EDUCONNECT: AI-ENABLED PEER-TO-PEER LEARNING FRAMEWORK FOR UNIVERSITY STUDENTS

LYNN AMOIT

M23B38/007

HELLEN NOELINE NAMBOOZE

S23B38/014

RONALD TUSIIME

M23B38/017

A PROJECT REPORT SUBMITTED TO THE FACULTY OF ENGINEERING DESIGN, AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN DATA SCIENCE AND ANALYTICS OF UGANDA CHRISTIAN UNIVERSITY

June, 2026

UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

DECLARATION

We hereby declare that this project report titled **"EduConnect: AI-Enabled Peer-to-Peer Learning Framework for University Students"** is our original work developed for academic purposes as a Research Project. It has not been plagiarised and has not been submitted to any other institution for any academic award. All external sources, libraries and supporting research materials used in this report are formally acknowledged in the references section.

Amoit Lynn

Signature: .... Date: ....

Nambooze Hellen Noeline

Hellen. N.

Signature: .... Date: ....

Tusiime Ronald

Signature: .... Date: ....

ETHICAL GENERATIVE AI USAGE DECLARATION

We declare that any use of generative artificial intelligence (AI) tools during the development of this project has been conducted responsibly, transparently and ethically. AI was used only to assist with tasks such as drafting non-critical text, code scaffolding, and idea exploration. All AI-generated contributions were reviewed, validated and edited by the authors to ensure accuracy, originality, and compliance with academic integrity standards. No confidential or personally identifying data was provided to external AI services.

We acknowledge the limitations of current AI systems and commit to documenting AI-assisted steps in project notes and repository changelogs, crediting tools where appropriate and ensuring human oversight for all substantive content and decision-making.

Amoit Lynn

Signature: .... Date: ....

Nambooze Hellen Noeline

Hellen. N.

Signature: .... Date: ....

Tusiime Ronald

Signature: .... Date: ....

APPROVAL

This project report is submitted to the Department of Computing & Technology, Faculty of Engineering Design & Technology, Uganda Christian University, in partial fulfilment of the requirements for the award of the Bachelor of Science in Data Science and Analytics.

Supervisor 1: Dr. Samuel Kakuba

Signature: .... Date: 29/05/2026

Supervisor 2: Eng. Osolo Raymond

Signature: .... Date: 29 / 05 / 2026

ABSTRACT

EduConnect is a web-based, AI-enabled peer learning platform designed to revolutionize how university students discover study partners, access high quality academic resources, and sustain progress. The project addresses three critical local challenges: fragmented peer-learning coordination, limited personalisation in study support and weak integration between collaboration tools and progress tracking.

The implemented client-server architecture utilises a React frontend, Node.js/Express API and SQLite database persistence (`sql.js`) with optional Python services for advanced NLP intent classification and sequence-to-sequence quiz generation. EduConnect integrates intelligent peer recommendations, multi-discipline resource browsing, adaptive quiz workflows, study groups with direct and group messaging and a chatbot (EduBot) for intent based assistance. A key technical contribution is a trained **multinomial logistic regression model** for predicting next study topics based on user engagement statistics, trained in Python and exported to a JSON weight matrix for native JavaScript inference.

Pilot survey evidence from 50 students indicates strong adoption potential and problem relevance. The average student difficulty in finding study partners was rated at 3.80/5, and overall interest in the platform was 4.36/5, with 78% of respondents willing to join beta testing. Combined with a fully functional, deployed MVP, these findings support the technical viability and educational significance of the EduConnect framework.

Keywords: Peer learning, educational technology, recommendation systems, logistic regression, NLP intent classification, student collaboration.

DEDICATION

This report is dedicated to our lecturers, who constantly challenged us to push the boundaries of technology.

To our families, for their unwavering emotional and financial support throughout our academic journey.

And to our peers at Uganda Christian University, whose collaborative drive inspired the creation of EduConnect.

Amoit Lynn, Nambooze Hellen Noeline, Tusiime Ronald

Contents

List of Tables

List of Figures

LIST OF APPENDICES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
B2B	Business-to-Business
B2C	Business-to-Consumer
BSDS	Bachelor of Science in Data Science
CSV	Comma-Separated Values
DM	Direct Message
ETL	Extract, Transform, Load
LMS	Learning Management System
ML	Machine Learning
MVP	Minimum Viable Product
NLP	Natural Language Processing
SSO	Single Sign-On
UCU	Uganda Christian University
UI/UX	User Interface / User Experience

ACKNOWLEDGEMENTS

We express our profound gratitude to Uganda Christian University, the Faculty of Engineering Design & Technology, and the Department of Computing & Technology for providing the resources and environment that made this research possible. We are deeply indebted to our project supervisors and course instructors for their rigorous feedback, structural insights, and continuous guidance.

Special appreciation goes to the 50 university students who participated in our pilot survey and validation exercises. Their practical feedback on study partner matching, adaptive quizzes, and collaborative study plans directly shaped the EduConnect feature set and refined our user-facing micro-interactions.

Amoit Lynn, Nambooze Hellen Noeline, Tusiime Ronald
April 2026

CHAPTER 1: INTRODUCTION

1.1 Background to the Study

Digital learning solutions have expanded rapidly across higher education institutions in Sub-Saharan Africa. However, coordinating meaningful, collaborative peer learning support remains a localised challenge. In many university contexts, including Uganda Christian University (UCU), students rely heavily on informal fragmented channels such as WhatsApp groups or physical common rooms that do not support automated partner compatibility matching, progress tracking or structural learning guidance in a centralised environment.

Peer learning is well documented to improve academic performance and student retention. Yet when students are left to form groups manually, they tend to self select into homogeneous circles, leaving struggling students isolated and failing to optimise study partner matches based on complementary strengths, overlapping schedules, and shared academic tracks.

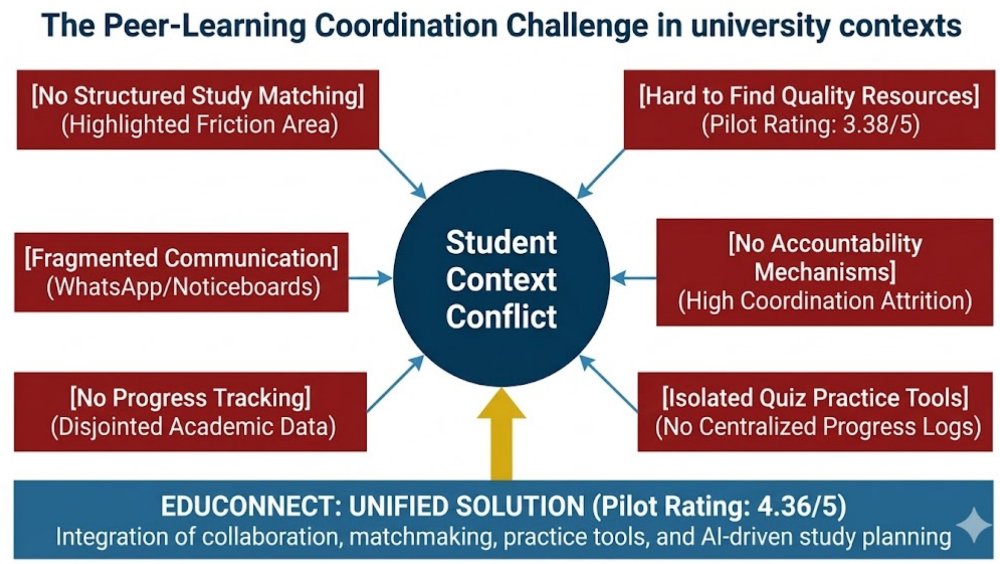

Figure 1.1: The peer-learning coordination challenge in university contexts.

