CITY UNIVERSITY MALAYSIA
Est. 1984

# FACULTY OF INFORMATION TECHNOLOGY

## SUBJECT NAME: FUNDAMENTAL OF COMPUTATIONAL THINKING: PYTHON

## SUBJECT CODE: BIT1054

## LECTURER NAME: NAZMIRUL IZZAD BIN NASSIR

## ASSIGNMENT: FINAL PROJECT – GROUP 20

## PREPARED BY:


| Name | ID | Class Section | Program | Passport Number |
|---|---|---|---|---|
| Hasan Mohammad Kamrul | 202310010059 | Tuesday- 08:00 AM | BIT | A03741506 |
| Sheikh MD Shakil | 202311010024 | Wednesday- 08:00 AM | BIT | A11114107 |
| Rabby MD Fajleh | 202401010268 | Wednesday- 08:00 AM | BIT | A02967639 |
| Islam Nazrul | 202310010029 | Tuesday- 08:00 AM | BIT | A07604502 |

# Project Title: Smart Study plan

## Table of Content:



## Contribution Breakdown by Team members:

| Team Members | Task completed by |
|---|---|
| Sheikh MD Shakil | Problem Research & Proposal. |
| Hasan Mohammad Kamrul | Pseudocode & Flowchart, Core Python Coding. |
| Rabby MD Fajleh | File Handling & Process Tracking. |
| Islam Nazrul | Documentation, Screenshot & Final Report. |

## 1. Introduction & Problem Background

One of the most difficult things for students is managing study time. That could sometimes be good the general standard of what you need to know here is low; most of us spend too long on a subject or end up not catching up on something else to be as well prepared as possible, but sometimes it doesn't come off, and sometimes we forget what we need to get to. There are apps, of course, but a lot of them are overly complicated or require a connection to the internet.

Our group collectively agreed to develop SmartStudy Pro, a study planner which using python to help students plan the most effective study schedule for them and update based on the subjects they believe to be most important for their exams, and be able to mark what they have done, as well as save and load checkpoints, as well as being able to determine if they have completed all their tasks.

This project was selected because time management and self-regulationare important for students. With SmartStudy Pro, we wanted to make study planner that's light yet powerful, ran on python and didn't require separate software or internet.

## 2. Problem Identification and Justification

**Problem:** Students have a hard time juggling multiple subjects, prioritizing the important ones and remember their study plan.

**Justification:** A "smart" planner that acted on demand, assigning hours according to priority, and being able to save/load, that could help improve study habits.

**Intended End Users:** University students that are studying for exams or that are studying for everyday use.

**Goal:** A friendly program written in python to create a custom, prioritized and trackable study schedule.

## 3. Computational Thinking Process

We applied **computational thinking** to design SmartStudy Pro:

### Decomposition

- Input subjects and their priority levels.
- Input total available study hours.
- Calculate hours based on subject priority.
- Display study plan with hours and progress.

- Provide options: update, complete, save, exit.

### Abstraction

- We only consider the essential details: **subject name, priority, allocated hours, and completion status**.
- Irrelevant data like student's free time activities are ignored.

### Pattern Recognition

- High priority subjects get more time than medium or low.
- Updates and completions follow a repeated checking process

