KATHMANDU UNIVERSITY

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report on
"LearnBuddy"

(Code No:COMP 311)

(For partial fulfillment of 3rd Year/1st Semester in Computer Science)

Submitted By:
Saurav Maske(10)
Utsav Subedi(18)
Aarju Taujale(19)
Aakash Thakur(20)

Submitted To:

Suman Shrestha

Department of Computer Science And Engineering

Submission Date:16/11/2025

Abstract

LearnBuddy is an adaptive intelligent learning system designed to provide students with equal access to high-quality educational support. Many existing e-learning platforms deliver the same content to all learners, which can leave students with different abilities underserved. LearnBuddy addresses this challenge by using artificial intelligence to assess each student's performance, predict suitable difficulty levels, and generate quiz questions tailored to their individual learning needs. This ensures that every student receives the right level of challenge and support, enabling them to learn at their own pace while building confidence and understanding of school-level concepts. In addition to adaptive quizzes, LearnBuddy includes a built-in AI chatbot that provides explanations, answers questions, and guides learners through difficult topics, making the learning experience more interactive and engaging. Developed using Flutter for a mobile-first interface, FastAPI for the backend, PostgreSQL for data management, and transformer-based machine learning models for intelligence features, LearnBuddy offers a scalable, practical, and inclusive solution for modern education. By combining adaptive learning, real-time feedback, and AI-powered support, the system aims to promote personalized, fair, and effective learning for students of all abilities.

Keywords: LearnBuddy, Adaptive Learning, Intelligent Tutoring System, Machine Learning, Dynamic Quiz Generation, Personalized Chatbot, Difficulty Prediction, Educational Technology, Flutter, FastAPI, PostgreSQL, Student Performance Assessment.

Table of Contents

List of Figures

Acronyms/Abbreviations

- AI — Artificial Intelligence
- ML — Machine Learning
- UI — User Interface
- API — Application Programming Interface
- REST — Representational State Transfer
- GPU — Graphics Processing Unit
- SDK — Software Development Kit

Introduction

The increasing use of digital education has created a need for learning systems that do more than just deliver content—they must also adapt to each student's individual learning style. Traditional e-learning platforms often follow a one-size-fits-all approach, which can leave students with different abilities struggling to keep up. Advances in artificial intelligence (AI) and machine learning now make it possible to analyze a student's performance in real time and adjust learning materials accordingly. LearnBuddy uses these technologies to provide a personalized learning experience that grows and adapts with each learner.

By combining adaptive quizzes, instant feedback, and an AI-powered chatbot in a mobile-friendly platform, LearnBuddy aims to bridge the gap between traditional classroom learning and modern personalized education. The system helps students understand school-level concepts more effectively while demonstrating how AI can be applied to real-world learning challenges.

1.1 Background

Digital education has made it easier for students to access learning materials anytime and anywhere. However, most current systems provide the same content to all students, offering little personalization. As a result, learners often face challenges such as quizzes that are too easy or too difficult, limited guidance, and minimal feedback on their progress.

LearnBuddy addresses these challenges by creating an intelligent learning platform that adapts to each student's performance. The system evaluates a student's skill level using machine learning and generates quiz questions that match their ability. Additionally, a built-in AI chatbot provides explanations and answers questions in real time, making the learning experience more interactive, engaging, and tailored to individual needs.

1.2 Objectives

The primary goal of LearnBuddy is to create a learning system that adapts to each student's abilities, providing personalized support and fostering better understanding of school-level concepts. By combining adaptive quizzes, AI-driven explanations, and real-time performance tracking, the system aims to enhance learning outcomes, improve engagement, and ensure that students of all abilities have access to high-quality educational tools. The specific objectives of the project are outlined below:

- Develop a mobile-first adaptive learning application using Flutter.
- Dynamically generate quiz questions based on the student's skill level.
- Integrate machine learning models for difficulty prediction and question creation.
- Implement a chatbot that provides explanations, solutions, and answers to student queries.
- Design a robust backend using FastAPI and PostgreSQL.
- Continuously monitor and track student progress and improvement.

1.3 Motivation and Significance

Students often require personalized support that traditional classrooms cannot always provide. Intelligent tutoring systems improve learning outcomes by adapting content to each learner's pace. LearnBuddy is significant because:

- It supports self-paced learning for school students.
- It demonstrates the integration of mobile development, backend engineering, and machine learning.
- It provides a practical and scalable educational technology solution.
- It serves as a strong academic project showcasing software engineering and AI skills.