1.2 Problem Statement

University students lack a unified, intelligent framework that connects them to complementary study partners, provides structured learning resources, supports practice based learning and guides their study pathways through data-driven recommendations. Current solutions are highly fragmented students must navigate separate, disjointed platforms for communication, content access and schedule tracking. This results in significant coordination overhead, weak accountability, and high attrition rates in self organised study cohorts.

1.3 Main Objective

To design, implement, and validate EduConnect, an AI-enabled peer-to-peer learning framework that optimises student collaboration, resource access, and adaptive progress tracking in university settings.

1.4 Specific Objectives

1. To develop a responsive web platform that manages detailed student profiles and handles peer-learning interactions.
2. To engineer a hybrid, explainable recommendation engine for study partner matching based on content similarity and engagement signals.
3. To build a multi-discipline catalogue of learning resources paired with an adaptive quiz subsystem.
4. To implement an NLP-driven conversational agent (EduBot) capable of classifying user intents and routing them to appropriate platform features.
5. To train and integrate a Machine Learning model (Multinomial Logistic Regression) that predicts and recommends the next best study topic based on individual performance and cohort engagement patterns.
6. To validate market relevance, problem severity, and feature interest using primary pilot survey data from 50 target users.

1.5 Research Questions

1. How can a unified, AI-enabled web platform with detailed student profiles and a hybrid, explainable recommendation engine improve the efficiency of peer matching and study group coordination in a local university environment?
2. How can a multi-discipline resource catalogue with adaptive quizzes be combined with a trained multinomial logistic regression model to provide personalised, data-driven recommendations for a student's next study topic?
3. How can an NLP-driven conversational agent (EduBot) be implemented to classify user intents and route them to appropriate features, and what is the level of demand, problem severity, and adoption willingness (validated through a pilot survey of 50 target users) for such an intelligent peer-learning framework?

1.6 Scope of the Study

The target audience for EduConnect is undergraduate students across multiple disciplines at Uganda Christian University (UCU).

The technical implementation consists of a deployed web minimum viable product (MVP) incorporating a React frontend, a Node.js/Express API, SQLite backend storage, and a Python-based training and inference pipeline for machine learning features.

The core modules implemented include user authentication, onboarding profile capture, hybrid peer matching, a learning resource library, adaptive quizzes with automated question generation, personalised weekly study plans, chat channels (both direct and group messaging), and a protected administrative control dashboard.

Features that are explicitly out of scope for this project are multi-institutional federated databases, full real-time socket-based messaging, institutional Single Sign-On (SSO) integration, and production-grade Learning Management System (LMS) synchronisation.

1.7 Significance of the Study

This study will deliver tangible benefits to three distinct but interconnected groups, university students who will gain an intelligent, personalised tool for collaborative learning, academic institutions that will acquire a low cost, scalable framework to foster peer led support and track student engagement and the research and technopreneurship community which will receive a practical, validated template for building and deploying educational technology start ups in resource constrained environments.

For Students: EduConnect provides an accessible, personalised academic companion that matches them with optimal study partners and dynamically points them to resources that address their individual academic weak spots.

For Academic Institutions: The study demonstrates a low-cost, scalable method to foster peer-led academic support, track student engagement and gather aggregate diagnostic insights into academic pain points.

For Research and Technopreneurship: EduConnect combines software engineering with applied machine learning and user-centred survey research, serving as a practical template for executing and validating educational technology start-ups in developing economies.

1.8 Rationale and Justification of the Study

This study was undertaken because existing peer-learning coordination tools available to UCU students are either entirely informal (WhatsApp groups, physical noticeboards) or are large-scale commercial platforms not optimised for the local academic context. No existing tool simultaneously addresses partner matching, resource discovery, adaptive practice, and study planning within a single, low-bandwidth-friendly environment. EduConnect directly fills this gap by engineering a locally relevant, data-driven solution validated through primary survey evidence from 50 active UCU students.

1.9 Alignment with Programme Specialization

Programme: Bachelor of Science in Data Science and Analytics
Specialization: Data Science

EduConnect directly aligns with the AI & ML specialization through the following components:

1. **Supervised Machine Learning:** A multinomial logistic regression model was trained to predict the next study topic based on student features (quiz scores, study hours, course area, weak/strong topics). The training pipeline uses scikit-learn with feature engineering (one-hot encoding, multi-label binarization, z-score standardization).
2. **Natural Language Processing (NLP):** A DistilBERT transformer model was fine-tuned for intent classification to power EduBot, demonstrating practical application of transformer architectures, attention mechanisms, and transfer learning.
3. **Model Deployment & MLOps:** The trained logistic regression model was exported to JSON and inference re-implemented in pure JavaScript (logit calculation + softmax), demonstrating practical model deployment in resource-constrained environments without Python runtime dependencies.
4. **Data Engineering:** Custom ETL pipelines (`scripts/data_pipeline.py`) were built to extract student data from SQLite, perform standardization and encoding, and load feature matrices for model training.
5. **Evaluation & Validation:** Model performance was rigorously evaluated using confusion matrices, accuracy metrics, training/validation loss curves, and pilot survey statistical analysis (means, percentages, n=50).

1.10 Alignment with Sustainable Development Goals (SDG)

EduConnect contributes primarily to **SDG 4: Quality Education**:

- **Target 4.4:** Increase the number of youth with relevant technical and vocational skills. EduConnect exposes students to AI/ML tools (recommenders, chatbots) within an educational context, building digital literacy.
- **Target 4.a:** Build and upgrade education facilities that are child-, disability- and gender-sensitive. EduConnect includes built-in accessibility features: text-to-speech engine, multiple colour contrast modes, and dark/light theme support.
- **Target 4.c:** EduConnect reduces the burden on faculty by automating study partner matching, resource recommendation, and quiz generation, allowing instructors to focus on high-value teaching interactions.

Secondary contributions include **SDG 9** (Industry, Innovation, Infrastructure) through locally relevant EdTech innovation, and **SDG 10** (Reduced Inequalities) by providing free access to study partner matching, resources, and practice tools to all students regardless of their peer networks.

1.11 Alignment with National Development Plan IV and Uganda Vision 2040

NDP IV (2025/26–2029/30) focuses on industrialisation, human capital development, and innovation. EduConnect aligns with **Programme 7: Human Capital Development** by supporting digital learning continuity, and **Programme 8: Innovation, Technology Transfer & Digital Transformation** as an indigenous EdTech innovation developed by Ugandan students.

Uganda Vision 2040 aims to transform Uganda into a modern and prosperous country. EduConnect contributes to the **Science, Technology, Engineering and Innovation (STEI)** pillar, **Human Resource Development**, and the broader goal of a **knowledge-based economy** by producing graduates who are not just consumers but creators of digital solutions.

1.12 Summary of Chapters

Chapter One presents the background, problem statement, objectives, research questions, scope, significance, and rationale of the study.

Chapter Two reviews the relevant theoretical and empirical literature underpinning peer learning, personalised recommendations, and NLP-driven educational tools, and presents the conceptual framework.

Chapter Three describes the applied research design, data sources, development methodology, tools and technologies, and ethical considerations employed in building EduConnect.

Chapter Four details the full system architecture, all implemented functional modules, the database schema, the core user flow, and the complete API and frontend route surfaces.

Chapter Five presents and explains the six intelligent models embedded in EduConnect, covering their training pipelines, inference mechanisms, and mathematical formulations.

Chapter Six presents results from model evaluation, the pilot survey, and the deployed MVP, followed by interpretation of findings, technical strengths, system limitations, and completeness findings.

Chapter Seven outlines the EduConnect business model, value proposition, customer segments, revenue strategy, and go-to-market plan.

Chapter Eight draws conclusions from the study, presents recommendations for platform evolution and institutional scaling, and reflects on lessons learned.