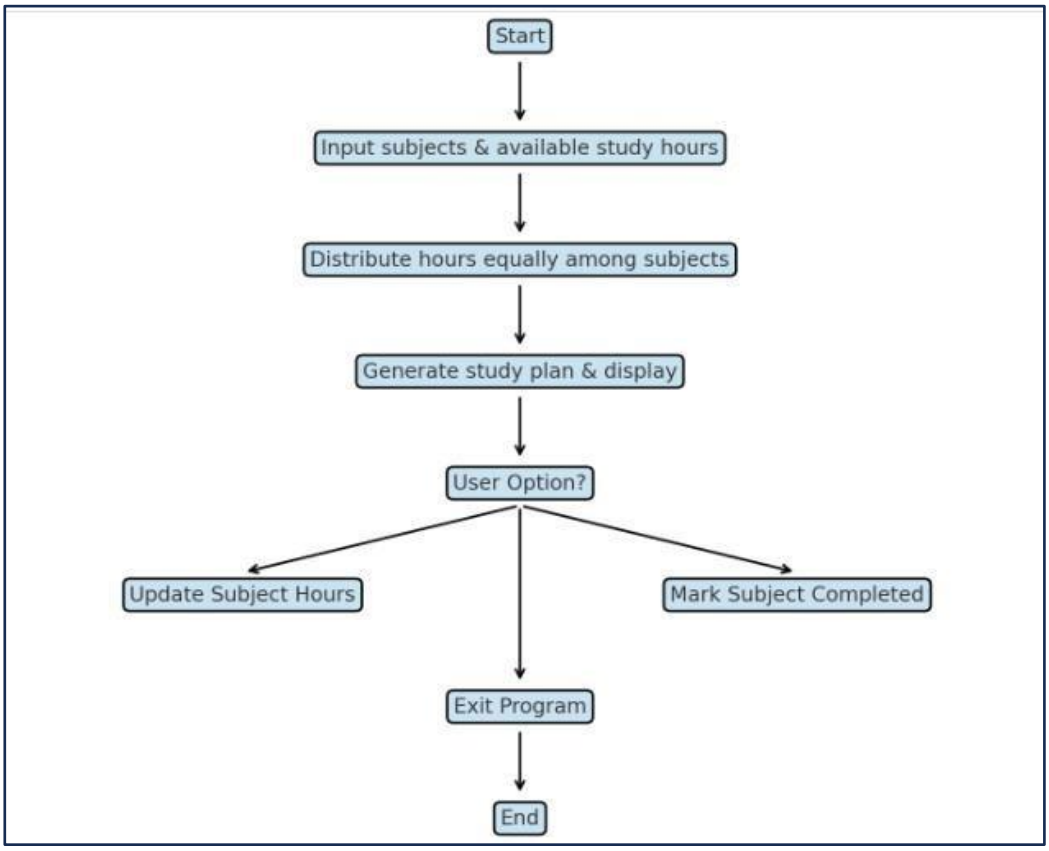
### Algorithmic Thinking

1. Start program.
2. If a saved plan exists, ask to load it.
3. Otherwise, create a new plan by inputting subjects and priorities.
4. Allocate hours proportionally using weights (High=3, Medium=2, Low=1).
5. Display the study plan.
6. Ask the user for options (Update, Complete, Save, Exit).
7. Repeat until exit.
8. Save final plan to file and end.

```
BEGIN SmartStudyPro

 DISPLAY "Welcome to SmartStudy Pro"

 IF saved file exists:
    ASK user "Load saved plan?"
    IF yes:
       LOAD plan
    ELSE:
       CREATE new plan
 ELSE:
    CREATE new plan

 FUNCTION  create_new_plan:
    INPUT subjects
    INPUT  total_hours
    FOR each subject:
       INPUT priority (High/Medium/Low)
    CALCULATE weights based on priority
    ALLOCATE hours proportionally
```

```
    SAVE plan to file

  LOOP UNTIL user chooses Exit:
    DISPLAY current plan (subject, hours, status)
    ASK option (Update, Complete, Save, Exit)

    IF Update:
      INPUT subject & new hours
      UPDATE subject hours
    IF Complete:
      INPUT subject
      MARK as completed
    IF Save:
      SAVE plan to file
    IF Exit:
      SAVE plan and break loop

  DISPLAY "Final plan saved. Good luck!"

END SmartStudyPro
```

Flowchart


Start
Input subjects & available study hours
Distribute hours equally among subjects
Generate study plan & display
User Option?
Update Subject Hours
Mark Subject Completed
Exit Program
End