1.4 Expected Outputs

The expected outcomes from this project are described below:

- A functional Flutter mobile and web application.
- A quiz module that generates questions dynamically according to student ability.
- A trained ML model for difficulty assessment and question generation.
- A chatbot for clearing doubts and providing explanations.
- A FastAPI backend with RESTful APIs.
- A PostgreSQL database for storing courses, quizzes, and user performance.
- Documentation, testing reports, and system design diagrams.

Related Works

1. mySecondTeacher :

mySecondTeacher is an e-learning platform aligned with the Nepalese SEE and NEB curriculum, offering 4,500+ interactive video lessons, eBooks, and custom test generation. Key features include diagnostic assessments, virtual classrooms, and dashboards for students, parents, teachers, and school leaders.

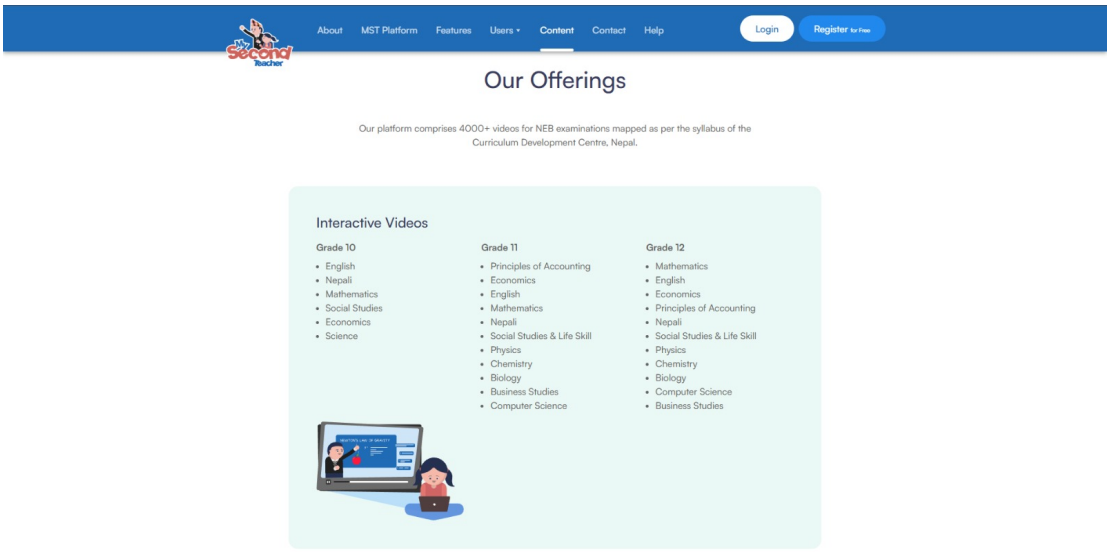

Figure 2.1: mySecondTeacher

2. Mero School:

Nepal's largest online learning platform, offering video based courses for Grade 1–12, engineering entrance, language learning, and skill development. It provides 24/7 access, downloadable content for offline viewing, and a mix of free trial and paid plans.

