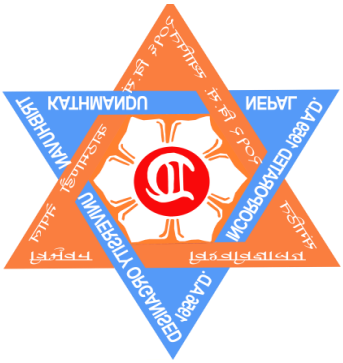

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING THAPATHALI CAMPUS

Proposal

On

StudyBuddyyy - Your Personalized AI Learning Companion

Submitted by:

Abinash Yadav [THA081BEI002]

Rupeshwar Sah [THA081BEI035]

Laxman Shakya [THA081BEI016]

Muhan Narshingh Shrestha [THA081BEI021]

Submitted to:

Department of Electronics and Computer Engineering

Thapathali Campus Kathmandu, Nepal

February, 2024

ACKNOWLEDGEMENTS

The successful development of the StudyBuddyyy project would not have been possible without the generous support and guidance of several individuals and resources. We would like to express our sincere gratitude to everyone.

We are particularly grateful to Professor Prajwal Pakka for his flexibility and understanding in allowing us to adopt a hybrid C and Python approach for StudyBuddyyy. His permission to leverage Python for the complex Artificial Intelligence and API handling tasks, despite the project's foundation in C, was crucial to its success. This strategic decision enabled us to effectively utilize Python's libraries and tools, enhancing both development efficiency and the overall functionality of StudyBuddyyy. Professor Pakka's open-mindedness and willingness to embrace innovative solutions have been truly inspiring and have greatly contributed to the technical capabilities and potential of this project.

Thank you, Professor Pakka, for your essential support and encouragement.

ABSTRACT

In today's demanding academic environment, students face significant challenges in efficiently processing and retaining vast amounts of information. Traditional study methods often prove time-consuming and less engaging, leading to information overload and passive learning experiences. This project proposes the development of StudyBuddyyy, a desktop application designed to function as a personalized learning companion. StudyBuddyyy aims to streamline the learning process by offering on-demand generation of study notes and quizzes on diverse topics, concise summarization of YouTube video content, and the creation of visual mind maps to enhance concept understanding. By integrating a user-friendly C-based interface with a powerful Python-based content generation engine powered by the learning-toolbox library, StudyBuddyyy seeks to empower students with an accessible and effective tool to optimize their study habits and improve learning outcomes. This proposal outlines the project's objectives, methodology, system architecture, and expected deliverables, demonstrating its potential to address the identified challenges and provide a valuable resource for students.

Keywords: Educational Tool, Learning Assistant, learning-toolbox, Mind Map, Content Generation, Study Notes, YouTube Summarizer, Hybrid Application

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF ABBREVIATIONS

API	Application Programming Interface
UI	User Interface
MD	Markdown
PDF	Portable Document Format
LLM	Large Language Model

1. INTRODUCTION

1.1 Background Introduction

In the rapidly evolving landscape of education, the demand for effective and personalized learning tools is constantly increasing. Students are continuously challenged to absorb and synthesize vast amounts of information across diverse subjects. Traditional learning methods, while foundational, often fall short in addressing the needs of modern learners who require efficient, engaging, and readily accessible resources to aid their study process. The integration of technology into education offers promising avenues for creating innovative solutions that can enhance learning outcomes and make studying a more productive and less daunting experience. This project, StudyBuddyyy, aims to contribute to this technological evolution in education by developing a user-centric application designed to personalize and optimize the learning journey.

1.2 Motivation

The primary motivation behind StudyBuddyyy stems from the observed inefficiencies and challenges inherent in conventional study practices. Students frequently encounter:

- Information Overload: The sheer volume of study material can be overwhelming, leading to stress and reduced comprehension.
- Time-Intensive Content Creation: Manually creating study notes, quizzes, and visual aids is a laborious process, diverting time from actual learning.
- Need for Active Learning: Passive reading is often less effective than active engagement with the material, highlighting the need for interactive learning tools.
- Leveraging Video Resources Efficiently: While platforms like YouTube offer immense educational content, extracting key information and creating summaries from videos can be time-consuming.

These pain points underscore the need for a tool that can automate content creation, facilitate active learning, and efficiently process information, thereby motivating the development of StudyBuddyyy as a personalized learning companion.

