3.0

Created by : Sameer Sekhar sahu

Class       : XII-AP

Project Code : PRXMAKEV3120

Email : Sameer.sahu8527@gamil.com

GitHub Profile : https://github.com/pokewizardSAM

press enter to continue
```

Conclusion

This project will make the management of records of period very easy for the admin--stration department by allowing them to swiftly fetch and push data into a simplified database. This will help them in maintaining a clean a record which can be boat easy to read and to understand. We have tried our best to make it user-friendly while being bound by the
constraints of the terminal. Adding a UI would have been useful, but we simply didn’t see it as our major obstacle and tried to tinker around the cool ascii stuff that a terminal can do. We used our every possible methods in our bound to make this useful and stable On every major operating system and recreating functions to avoid excessive dependency. Overall this project was a lot of fun to tweak with and we learned a lot from this endeavor

Cheers !!

Bibliography

Books: computer science with python class XII by : Sumita Arora

Tools: Openllm sponsored by Kaggle and alpaca for debugging

Tabulate docs : https://pypi.org/project/tabulate/

MySQL docs: https://dev.mysql.com/doc/refman/8.2/en/

Sites: stackoverflow for debugging

| Digital version of the file can be downloaded from here |