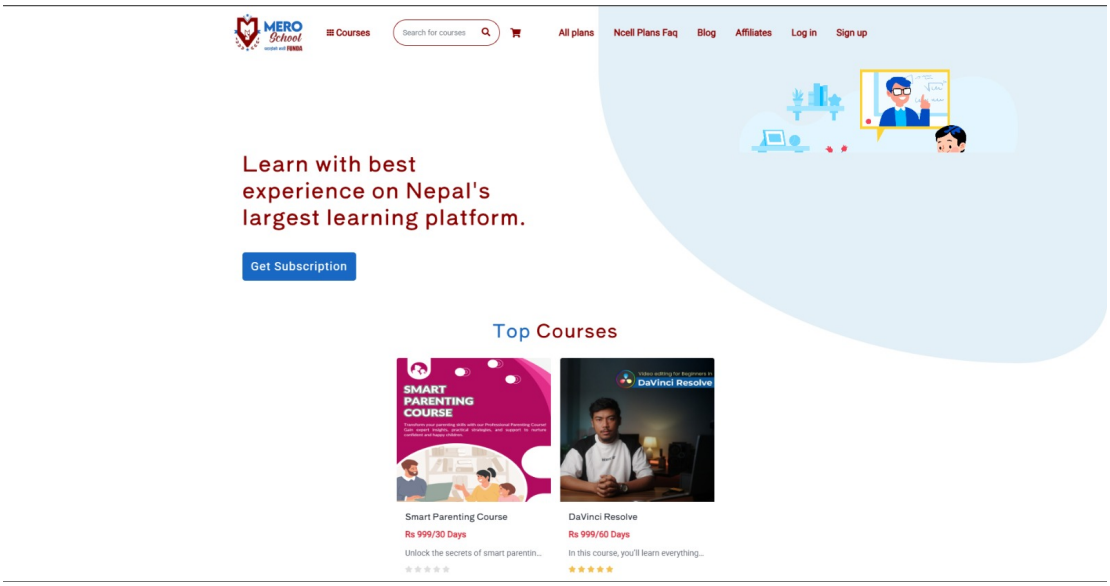

Figure 2.2: Mero School

3. MeroSiksha:

MeroSiksha is a personalized digital-learning app for students from school to university, offering over 10,000 notes, 9,000+ videos, 200,000+ quizzes, past papers, flashcards, and more plus instant tutor support.

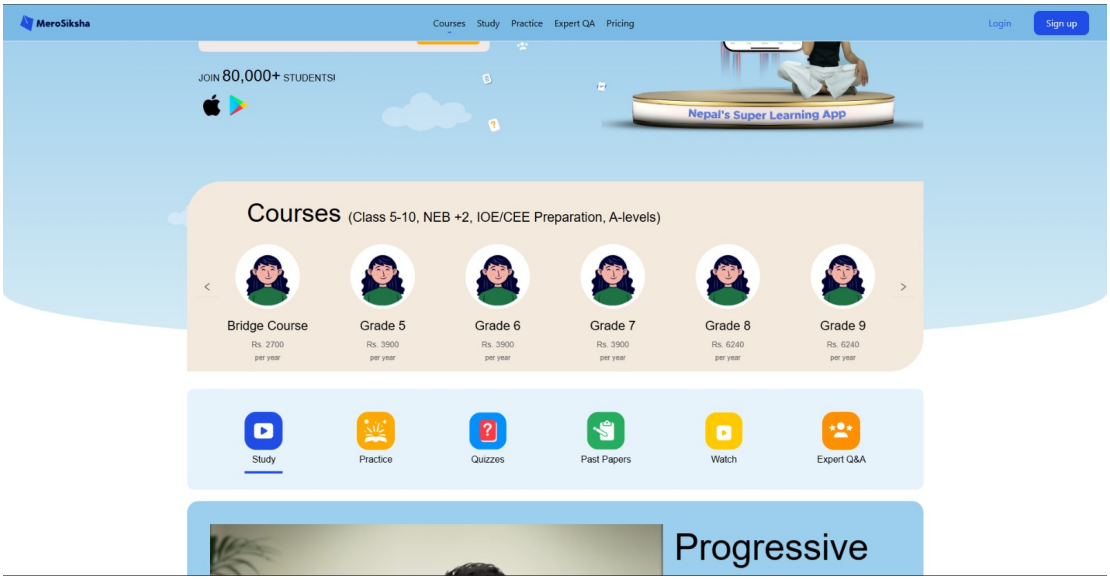

Figure 2.3: MeroSiksha

Procedure and Methods

3.1 Requirement Analysis and Planning

Activities include:

- Identifying ML capabilities needed: difficulty prediction, question generation
- Gathering requirements for user flow
- Planning API endpoints for quizzes, chatbot, course data, and user tracking
- Selecting the tech stack: Flutter, FastAPI, PostgreSQL, ML models
- Preparing UML diagrams, workflow diagrams, and data models

3.2 Frontend Development

The frontend will be developed using Flutter with a single codebase for Android and web. Key features:

- User onboarding and course selection UI
- Chapter navigation
- Adaptive quiz interface with timer and scoring
- Chatbot interaction screen
- User performance dashboard

3.3 Backend Development

FastAPI serves as the backend engine responsible for business logic and secure communication:

- Creating RESTful endpoints for user authentication, quiz generation, difficulty prediction, chatbot interaction, and performance tracking.

- Handling quiz generation requests by coordinating between the frontend, database, and ML module.
- Incorporating clean architecture for scalability and maintainability.
- Logging system events, validating incoming data, and managing asynchronous request handling for efficiency.

3.4 Database Design

The LearnBuddy database will be built using PostgreSQL to manage all core relational data efficiently. The schema will include tables for users, courses, chapters, quiz questions, quiz results, and skill levels, with foreign keys ensuring clear relationships and data consistency.

3.5 Machine Learning

LearnBuddy's ML module includes:

- Question Generation: FLAN-T5 or T5-small to generate quiz questions from chapter text.
- Difficulty Prediction: BERT or similar classifier to assess student intelligence based on quiz history.
- Chatbot System: Transformer-based model or API-based LLM to answer doubts. Optional RAG (Retrieval-Augmented Generation) using Sentence-BERT + Qdrant for chapter-specific answers.
- Training Evaluation: Fine-tuning models on sample school-level datasets, evaluating question quality, and measuring skill prediction accuracy.