1.3 Problem Definition

This project directly addresses the following key problems in the current learning environment:

- Inefficient Study Material Generation: Students lack readily available tools to quickly generate comprehensive study materials (notes, quizzes) tailored to specific topics, leading to wasted time and effort.
- Lack of Integrated Learning Resources: There is a need for a unified platform that integrates various learning functionalities, such as note generation, video summarization, and mind mapping, into a single, user-friendly application.
- Limited Support for Diverse Learning Styles: Current tools often do not cater to diverse learning preferences, such as visual learning (mind maps) and active recall (quizzes), hindering effective knowledge retention.
- Time-Consuming Video Content Processing: Students struggle to efficiently extract key information from educational videos, requiring a solution to rapidly summarize video content for study purposes.

1.4 Objectives

The main objectives of this project are:

- To develop a functional desktop application (StudyBuddyyy) that provides users with on-demand generation of study notes, quizzes, and mind maps on user-specified topics.
- To implement a YouTube video summarization feature within StudyBuddyyy, enabling users to obtain concise summaries of video content for efficient learning.
- To create a user-friendly interface in C that seamlessly integrates with a Python-based content generation engine, leveraging the learning-toolbox library.

2. LITERATURE REVIEW

2.1 Traditional Learning Methods

- Traditional learning methods encompass textbook-based learning, classroom lectures, and manual note-taking. These methods have been the cornerstone of education for centuries.
- These methods rely on linear presentation of information in textbooks, one-way communication in lectures, and handwritten notes for information processing and retention. Technology is minimally integrated, primarily for content delivery (e.g., projectors).
- Traditional methods provide a structured foundation for learning, fostering discipline, in-depth study of core concepts, and direct interaction with educators.
- These methods can be passive, time-consuming in content creation (manual notes), less adaptable to diverse learning styles, and may struggle with the volume of information in modern curricula. They often lack interactive and personalized elements.
- While fundamental, traditional methods often fail to efficiently address information overload, cater to visual or active learners, or leverage technology to enhance engagement and efficiency in study material creation.
- The limitations of traditional methods, particularly in efficiency and engagement, directly fuel the motivation for StudyBuddyyy. StudyBuddyyy aims to overcome these drawbacks by automating content generation, offering interactive quizzes and mind maps, and leveraging technology to streamline the learning process, addressing the core problems outlined in the motivation section. [1]

2.2 Digital Learning Platforms and Tools

- This area includes online learning platforms (e.g., Coursera, Khan Academy), note-taking applications (e.g., Evernote, OneNote), and basic study tools available online.
- Digital platforms utilize web-based interfaces, video lectures, interactive exercises, and cloud-based storage. Note-taking apps offer digital note organization and syncing. Algorithms are used for content delivery and basic personalization in some platforms.
- Digital tools offer flexibility, accessibility, and a wider range of learning resources. They can enhance organization, collaboration (in some platforms), and provide access to global educational content.
- Many platforms are generic, not personalized study assistants. Note-taking apps require manual content creation. Integration between different tool types is often lacking.

Content generation features are typically limited or absent in readily available tools for students.

- While digital tools offer improvements, they often lack the specific functionality of automated, personalized content generation and integration of diverse study methods within a single desktop application. Many online platforms are course-centric rather than topic-centric study aids.
- The limitations of existing digital tools, particularly the lack of integrated content generation and personalized study assistance, directly justify the need for StudyBuddyyy. StudyBuddyyy aims to bridge this gap by providing a focused, desktop-based application that actively generates study materials and integrates key learning functionalities, offering a more streamlined and efficient solution compared to disparate online resources.

