# CITY UNIVERSITY

## BIT1054FUNDAMENTAL OF COMPUTATIONAL THINKING:PYTHON

**GROUP PROJECT: PROPOSAL**

**TUTOR: SIR NAZMIRUL IZZAD BIN NASSIR**

**PREPARED BY**

| NAME | ID | CLASS |
|---|---|---|
| HASAN MOHAMMAD KAMRUL | 202310010059 | TUESDAY(8.00 TO 11.00) |
| SHEIKH MD SHAKIL | 202311010024 | WEDNESDAY(8.00 TO 11.00) |
| RABBY MD FAJLEH | 202401010268 | WEDNESDAY(8.00 TO 11.00) |
| ISLAM NAZRUL | 202310010029 | TUESDAY(8.00 TO 11.00) |

# Project Proposal Document

**PROJECT TITLE:** *SMART STUDY PLANNER*

– *Personal Deadline Boss*

## THE FIRST IS- PROBLEM IDENTIFICATION AND JUSTIFICATION

Every student knows this story: you promise yourself you'll start that assignment early… then suddenly the deadline is tomorrow, you've done nothing, and now you're drinking coffee at midnight wondering where your life went wrong.

The truth is, most of us aren't bad at studying — we're bad at planning our time. We either forget the deadlines, underestimate how long tasks will take, or keep jumping between subjects without a clear strategy.

This project aims to solve that with a simple Python program that acts like a mini "time boss" — you tell it what needs to be done, when it's due, and how many hours it might take, and it will break it down into daily study goals. That way, you can actually finish before the deadline without sacrificing your sleep (or your sanity).

## THE SECOND IS - COMPUTATIONAL THINKING ANALYSIS

**Decomposition** – Break the big problem into steps:

1. Input: task name, deadline, total hours needed.
2. Calculate: days left until deadline.
3. Divide: total hours ÷ days left = daily study hours.
4. Store: keep tasks in a simple list.

5. Output: show a clean summary of all tasks and how much to study each day.

**Abstraction** – Ignore what's not essential:

- No fancy graphics, no location info, no break timers.
- Just the key facts: task name, deadline, hours.

**Algorithm** – Step-by-step logic:

1. Ask how many tasks the user wants to plan.
2. For each task, get name, deadline, and total hours.
3. Convert deadline to a date and find days left.
4. If days left $\leq 0$ → warn that it's too late.
5. Otherwise → calculate daily hours and save it.
6. At the end → print the full study schedule.

## THE THIRD IA - PROPOSED PYTHON-BASED SOLUTION

The program will be a simple, text-based tool that runs in the terminal. Users will:

a.Enter all their study tasks.

b.Get instant calculations for how much to study per day.

c.See warnings if deadlines are too close.

This makes the program perfect for students who want a quick and no-nonsense way to organize their time without downloading complex apps.

## THE FINAL IS - EXPECTED BENEFITS

a. **No more last-minute panic** — know exactly what to do every day.

b. **Saves mental energy** — no need to manually figure out study hours.

c. **Better time management** — spreads workload evenly.

d. **Peace of mind** — you'll actually know if you're on track or in trouble.

## THE OBJECTIVES –

Short, bullet-point goals of your project.
Example:

a. Automate daily study schedule creation.

b. Prevent last-minute cramming.

c. Make a simple and fast tool for students.