3.6 Testing and Debugging

The testing and debugging phase ensures that all components of LearnBuddy function correctly and efficiently. The frontend will be tested to verify interface responsiveness, quiz behavior, and overall user experience. Backend APIs will be validated using tools such as Postman to confirm accurate data exchange and

3.7 Deployment

For academic demonstration, LearnBuddy will be deployed using a simple and reliable local setup. The Flutter application will run on mobile devices or through a web browser. The FastAPI backend will operate locally using Uvicorn, while PostgreSQL will run either locally or in a Docker container. This approach keeps deployment straightforward and ensures all components work smoothly without requiring complex hosting environments.

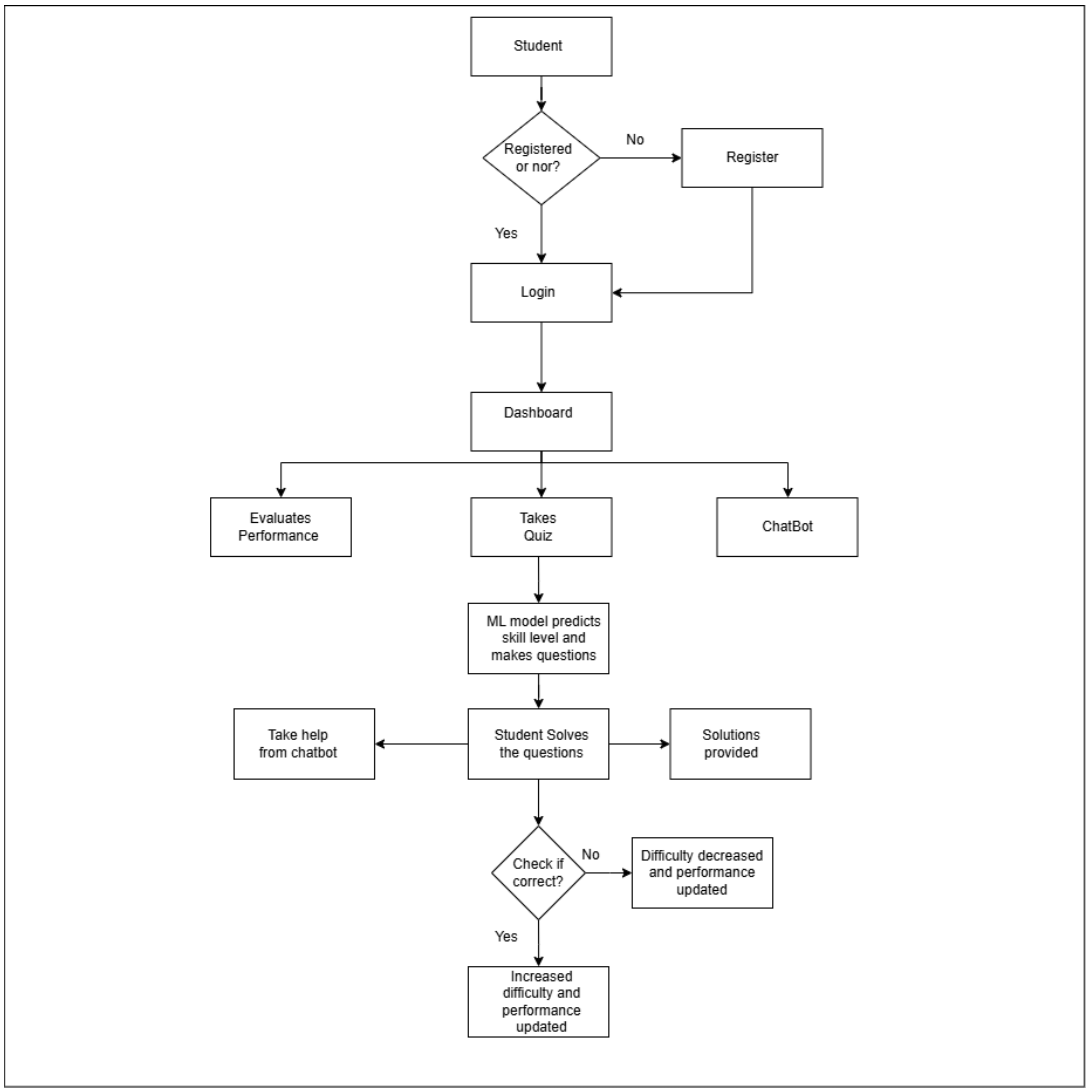

Figure 3.1: Workflow Diagram of LearnBuddy

System Requirement Specifications

4.1 Software Specifications

- Flutter SDK
- Android Studio or VS Code
- Python 3.10+
- FastAPI
- PostgreSQL
- SQLAlchemy
- PyTorch
- Git