CHAPTER 2: LITERATURE REVIEW AND CONCEPTUAL FOUNDATION

2.1 Introduction

This chapter reviews the theoretical and empirical literature that guided the design of EduConnect. It examines foundational theories of peer learning and personalised recommendation, surveys empirical studies in collaborative educational technology, and presents the conceptual framework developed to guide the study. The chapter concludes by identifying the research gap that EduConnect addresses.

2.2 Theoretical Literature Review

2.2.1 Peer Learning and Collaborative Outcomes

Educational theory emphasises that peer-to-peer collaboration significantly enhances cognitive retention, critical thinking, and emotional resilience in higher education. Vygotsky's "Zone of Proximal Development" suggests that learning is optimised when students collaborate with peers who possess slightly more advanced skills in specific areas [1]. EduConnect operationalises this by matching students who have complementary strengths and weaknesses for example, pairing a student strong in Linear Algebra and weak in Python with another who has the inverse profile.

Social constructivism, as elaborated by Bruner, further establishes that knowledge is most effectively acquired through active participation and dialogue with peers rather than passive reception of information [2]. This theoretical position directly underpins EduConnect's study group, messaging, and quiz modules, which are designed to encourage active, collaborative knowledge construction rather than solo study.

2.2.2 Personalised Learning and Recommendation Theory

Traditional recommender systems rely heavily on massive user datasets (collaborative filtering), which are prone to the "cold-start" problem in new or small-scale platforms. Content-based filtering, which matches users based on profiles and item attributes, is highly effective for educational contexts. EduConnect utilises a hybrid recommender system that calculates a compatibility score combining content similarity (interests, course, weak and strong topics), temporal availability (overlapping study hours), and peer engagement signals (average response rates, meeting attendance).

Cognitive load theory further informs EduConnect's interface design: by surfacing only relevant partners, resources, and quiz items rather than overwhelming students with unfiltered content, the platform reduces extraneous cognitive load and directs mental effort toward productive learning activity [3].

2.2.3 Natural Language Processing in Educational Chatbots

The application of transformer-based language models to educational chatbots has grown substantially since the introduction of BERT [4]. Lightweight distillation models such as DistilBERT retain approximately 97% of BERT's language understanding capability at 60% of the parameter count, making them suitable for deployment in resource-constrained environments [5]. EduBot's primary intent classifier is built on this foundation, fine-tuned on domain-specific EduConnect intent data.

2.3 Empirical Literature Review

2.3.1 Collaborative Learning Frameworks and Grouping Models

Wang et al. [6] proposed a genetic algorithm-based grouping model integrated with cognitive diagnosis, demonstrating that algorithmically formed study groups outperform self-selected groups on knowledge retention metrics. Their finding that heterogeneous grouping produces stronger outcomes directly validates EduConnect's approach of pairing students with complementary, rather than identical, academic profiles.

Zou, Xie, and Wang [7] found that technology-enhanced peer feedback significantly improved collaborative writing quality and critical thinking tendency in higher education students. Their study provides empirical grounding for EduConnect's feedback and study-group messaging modules.

2.3.2 Personalised Recommendation in Education

Ali and Shafi [8] demonstrated that hybrid collaborative and content-based filtering, enhanced with optimised RoBERTa embeddings, substantially outperforms single-approach recommenders in course recommendation tasks. Sharma and Sharma [9] similarly showed that clustering algorithms applied to student profile data yield more accurate and accepted personalised learning recommendations. Both studies validate EduConnect's hybrid matching architecture.

Ouyang et al. [10] examined instructor-student collaborative partnerships in online learning communities, finding that structured, algorithmically mediated introductions between students led to more sustained engagement than unstructured open-access communities. This supports EduConnect's explainable match recommendation, which provides specific textual reasons for each peer suggestion.

2.3.3 NLP and Intent Classification for E-Learning

Al-Badi et al. [11] applied NLP methods to identify barriers affecting e-learning adoption, demonstrating that intent-aware language models provide richer insights into learner friction than keyword-based tools. Abidi et al. [12] developed a hybrid deep learning model for e-learning system prediction, achieving high classification accuracy on student query intents. These empirical results directly informed EduBot's dual-engine design: a fine-tuned transformer as the primary classifier with a cosine-similarity embedding fallback.

Ansari and Khan [13] explored the role of social media in collaborative learning, finding that students who had structured digital touchpoints with peers reported higher academic motivation and reduced isolation. This finding underpins EduConnect's integrated direct and group messaging modules.

2.4 Conceptual Framework

The EduConnect conceptual framework comprises four distinct layers:

1. **Input Layer:** Captures student registration attributes, academic tracks, multi-select skill matrices, scheduling availability, and real-time quiz attempts.
2. **Intelligence Layer:** Hosts the hybrid peer-matching algorithm, the cohort analytics engine, the scikit-learn Logistic Regression next-topic model, and the EduBot NLP intent classifier.
3. **Interaction Layer:** Renders the customised React dashboard, study timers, group chat spaces, automated quiz interfaces, and administrative reporting dashboards.
4. **Outcome Layer:** Measures increased study hour consistency, improved resource discovery speed, highly aligned study partner pairings, and elevated exam confidence.

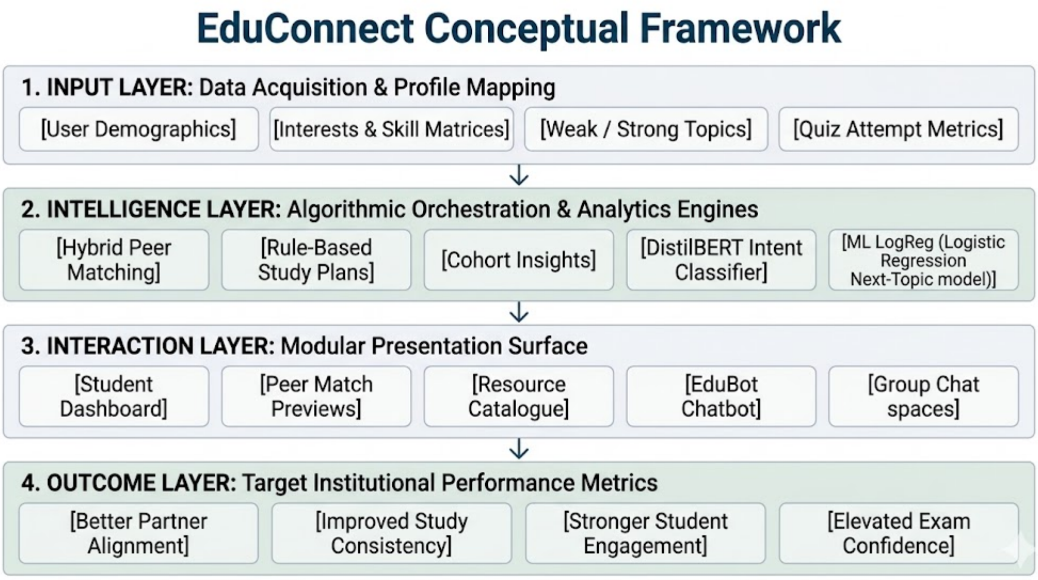

Figure 2.1: EduConnect conceptual framework.

2.5 Research Gap Addressed

Most educational platforms are siloed; they focus exclusively on course content delivery (e.g., Moodle), communication (e.g., Slack), or isolated flashcards (e.g., Quizlet). The empirical literature reviewed above consistently highlights the need for unified platforms that combine algorithmic partner matching, adaptive practice, and structured communication within a single environment. EduConnect addresses this structural gap by unifying collaboration, automated matchmaking, practice tools, and AI-driven study planning into a single, cohesive, client-server web application optimised for low-bandwidth environments common in regional institutions.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter describes the applied research design, data sources, development methodology, tools and technologies, and ethical considerations employed in building EduConnect. The methodology follows an exploratory and descriptive approach: exploratory in identifying the gap in peer-learning coordination, and descriptive in systematically documenting the architecture, models, and survey outcomes of the implemented solution.