Python code implementation

```
import json
import os

def save_plan(plan, filename="study_plan.json"):
    with open(filename, "w") as f:
        json.dump(plan, f)

def load_plan(filename="study_plan.json"):
    if os.path.exists(filename):
        with open(filename, "r") as f:
            return json.load(f)
    return None

def create_plan():
    subjects = input("Enter subjects (comma separated): ").split(",")
    subjects = [s.strip() for s in subjects if s.strip()]
    total_hours = int(input("Enter total study hours available: "))

    priorities = {}
    for sub in subjects:
        p = input(f"Set priority for {sub} (High/Medium/Low): ").lower()
        priorities[sub] = p

    # Priority weighting
    weight_map = {"high": 3, "medium": 2, "low": 1}
    total_weight = sum(weight_map[priorities[sub]] for sub in subjects)

    plan = {}
    for sub in subjects:
        weight = weight_map[priorities[sub]]
        allocated_hours = (weight / total_weight) * total_hours
        plan[sub] = {"hours": round(allocated_hours), "completed": False}

    save_plan(plan)
    return plan

def display_plan(plan):
    print("\n SmartStudy Pro Plan:")
    for sub, data in plan.items():
        status = "✅ Done" if data["completed"] else ⏳ Pending"
        print(f"- {sub}: {data['hours']} hour(s) | {status}")
```

```
def update_plan(plan):
    subject = input("Enter subject to update: ").strip()
    if subject in plan:
        new_hours = int(input("Enter new hours: "))
        plan[subject]["hours"] = new_hours
        print(f"✅ Updated {subject} to {new_hours} hour(s).")
    else:
        print("❌ Subject not found.")

def mark_completed(plan):
    subject = input("Enter subject completed: ").strip()
    if subject in plan:
        plan[subject]["completed"] = True
        print(f"🎉 Great! You completed {subject}.")
    else:
        print("❌ Subject not found.")

# Main Program
print("===== SmartStudy Pro =====")
plan = load_plan()

if plan:
    choice = input("Found a saved plan. Load it? (y/n): ").lower()
    if choice != "y":
        plan = create_plan()
else:
    plan = create_plan()

while True:
    display_plan(plan)
    choice = input("\nOptions: [U]pdate, [C]omplete, [S]ave, [E]xit: ").lower()
    if choice == "u":
        update_plan(plan)
    elif choice == "c":
        mark_completed(plan)
    elif choice == "s":
        save_plan(plan)
        print("💾 Plan saved successfully.")
    elif choice == "e":
        save_plan(plan)
        print("👋 Final plan saved. Good luck!")
        break
    else:
```

```
    print("Invalid option. Try again.")
```

## 4. Feature

**Feature 1: Progress Percentage**

- Each subject now shows how much of its allocated hours is completed (e.g., Math 3/5 → 60%).
- Students can mark partial completion instead of full subject done.