3. PROPOSED SYSTEM ARCHITECTURE

3.1 System Architecture

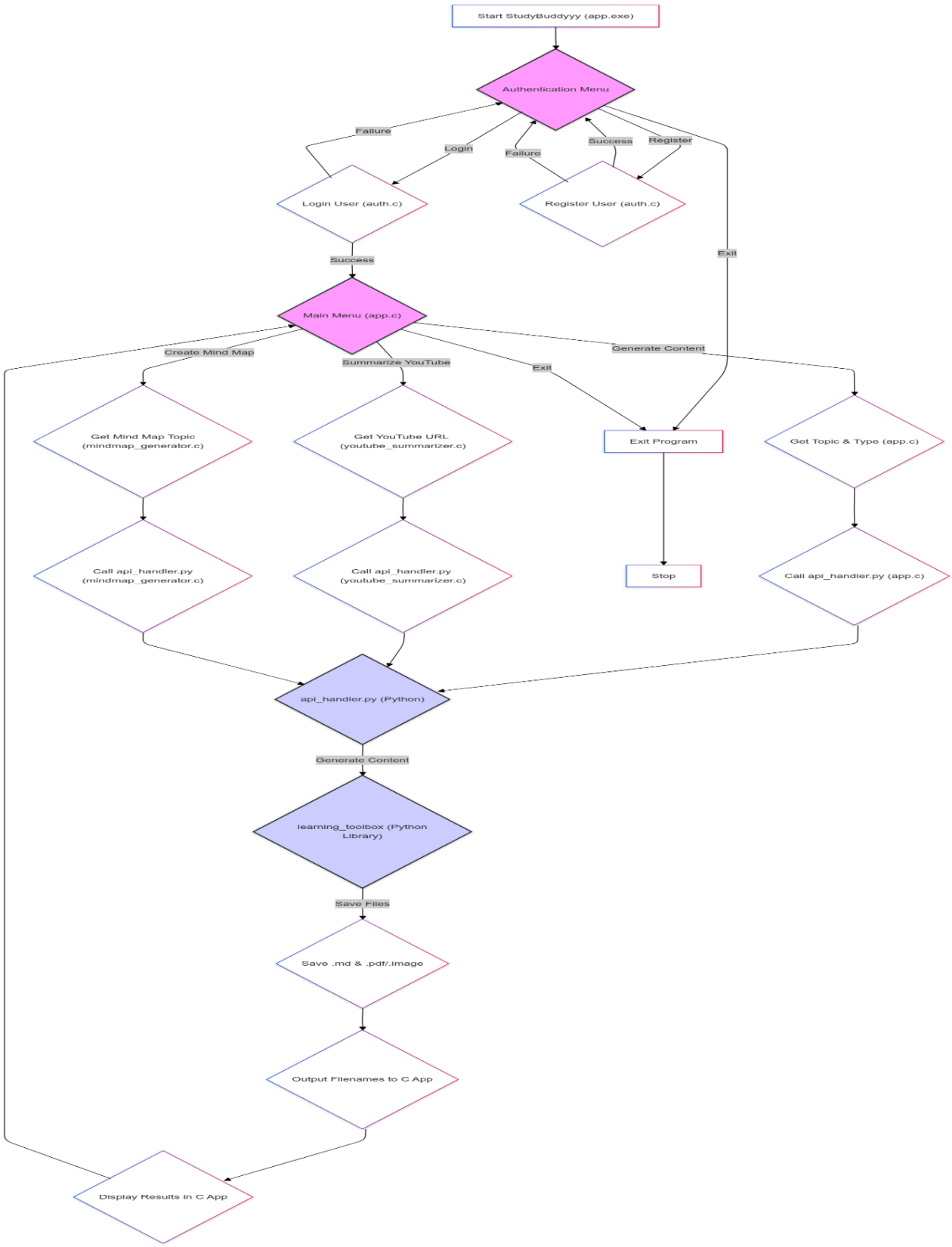

Figure 1 System Architecture

StudyBuddyyy is designed as a hybrid application, combining the strengths of C for user interface and Python for content generation. The core components and their interactions are as follows (visualized in Figure 1: **System Architecture**)

- User Interface (C Application): This module, developed in C, provides the user-facing application. It handles user authentication (login/registration), presents menus and prompts, captures user input (topics, URLs, content types), and displays output to the user. It acts as the control center for the application.
- API Handler (Python Script): This Python script, api_handler.py, serves as the bridge between the C application and the content generation engine. It receives requests from the C application (topic, content type), interacts with the learning-toolbox [2] library, and returns generated content and file information back to the C application.
- Content Generation Engine (learning-toolbox Library): This external Python library is the core of the content generation functionality. It is responsible for generating study notes, quizzes, YouTube video summaries, and mind maps based on user requests. It uses LLM (Large Language Model) for content creation.
- User Database (users.dat): This file stores user authentication information (usernames and passwords). Note: For enhanced security, password hashing will be considered for future iterations. [3]
- Output Files (.md, .pdf, .image): Generated study materials are saved as files in Markdown (.md), Portable Document Format (.pdf), and image formats (for mind maps). [4]