3.2 Research Design

This project adopted a **design-and-build applied research methodology** combined with quantitative empirical validation which was executed through five sequential phases: requirements synthesis, system architecture planning, iterative implementation (prototyping), model integration and a pilot empirical survey.

1. **Requirements Synthesis:** Analysed academic study friction points among UCU undergraduate cohorts.
2. **System Architecture Planning:** Framed the data schemas, modular API endpoints, and interface mockups.
3. **Iterative Implementation (Prototyping):** Built the database tables, API routes, and React components, testing each module locally using live mock data.
4. **Model Integration:** Trained the Machine Learning next-topic model in Python, exported model parameters to JSON, and deployed a lightweight Node.js engine for client-side inference.
5. **Pilot Empirical Survey:** Deployed the functional web platform and administered a validation survey to 50 active students to collect quantitative demand and interest metrics.

3.3 Area of Study

The study was conducted at Uganda Christian University (UCU), Mukono, Uganda. UCU was selected because it represents a typical regional university environment with diverse academic faculties, a student population relying on informal peer coordination, and constrained digital infrastructure — precisely the conditions EduConnect is designed to serve.

3.4 Sources of Information

1. **Primary Platform Data:** Session interactions, user profiles, matching requests, and group message logs persisted in `sql.js`.

2. **Pilot Survey Dataset:** A CSV corpus (`pilot_survey.csv`) containing responses from 50 UCU undergraduates detailing resource difficulty, matching difficulty, preferred platform, and feature interest ratings.
3. **Reference Literature:** The UCU Academic Research Manual [14], technopreneurship business canvases, and historical scikit-learn model evaluation metrics [15].

3.5 Population and Sampling

The target population was undergraduate students across all faculties at UCU. A purposive sampling technique was employed to select 50 participants who were actively enrolled in degree programmes, had prior experience using digital study tools, and were willing to evaluate the EduConnect platform. Purposive sampling was appropriate given the exploratory nature of the validation exercise.

3.6 Variables and Measurements

The key variables measured in the pilot survey were: difficulty finding study partners (1–5 Likert scale), difficulty finding learning resources (1–5 Likert scale), interest in peer matching (1–5 Likert scale), interest in AI quiz support (1–5 Likert scale), interest in study planning (1–5 Likert scale), overall platform interest (1–5 Likert scale), platform preference (categorical: Web or Mobile), and willingness to beta test (binary: Yes or No).

For the NLP intent classifier, the primary performance variable was classification accuracy across six intent categories, measured on a held-out validation set.

3.7 Development Method

An Agile Scrum approach was utilised to execute the software lifecycle:

- Sprints of 1–2 weeks focused on specific functional units (e.g., profile intelligence, followed by recommendations, and then messaging).
- Rigorous integration testing checked API boundaries between Express controllers, the SQLite data helper, and React contextual providers.

3.8 Data Collection Instruments

The primary data collection instrument was a structured digital survey administered via Google Forms. The survey instrument comprised Likert-scale rating questions, binary preference questions, and one open-ended feedback field. For the model evaluation component, a held-out test split of the intent classification dataset served as the quantitative instrument for measuring classifier accuracy and confusion.

3.9 Tools and Technologies

- **Frontend Stack:** React, Vite, Tailwind CSS, Context API for state management.
- **Backend Stack:** Node.js, Express, RESTful routing, JWT authentication.
- **Data Persistence:** SQLite (via `sql.js` locally and file-based storage server-side).
- **ML & NLP Pipeline:** Python (version 3.10+), `scikit-learn` [15], `pandas`, `pypdf` for technical document analysis, and Flask for advanced microservices.
- **Deployment Environments:** Netlify (for client static hosting), Render (for server API deployment).

3.10 Quality and Error Control

To ensure reliability and validity of the survey instrument, the questionnaire was reviewed by both project supervisors before distribution. Pilot responses were checked for internal consistency using mean and standard deviation analysis. For the machine learning models, standard train-validation splits (80/20) were applied, and model performance was assessed using confusion matrices and accuracy scores to verify that results were not attributable to class imbalance or overfitting.

3.11 Data Processing and Analysis

Survey data was exported from Google Forms as a CSV file and analysed using descriptive statistics (means and percentages) to summarise student interest and difficulty scores. Machine learning model outputs were evaluated using confusion matrices and validation accuracy metrics computed in Python with `scikit-learn` [15]. NLP intent classification results were visualised as confusion matrix heatmaps to compare baseline and fine-tuned model performance.

3.12 Ethical Considerations

- All pilot survey participants provided informed consent before submitting data.
- No personally identifiable student records are exposed outside the application's secure admin panels.
- All model recommendations are purely advisory and explainable to prevent automated academic stereotyping.
- Participation in the pilot survey was entirely voluntary, and respondents were informed of their right to withdraw at any time without consequence.

3.13 Methodological Constraints

The primary methodological constraint is the limited sample size of 50 students, which, while sufficient for a pilot validation, restricts the generalisability of the survey findings to the broader UCU student population. Additionally, the NLP intent classifier was trained on general-purpose conversational intent data from Hugging Face prior to fine-tuning, meaning that the initial pre-training vocabulary may not perfectly mirror EduConnect-specific student query patterns. The SQLite database also imposes single-writer concurrency limits that constrain the platform's scalability beyond the current pilot deployment.

3.14 Strict Full-Repository Audit Procedure

To achieve institutional compliance and resolve technical debt, we performed a thorough repository audit encompassing:

- Deep code scans of client source files (`src/*`,the React user interface), server controllers (`server/*`, the Node.js API and business logic), and optional Python services (`backend/*`, the NLP and ML training microservices).
- Identification and resolution of document-code drift between initial design documents and the actual production implementation.

CHAPTER 4: SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Architecture

EduConnect operates on a highly responsive Client-Server architectural model with four primary components:

1. React UI for student interactions (Dashboard, Recommendations, Quizzes).
2. Express API Server for business logic, authentication, study plan, and quiz generation.
3. SQLite Database for users, feedback, and activity persistence.
4. Python Backend for NLP intent classification and ML training services.

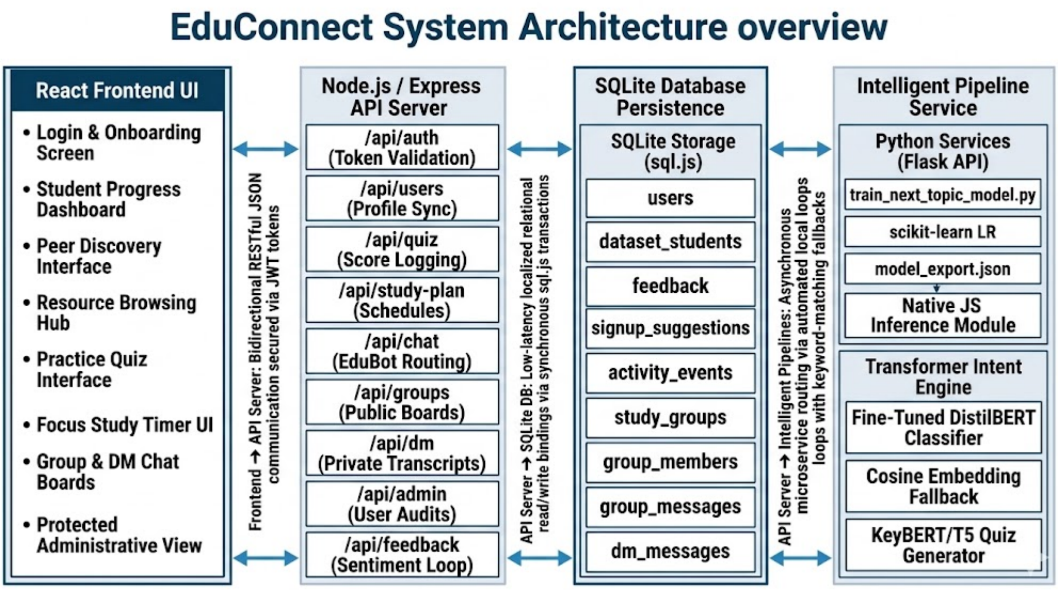

Figure 4.1: EduConnect system architecture overview.