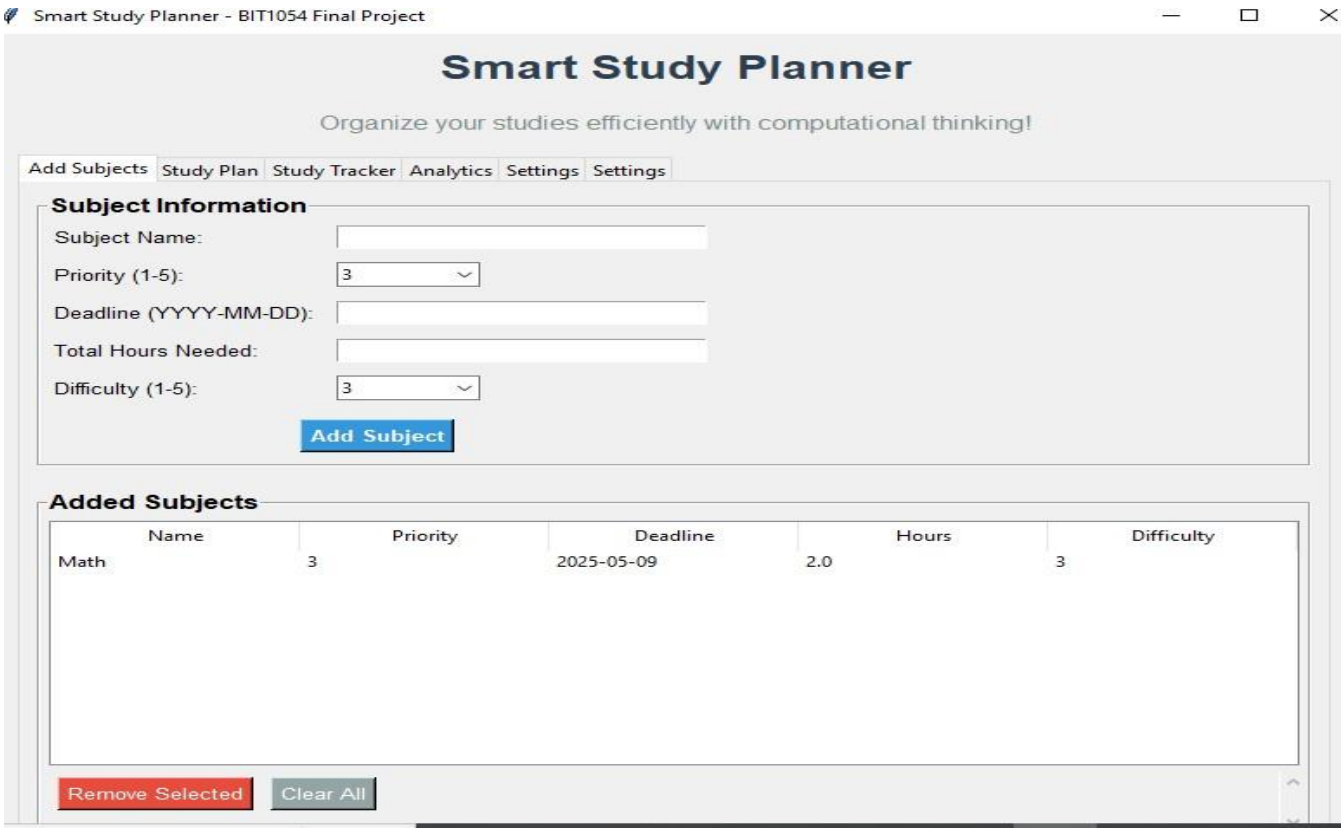
**Feature 2: Subject Summary**

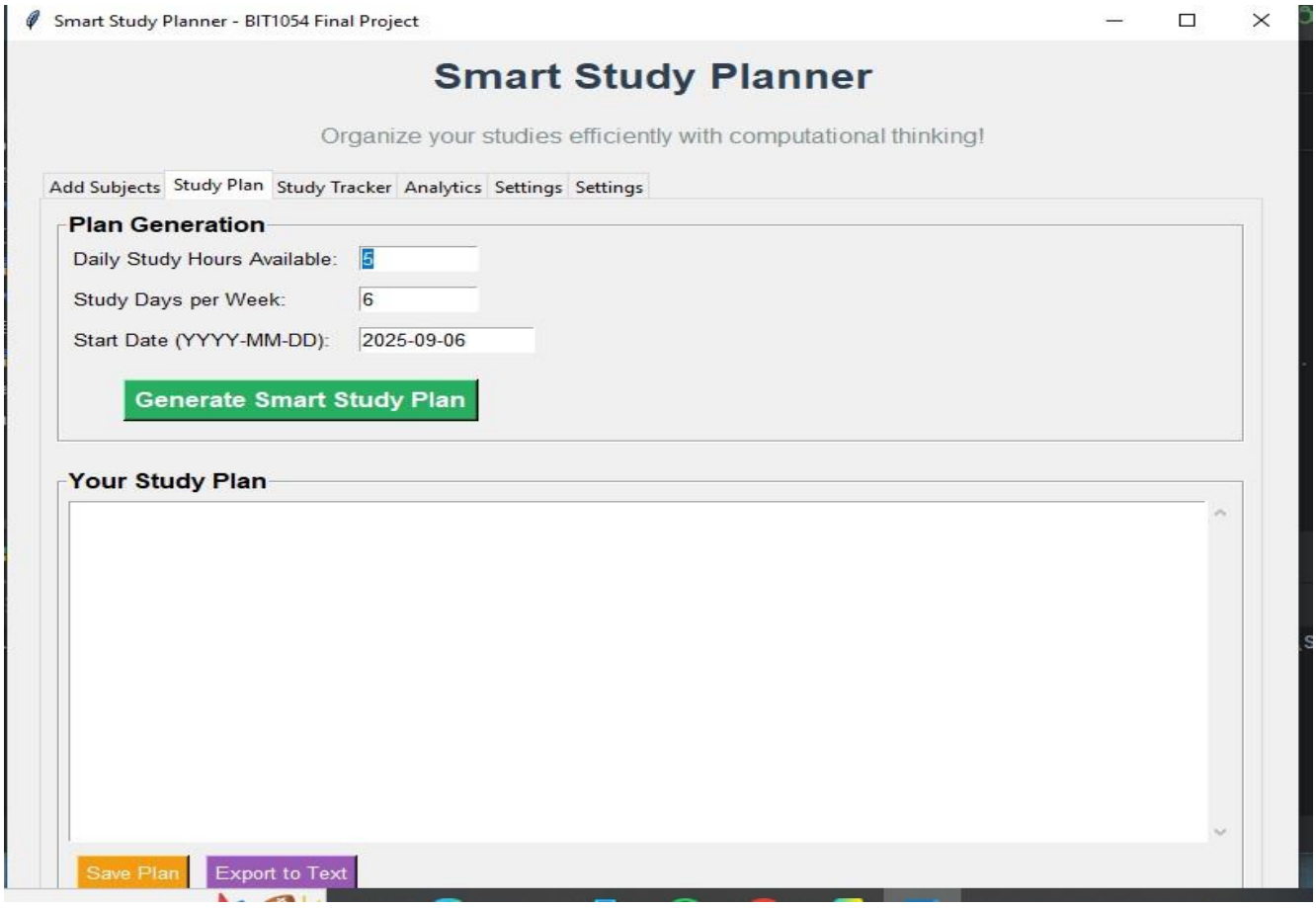
- At any time, students can see total subjects, completed subjects, pending subjects, and overall progress %.