3.2 Data Flow Diagram

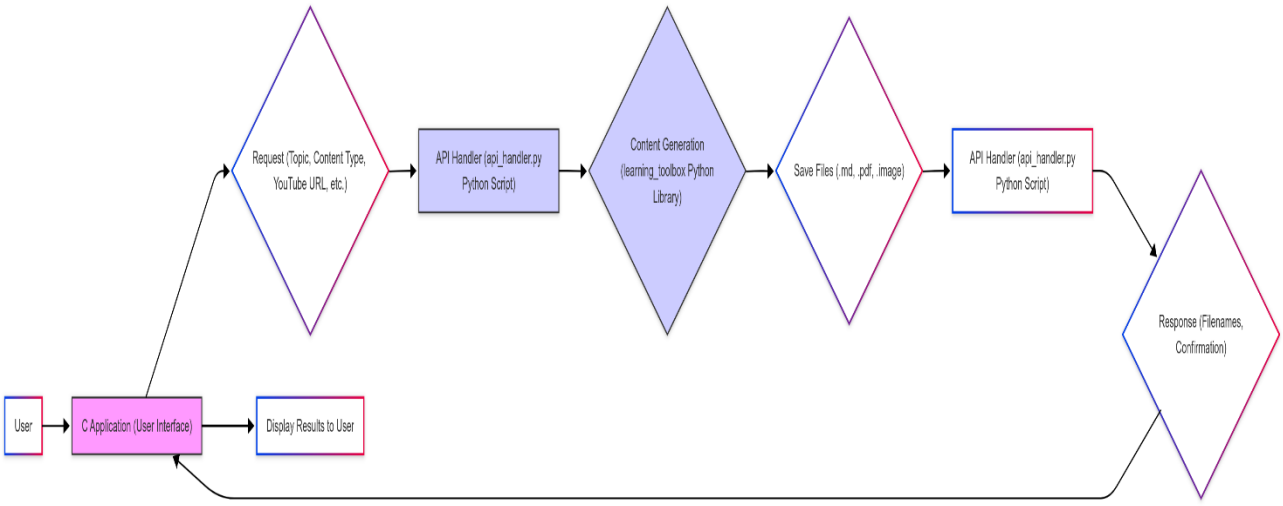

Figure 2 Data Flow Diagram

The data flow within StudyBuddyyy proceeds as follows:

- User Interaction: The user interacts with the C Application's User Interface to select a feature (e.g., Generate Notes) and provide input (e.g., topic "Quantum Physics").
- Request to Python Script: The C Application constructs a command-line request containing the user input and feature request and executes the api_handler.py Python script.
- Content Generation by Python: The api_handler.py script receives the request, utilizes the learning-toolbox library to generate the requested content (e.g., study notes on "Quantum Physics").
- File Saving: The learning-toolbox library (or api_handler.py) saves the generated content as files (e.g., Quantum_Physics_notes.md, Quantum_Physics_notes.pdf).
- Response to C Application: The api_handler.py script outputs filenames and potentially content information to the standard output. The C Application reads this output.
- Display Results: The C Application parses the output from the Python script and displays relevant information (filenames, confirmation messages) to the user through the User Interface.

3.3 Tools and Environment

The following tools and technologies will be utilized for project development:

- Programming Languages: C (for application UI and control flow), Python (for content generation logic).
- C Compiler: GCC (or MinGW for Windows).
- Python Interpreter: Python 3.x.
- Python Libraries: learning-toolbox (version 0.2.0), potentially others as required by learning-toolbox.
- Node.js and npm: For @mermaid-js/mermaid-cli (mind map feature).
- Operating System: Windows (for development and target platform). Cross-platform compatibility may be considered for future enhancements.
- IDE/Text Editors: VS Code, Code::Blocks, or preferred text editors.

4. METHODOLOGY

The development of StudyBuddyyy will follow an iterative and phased methodology, focusing on incremental feature implementation and rigorous testing. The project will be divided into the following key phases:

4.1 Phase 1: Core Structure & Authentication

This initial phase will establish the foundational structure of the StudyBuddyyy application.

- *Tasks:*
    - Set up the C project environment and create the basic project structure (file organization, build system).
    - Develop the core User Interface framework in C, including menus, basic input/output handling, and screen management functions (using stdio.h, windows.h, etc.).
    - Implement user authentication functionalities (registration and login) using C and file-based storage (auth.c, auth.h, config.h).
    - Implement basic error handling and input validation within the C application.

4.2 Phase 2: Content Generation Engine

This phase will focus on integrating the Python-based content generation capabilities with the C application.

- *Tasks:*
    - Develop the api_handler.py Python script to act as the interface to the learning-toolbox library.
    - Implement the "Generate Learning Content" feature (study notes and quizzes) within the C application (app.c), including topic and content type input prompts.
    - Establish inter-process communication between the C application and the Python script using _popen (or similar mechanism).
    - Parse and process output from the Python script within the C application to display generated content and filenames to the user.
    - Implement file saving functionality for generated Markdown and PDF files within the Python script (leveraging learning-toolbox's save_files function).