4.2 Functional Modules Implemented

EduConnect currently implements the following complete module set:

4.2.1 Authentication and Session Security

Secure, token-based signup and login flows with role-based routing protecting administrative and standard student components. Session restoration and protected route guards are implemented. A default admin bootstrap facilitates system management.

4.2.2 Onboarding and Suggestion Engine

Rich profile builder displaying autocomplete dropdown options populated from aggregate student values (`signup_suggestions`). Dynamic onboarding covers personal, academic, skills, interests and weak and strong topic preferences.

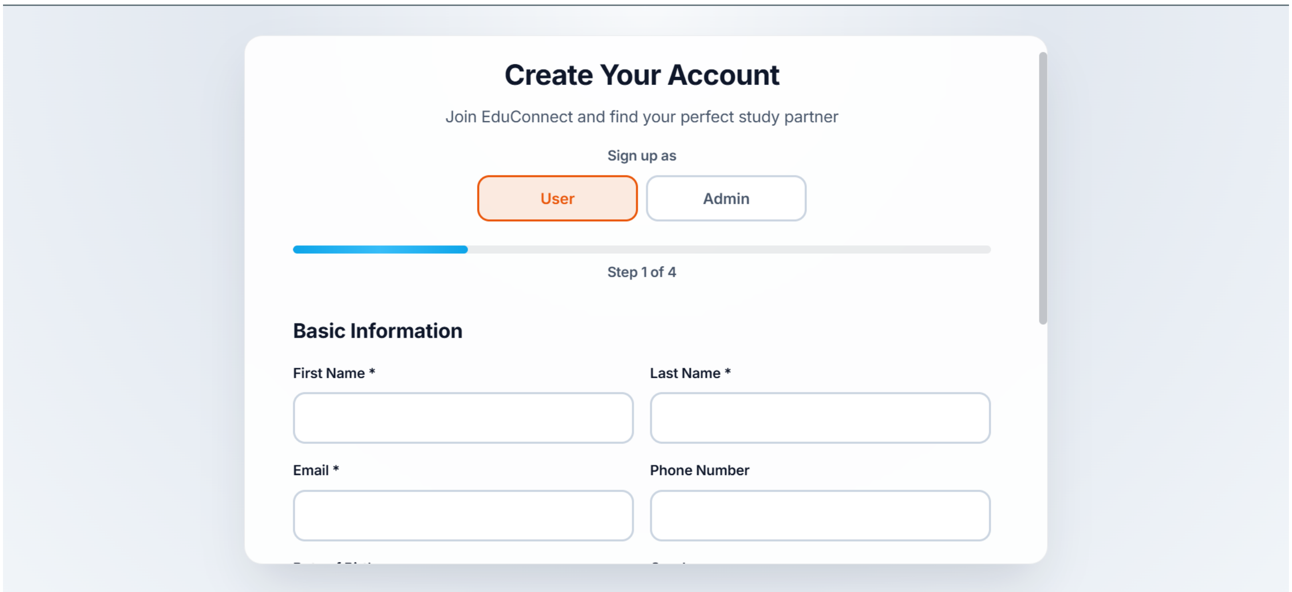

Figure 4.2: Student profile and onboarding interface.

4.2.3 *Productivity Workspace*

Focus study timer featuring automated idle detection, pause states, and integrated session logging connected to weekly user analytics charts. Personalised dashboard with cards, charts, recommendations preview and study plan are included.

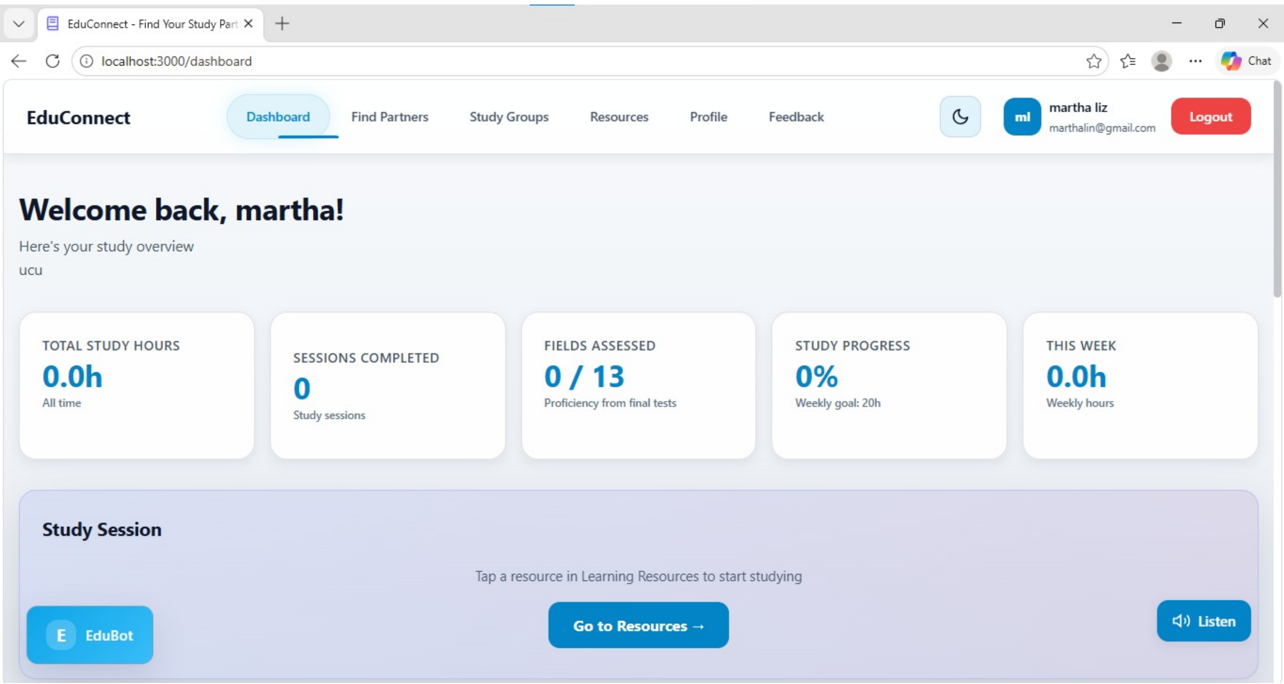

Figure 4.3: EduConnect student dashboard.

4.2.4 *Explainable Hybrid Study Partner Recommendation*

A profile-matching pipeline that calculates compatibility percentages based on skill overlaps, available days and response reliability, displaying specific reasons for each match. Dataset and live user merging with deduplication safeguards are implemented.