**Feature 3: Reset Plan Option**

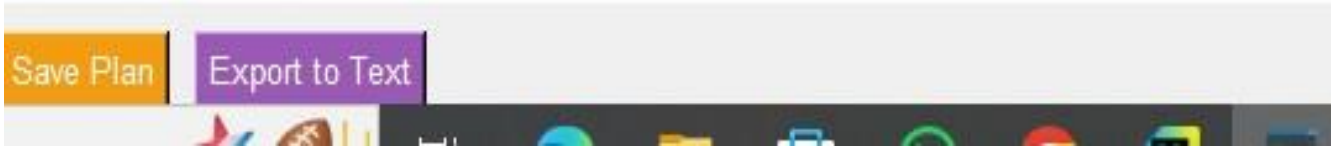
- If students want to start fresh, they can reset the entire plan without closing the program.

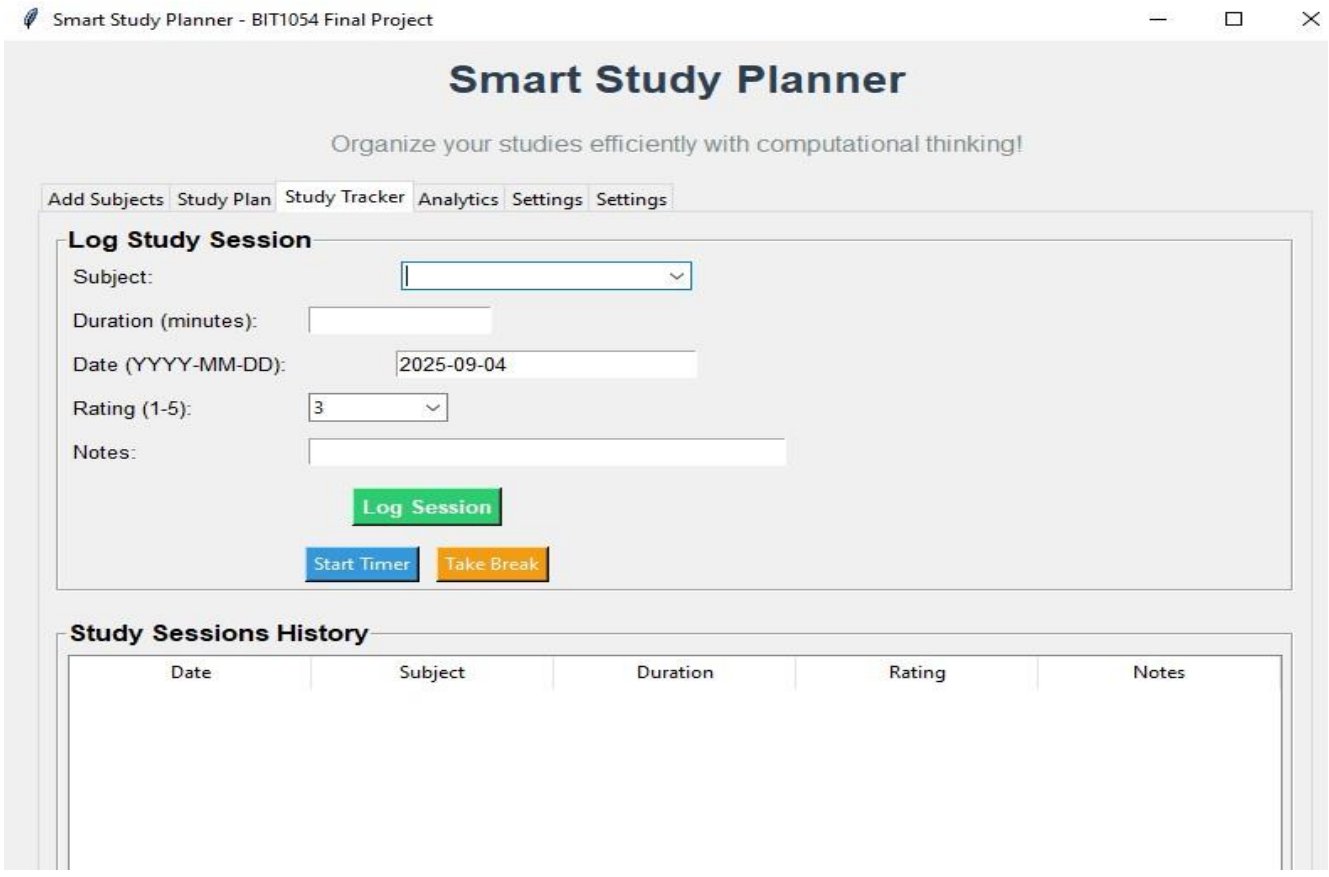
## 5. Visuals Output

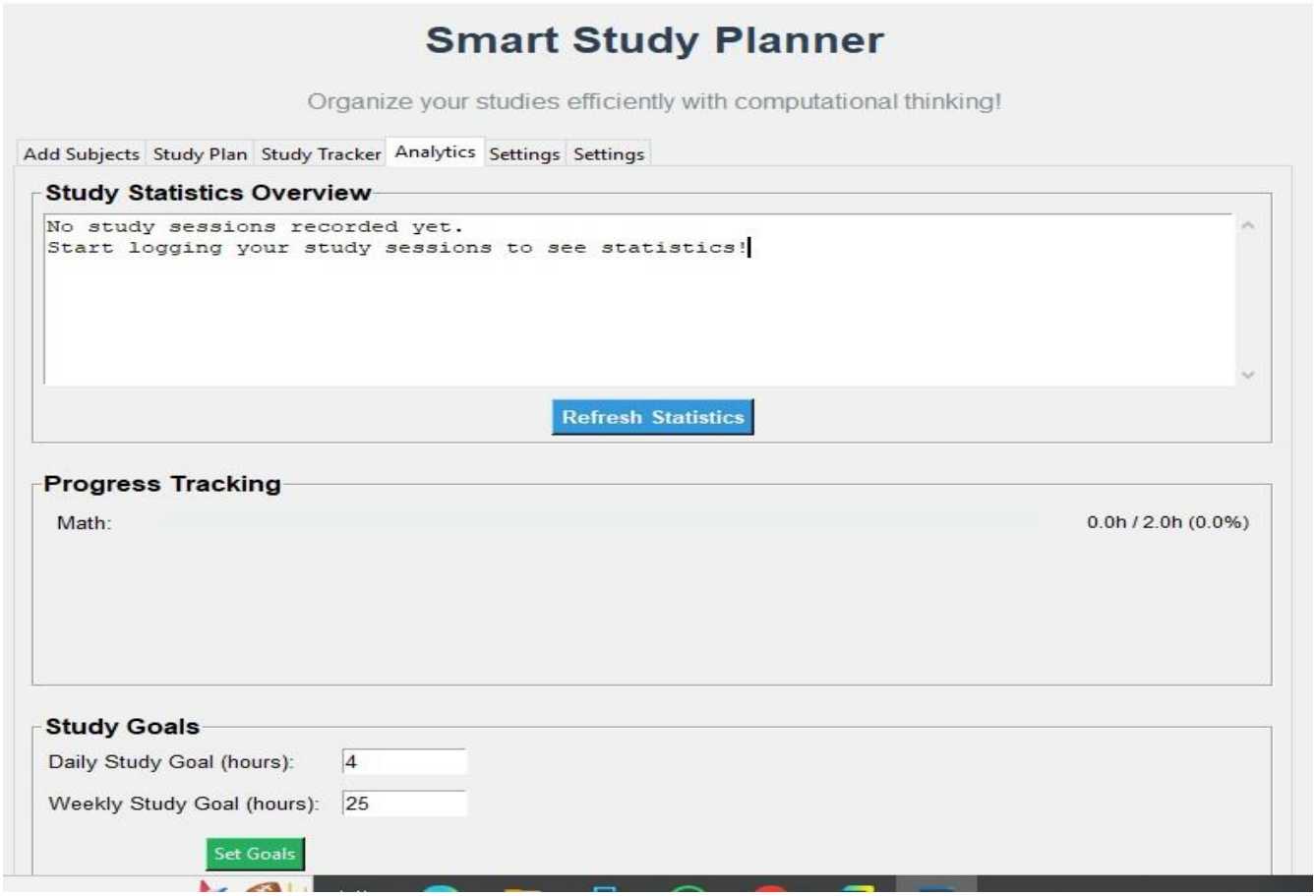


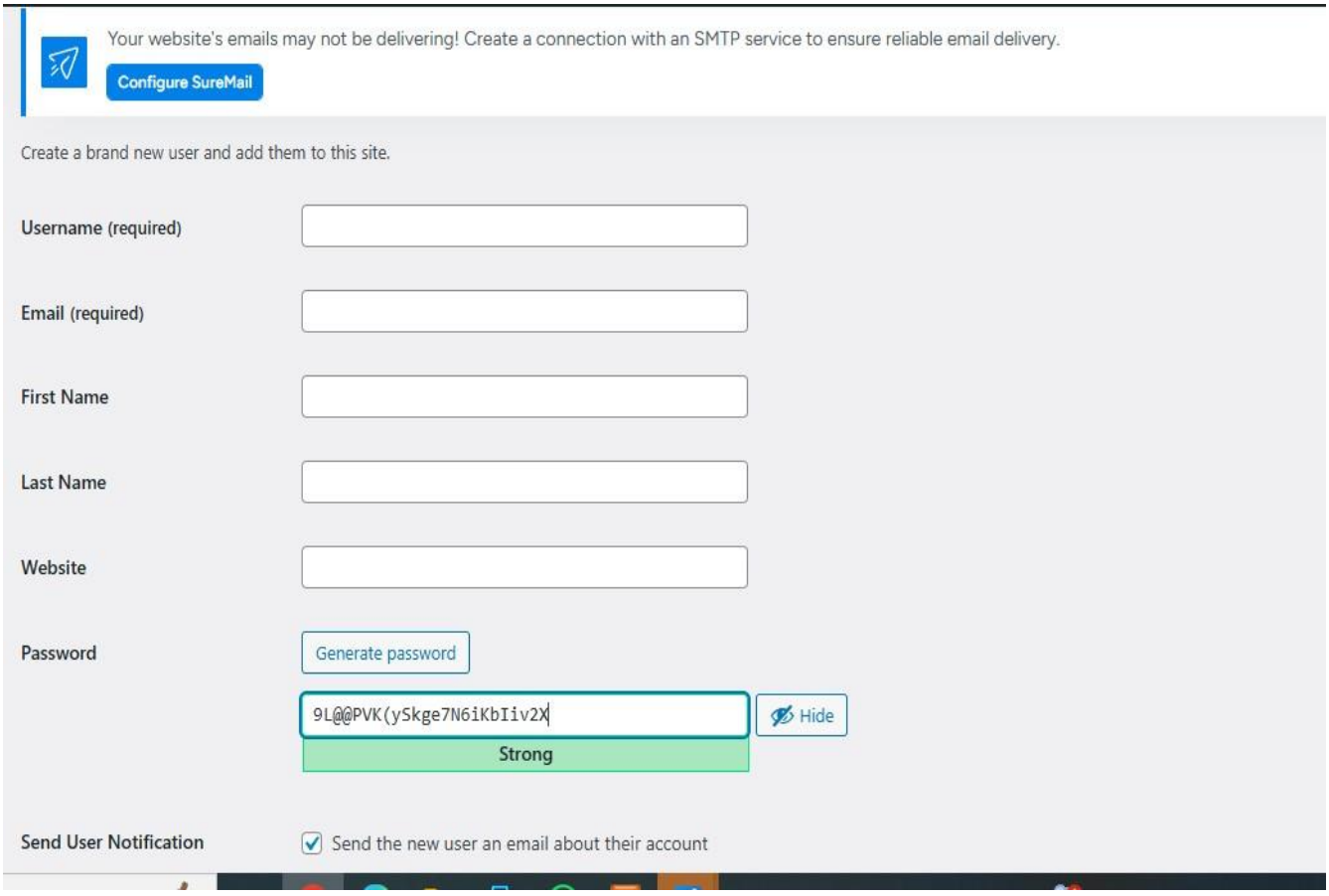



```
============================================================
MART STUDY PLAN - GENERATED WITH COMPUTATIONAL THINKING
============================================================
enerated on: 2025-09-04 12:23:04
lan starts: 2025-09-06
aily study hours: 5.0
tudy days per week: 6
otal hours needed: 2.0

UBJECTS OVERVIEW (Sorted by Urgency):
----------------------------------------
. Math
   Priority: 3/5, Difficulty: 3/5
   Deadline: 2025-05-09, Hours needed: 2.0
   Urgency Score: 100.0/100

AILY STUDY SCHEDULE:
```




Smart Study Planner
Organize your studies efficiently with computational thinking!