4.3 Phase 3: Feature Integration & Testing

This phase will focus on integrating the remaining features (YouTube summarizer, mind map generator) and conducting comprehensive testing.

- *Tasks:*
    - Develop and integrate the "Summarize YouTube Video" feature (youtube_summarizer.c, youtube_summarizer.h) in the C application, utilizing api_handler.py and learning-toolbox.
    - Implement the "Create Mind Maps" feature (mindmap_generator.c, mindmap_generator.h) in the C application, similarly utilizing api_handler.py and learning-toolbox.
    - Conduct thorough testing of all features (authentication, content generation, and summarization, mind maps) to identify and resolve bugs and usability issues.
    - Optimize application performance and user experience based on testing feedback.
    - Develop project documentation, including a user guide and code comments.

5. SCOPE AND APPLICATIONS

StudyBuddyyy is designed to provide the following core functionalities:

- User Authentication: Secure user registration and login system to manage user accounts.
- On-Demand Study Note Generation: Generate detailed study notes on a user-specified topic in Markdown and PDF formats.
- Interactive Quiz Generation: Create quizzes on a user-specified topic in Markdown and PDF formats.
- YouTube Video Summarization: Summarize YouTube video content from a provided URL into concise text summaries in Markdown and PDF formats.
- Mind Map Generation: Create visual mind maps for a given topic, saved as image files.
- Command-Line Interface: User interaction through a user-friendly command-line interface.
- File Output: Generated study materials saved as .md, .pdf, and image files in the application directory.

The primary applications of StudyBuddyyy are:

- Personalized Study Aid for Students: Empowering students with a tool to efficiently generate study materials for various subjects.
- Time-Saving Resource for Learning: Reducing the time spent on manual note-taking and content creation, allowing students to focus on understanding concepts.
- Enhanced Learning Engagement: Providing interactive quizzes and visual mind maps to promote active learning and knowledge retention.
- Efficient YouTube Video Learning: Facilitating quick comprehension of video-based educational content through automated summarization.

6. TIME ESTIMATION

- Phase-wise Time Allocation (Approximate):
    - Phase 1: Core Structure & Authentication: 1 Day (Feb – 10)
    - Phase 2: Content Generation Engine: 1 Day (Feb – 11)
    - Phase 3: YouTube Summarizer: 1 Day (Feb – 12)
    - Phase 4: Mind Map Generator: 1 Day (Feb – 13)
    - Phase 5: Testing & Refinement: 1 Day (Feb – 14)
- Total Project Duration: 5 Days

7. FEASIBILITY ANALYSIS

StudyBuddyyy demonstrates feasibility across several dimensions:

- **Practical Use**: StudyBuddyyy directly addresses a practical need for students by providing a tool to efficiently generate study materials and process learning resources. It offers tangible benefits in terms of time-saving and enhanced learning efficiency, making it a practically valuable application for students across various disciplines. Its desktop application format ensures accessibility and ease of use.
- **Cost-Effectiveness**: The project is highly cost-effective, primarily utilizing readily available and open-source software tools (GCC, Python, learning-toolbox, etc.). The development cost is minimal, focusing on development effort rather than expensive software or hardware resources. The reliance on the learning-toolbox library leverages existing content generation capabilities, reducing development complexity and time.
- **Technical Complexity**: While the project involves a hybrid architecture and integration of different technologies (C, Python, external libraries), the technical complexity is deemed manageable within the project timeframe and scope. The modular design and phased methodology will allow for systematic development and testing, mitigating potential technical challenges. The use of established libraries like learning-toolbox simplifies complex content generation tasks.

Overall, StudyBuddyyy presents a feasible and valuable project with strong practical applications, cost-effectiveness, and manageable technical complexity, making it a worthwhile endeavor for development and potential future enhancements.

References

[1] T. Milberg, "The future of learning: How AI is revolutionizing education 4.0," 28 4 2024. [Online]. Available: https://www.weforum.org/stories/2024/04/future-learning-ai-revolutionizing-education-4-0/.

[2] A. Yadav, "learning_toolbox," [Online]. Available: https://pypi.org/project/learning-toolbox/0.2.0/.

[3] [Online]. Available: https://www.geeksforgeeks.org/basics-file-handling-c/.

[4] "Markdown To PDF," [Online]. Available: https://stackoverflow.com/questions/75896773/markdown-to-pdf-for-python.

[6] "Auth," [Online]. Available: https://medium.com/@sams.iitdu/authentication-system-in-c-programming-fe96a6c315e8.