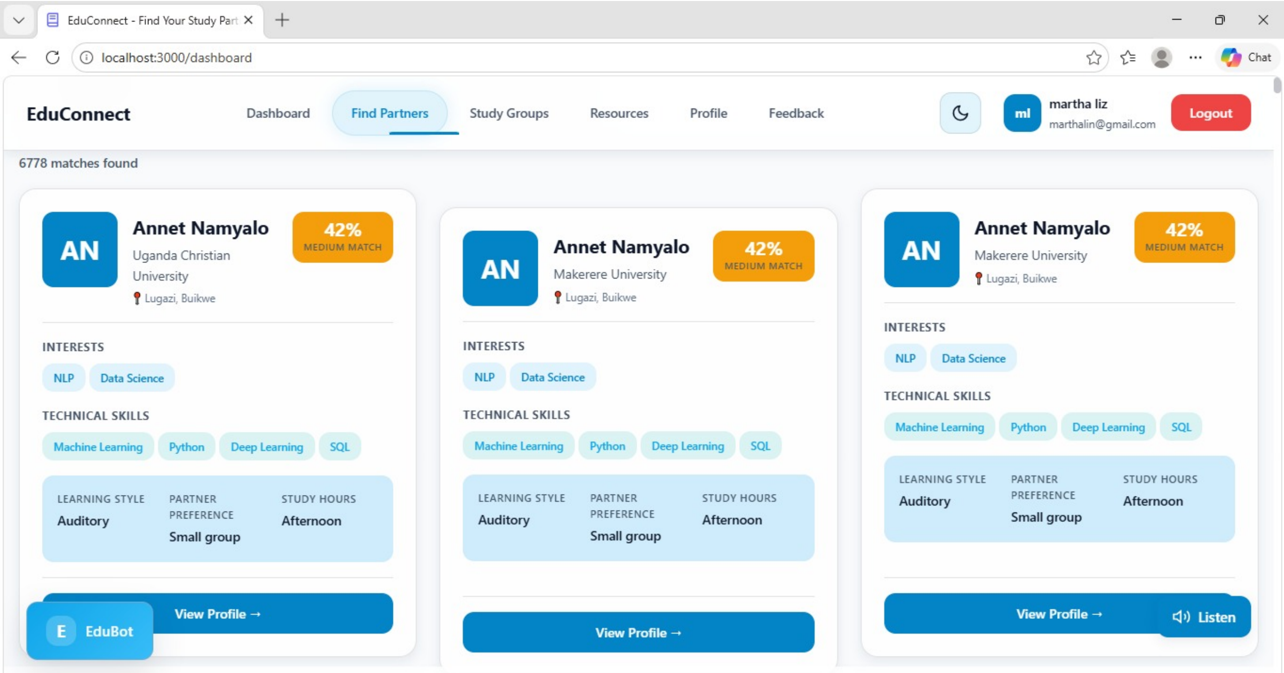

Figure 4.4: Study partner recommendation interface with explainable scores.

4.2.5 Multi-Discipline Resource Hub

Organised library divided into subject areas (Computing, Law, Business, Education, Humanities, Health, Agriculture, and others) utilising a global `LearningFieldsContext` to synchronise curriculum domains across the application. Category-aware browsing and external resource linking are supported.

4.2.6 Adaptive Quiz Subsystem

Supports pre-loaded subject quizzes and features dynamic server-side generation utilising resource context when a subject lacks manual test content. Admin-triggered full-field generation and persistent generated fields via `server/data/generatedQuizzes.json` are included.

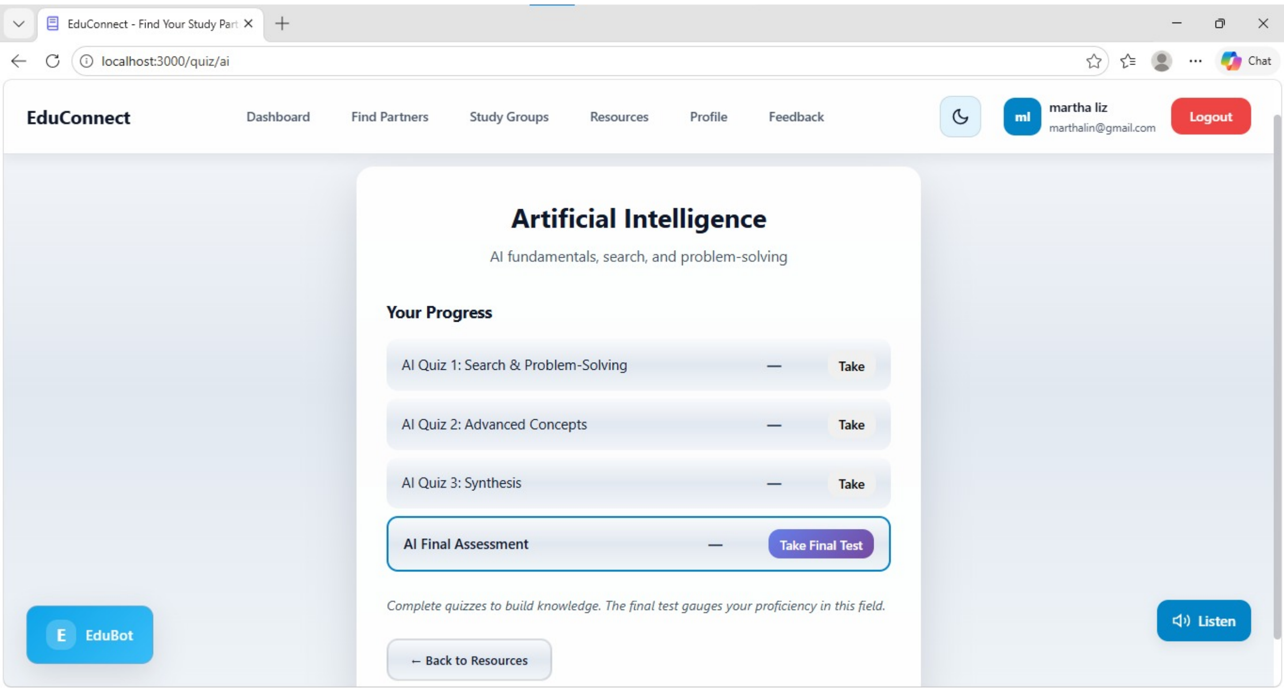

Figure 4.5: Quiz subsystem interface.

4.2.7 Study Plan and Learning Guidance

Instantly generates customisable weekly study schedules incorporating peer comparison graphs and ML topic highlights. Progress-aware suggestions with weak-topic emphasis and cohort-aware recommendations are included.

4.2.8 EduBot Intelligent Assistant

A text chatbot combining fast keyword intent routing with optional Python NLP semantic matching. Intent-based support for navigation and troubleshooting, resource recommendation by topic keywords and profile context, and optional LLM fallback for out-of-scope queries are all implemented.