Add Subjects
Study Plan
Study Tracker
Analytics
Settings
Settings
Study Statistics Overview
No study sessions recorded yet.
Start logging your study sessions to see statistics!
Refresh Statistics
Progress Tracking
Math:
0.0h / 2.0h (0.0%)
Study Goals
Daily Study Goal (hours):
4
Weekly Study Goal (hours):
25
Set Goals


Your website's emails may not be delivering! Create a connection with an SMTP service to ensure reliable email delivery.
Configure SureMail
Create a brand new user and add them to this site.
Username (required)
Email (required)
First Name
Last Name
Website
Password
Generate password
9L@@PVK(ySkge7N6iKbIiv2X
Hide
Strong
Send User Notification
Send the new user an email about their account


## 6. Reflection Task Division

**Reflection**

This project is what taught us how to turn a daily student problem into a functional Python solution. Initially, we only intended to divide the time perceiver equally but later, we considered that it may not always the case, so came up with a priority driven allocation mechanism.

We also took a look at how to use file handling (JSON) so that the plans could be saved and loaded, thus having it feel a bit more like a real-world application. Having the program be user friendly and simple all at once was the biggest challenge.

## Conclusion

SmartStudy Pro successfully tackles the issue of disorganized preparation scheduling, with a priority-based, customize-able, savable study planner. It shows computational thinking, the correct usage of Python basics (loops, conditionals, functions, file I/O), and good teamwork. It helped us to enhance our programming skills, problem-solving method, and working as a group. We are glad to say that SmartStudy Pro isn't just a coding task, but a neat tool that the students can really use in real life.