4.2 Hardware Specifications

Mobile Device Requirements:

8.0 or higher / iOS 12.0 or higher

- Minimum 3 GB RAM
- Stable internet connection for backend and AI features

Development and Testing Device:

- Laptop or desktop with at least 8 GB RAM (16 GB recommended for ML tasks)
- Intel i5 / Ryzen 5 processor or higher
- Minimum 20 GB free storage
- GPU optional for accelerating ML tasks

Project Planning and Scheduling

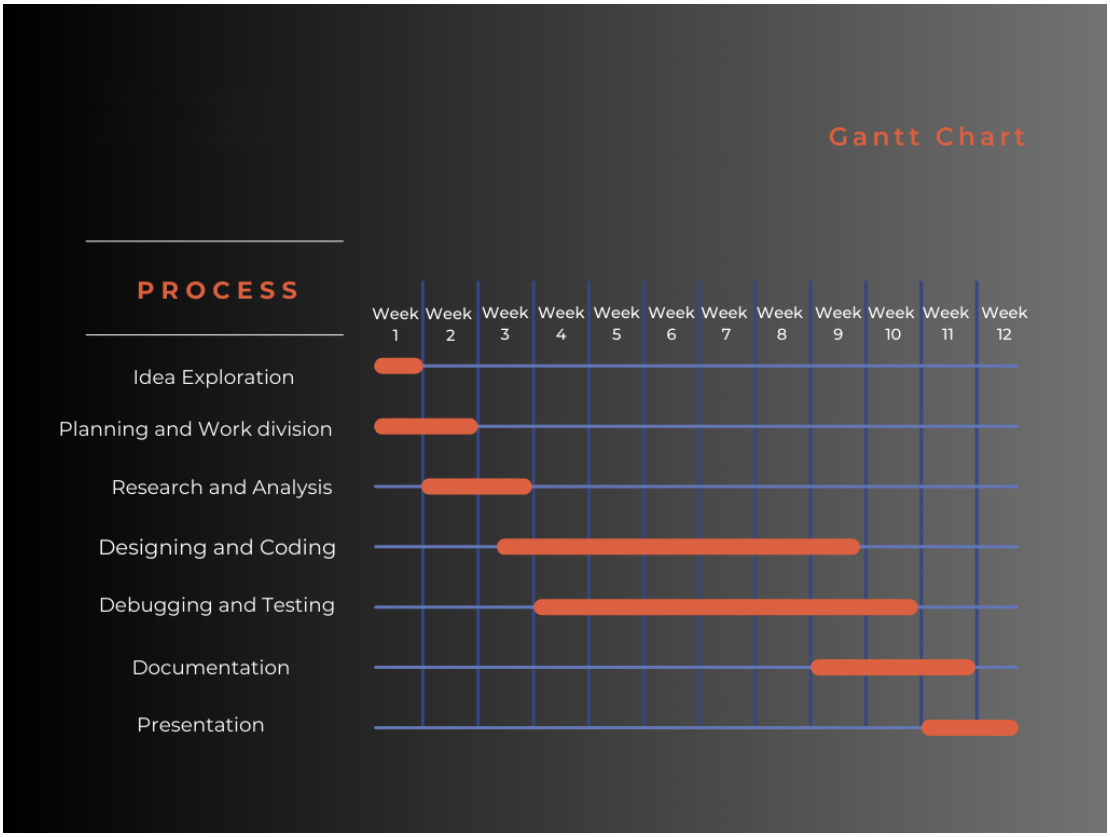

Figure 5.1: GANTT Chart

References

1. Attention Is All You Need," Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). NeurIPS. Retrieved from https://arxiv.org/abs/1706.03762
2. astAPI. (n.d.). FastAPI — Modern, fast (high-performance) web framework for building APIs with Python. Retrieved from https://fastapi.tiangolo.com/
3. PostgreSQL Global Development Group. (n.d.). PostgreSQL Documentation. Retrieved from https://www.postgresql.org/docs/
4. Liu, V., Latif, E., Zhai, X. (2025). Advancing Education through Tutoring Systems: A Systematic Literature Review. arXiv. Retrieved from https://arxiv.org/abs/2503.09748
5. Flutter. (n.d.). Flutter Documentation. Retrieved from https://docs.flutter.dev/