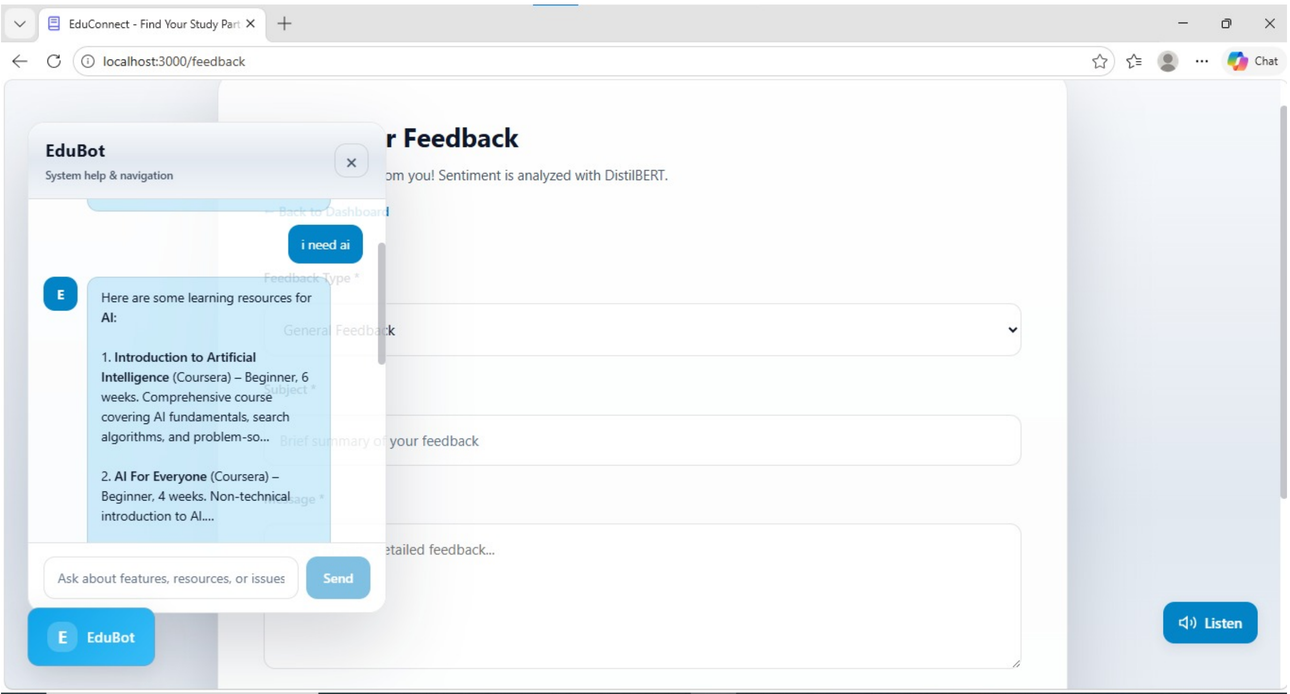

Figure 4.6: EduBot intelligent assistant interface.

4.2.9 Collaboration and Communication

Allows users to create and join public or private study groups and exchange messages via direct and group chat boards. Event logging supports behaviour-informed suggestions.

4.2.10 Feedback and Quality Loop

Collects explicit student suggestions and ratings, applying sentiment analysis to alert admins of user experience friction. Admin response workflows and optional email notification integration are provided.

4.2.11 Admin Control Dashboard

A central view for faculty to manage user lists, trigger quiz generation runs, and review platform feedback scores. Protected admin-only routes cover `/admin/dashboard`, `/admin/users`, `/admin/assessments`, and `/admin/quiz-generate`.

4.2.12 Accessibility and UX Support

Built-in text-to-speech engine to read study materials aloud, supporting multiple colour contrast modes. Theme support for light and dark modes, welcome messaging, and guided quick-access navigation are provided.

4.3 Database and Data Model

The relational SQLite schema contains the following production tables:

- `users`: Core login credentials, comprehensive academic profiles, and serialised `study_stats` JSON data.
- `dataset_students`: Reference student profiles imported for cohort matching and baseline recommendations.
- `feedback`: Student feedback submissions, sentiment categories, and admin text responses.
- `signup_suggestions`: Autocomplete dictionary entries for profile registration options.
- `activity_events`: Audit logs of user achievements (e.g., completing quizzes, joining study groups).
- `study_groups`, `group_members`, `group_messages`: Study group properties and chat histories.
- `dm_messages`: Persisted direct communication transcripts between matched partners.

4.4 Core User Flow

1. User signs up and completes profile and preferences.
2. Dashboard loads progress and recommendations.
3. User discovers peers and joins or creates groups.
4. User accesses resources and takes quizzes.
5. Study plan and model suggestions update guidance.
6. User interacts with EduBot for support and routing.

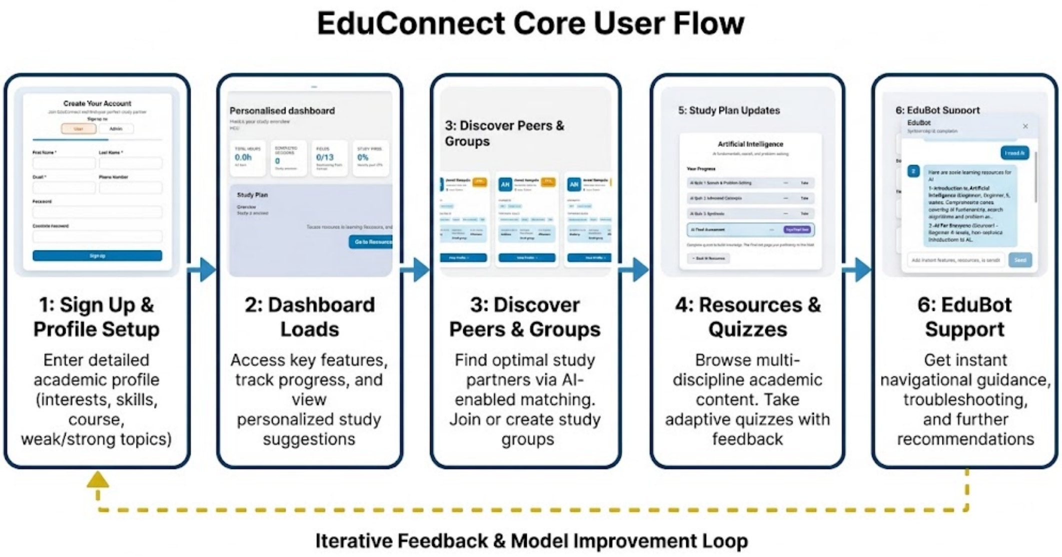

Figure 4.7: EduConnect core user flow.

4.5 Backend API Surface

The Node.js server exposes the following route groups:

- `/api/auth`: Handles token verification, signup, and login validation.

- `/api/users`: Manages profile retrieval, editing, and similarity pairing indices.
- `/api/quiz`: Manages quiz retrieval, scoring records, and automated question generation.
- `/api/study-plan`: Formulates customised schedules utilising user statistics and cohort averages.
- `/api/chat`: Receives chatbot user strings and runs intent classification workflows.
- `/api/groups`: Exposes controllers to create groups, list active members, and post messages.
- `/api/dm`: Manages private peer chat operations.
- `/api/admin`: Restricts access to administrative user-management utilities.
- `/api/feedback`: User feedback submission and management.
- `/api/activity`: Activity event logging.
- `/api/health`: API service health endpoint.

4.6 Frontend Route Surface

The React application includes the following routes:

- **Public Routes:** `/login`, `/signup`.
- **Student Interface:** `/dashboard`, `/profile`, `/recommendations`, `/resources`, `/analytics`.
- **Learning and Practice:** `/quiz/:fieldId`, `/quiz/:fieldId/:quizId`, `/study-plan`.
- **Messaging Interface:** `/groups`, `/groups/chat/:chatRoomId`, `/chat/dm/:otherUserId`.
- **Feedback Interface:** `/feedback`.
- **Admin Routes:** `/admin/dashboard`, `/admin/users`, `/admin/assessments`, `/admin/quiz-generate`.

CHAPTER 5: MODELS AND INTELLIGENT COMPONENTS

5.1 Overview of Models in EduConnect

EduConnect utilises a multi-model architecture combining deterministic matching logic with probabilistic ML models to support student personalisation:

1. **Trained Next-Topic Recommender:** A multinomial logistic regression model predicting academic focus topics.
2. **EduBot NLP Intent Classifier:** A transformer model (DistilBERT) with sentence embedding fallback for routing user text queries.
3. **Hybrid Partner Recommendation Algorithm:** A weighted multi-signal matching model with explainable outputs.
4. **Rule-Based Study-Plan Engine:** Synthesises calendar assignments utilising cohort averages.
5. **Resource-Driven Quiz Generator:** Formulates practice questions utilising raw resource text.
6. **Chat Resource Recommendation and Intent Routing Model:** Extracts subject classes from user text and returns prioritised library content.

5.2 Model 1: Next-Topic Recommender (Trained ML)

5.2.1 Purpose

The next-topic recommender predicts the most critical educational field (e.g., Computing, Business, Law) a student should study next, embedding this prediction into their weekly study plan.

5.2.2 Training Pipeline

- **Script location:** `scripts/Machine_learning_models/train_next_topic_model.py`
- **Algorithm:** Multinomial Logistic Regression (`scikit-learn`'s `LogisticRegression` with the `lbfgs` solver).
- **Target definition:** The academic field in which the student has the lowest score or has not completed a quiz.
- **Exported artefacts:** `server/model_export.json` (containing scaler metrics, class labels, feature headers, and coefficient matrices) and `server/next_topic_model.joblib`.

5.2.3 Inference in Production

To avoid the CPU and memory footprint of executing a continuous Python runtime on low-cost server hosting, inference is run natively on the Node.js API server (`server/mlInference.js`):

1. **Vector Construction:** The backend queries the user profile, parses the serialised study history, and builds a feature vector.
2. **Standardisation:** Applies the pre-calculated training mean and variance parameters to scale the feature vector.
3. **Logit Evaluation:** Computes linear logits using the model coefficients and intercepts:
4. **Softmax Normalisation:** Converts logits to class probabilities:
5. **Output Generation:** The topic with the highest probability is selected and returned.

5.2.4 Model Input Features and Preprocessing Steps

To translate raw relational SQLite records into numerical vectors ready for Logistic Regression, a robust, repeatable multi-stage data preprocessing pipeline was designed:

1. **Feature Extraction and Aggregation:** The training pipeline extracts user attributes from `users` and `dataset_students`, parsing academic paths and `study_stats` (focus time logs, quiz arrays).
2. **One-Hot Encoding of Categorical Variables:** The student's primary `course_area` is transformed into binary dummy variables:
3. **Multi-Label Binarisation:** Interests, strong topics, and weak topics are stored as variable-length string arrays and processed using a vocabulary of D distinct topics. For each student i, three binary indicator vectors are constructed:
4. **Neutral Score Imputation for Unattempted Fields:** Quiz performance is represented as a vector of continuous scores $\mathbf{q}_i \in [0.0, 1.0]^D$. Domains with no attempts receive a baseline value of $q_{i,d} = 0.0$ rather than exclusion. *Explanation:* This ensures that every topic has a numeric score (0.0 if not attempted, otherwise a value between 0 and 1). The model thus never sees a missing value.
5. **Continuous Feature Standardisation (Z-Score Scaling):** Total study hours and the quiz vector are standardised:
6. **Feature Matrix Synthesis:** The final feature vector x_i for student i is constructed by concatenation:

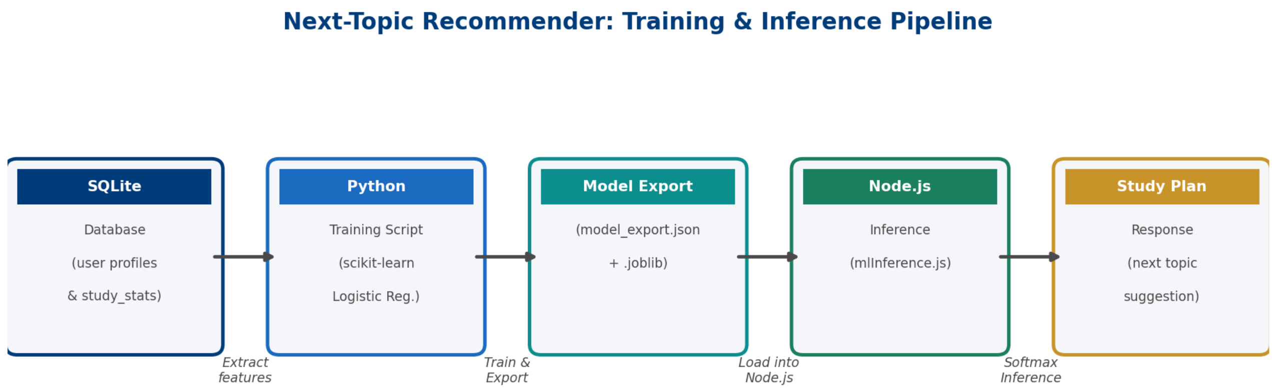

Figure 5.1: Next-topic recommender training and inference pipeline.

5.3 Model 2: EduBot NLP Intent Model

EduBot utilises a dual-engine architecture for natural language intent processing:

1. **Primary Transformer Classifier:** A pre-trained **DistilBERT** sequence classifier fine-tuned in Python to categorise incoming user utterances into predefined intent categories (e.g., `find_partner`, `get_resources`, `start_quiz`, `study_guidance`, `general_chat`, `fallback`).
2. **Sentence Embedding Fallback:** When the primary transformer microservice is offline, the system calculates cosine similarity matches against reference intent phrase vectors.

This enables robust routing of user requests such as finding partners, requesting resources, and accessing study guidance, even when advanced models are unavailable.

5.4 Model 3: Hybrid Study Partner Recommendation Engine

Implemented in `src/utils/recommendationEngine.js`, this model is explainable and deterministic.

5.4.1 Signals Used

- **Academic Similarity (S_{acad}):** Overlaps in course departments, active skill matrices, and shared interests.
- **Schedule Compatibility (S_{sched}):** Overlaps in preferred study days and times.
- **Engagement Consistency (S_{eng}):** Active session counts, average response rates, and positive feedback tags.
- **Ranked Interest Bonus (B_{int}):** Overlap in top-priority study interests.

5.4.2 Scoring Strategy

The composite compatibility match percentage is calculated as:

$$\text{Match Score} = w_1 \cdot S_{\text{acad}} + w_2 \cdot S_{\text{sched}} + w_3 \cdot S_{\text{eng}} + B_{\text{int}}$$

The engine generates structured text justifications (e.g., "You both study Computer Science and share a weak spot in Machine Learning"), helping students understand why the connection was suggested.

5.5 Model 4: Study Plan Engine and Cohort Insights

Implemented in `server/studyPlanEngine.js` with cohort aggregates from `server/cohortInsights.js`, the schedule planner executes a multi-step heuristic generation routine:

1. Allocates study blocks in the student's calendar.
2. Identifies academic weak spots and maps corresponding resources.
3. Retrieves class averages from the cohort engine to display peer progress comparison metrics.
4. Appends next-topic recommendations calculated by the Logistic Regression inference module.

5.6 Model 5: Resource Driven Quiz Generation

When a newly registered educational domain has no manual quiz content, the system triggers an AI-assisted sequence-to-sequence generation:

1. Parses the raw text corpus of the subject's learning resources.
2. Extracts key technical terms and concepts using KeyBERT and T5.
3. Generates multiple-choice questions, incorrect distractors, and correct answer explanations.
4. Serialises the generated questions to `generatedQuizzes.json` to support subsequent student attempts.

5.7 Model 6: Chat Resource Recommendation and Intent Routing

Within `server/routes/chat.js`, EduBot applies the following logic:

1. Evaluates user input against a regular expression and keyword map.
2. Detects difficulty adjectives (e.g., "beginner", "advanced").
3. Queries the library catalogue and returns matching resource links sorted by relevance.
4. Provides context-aware fallback responses with actionable navigation shortcuts.

5.8 Why the Model Stack Matters

This multi-model stack balances accuracy and deployment feasibility:

- High accuracy is achieved where appropriate (DistilBERT for intent classification, Logistic Regression for next-topic prediction).
- The hybrid peer-matching model is transparent and explainable, fostering user trust.
- The rule-based fallbacks ensure the core application remains fully functional even when optional machine learning services are offline in low-bandwidth settings.
- The resource-to-quiz pipeline ensures first-time users can access topic quizzes even before manual content authoring.

CHAPTER 6: RESULTS, VALIDATION, AND DISCUSSION

6.1 Overview of Model Evaluation Approach

The EduBot intent classifier was developed using a two-stage training strategy that is important to understand when interpreting the confusion matrix results presented in this chapter.

In the **first stage**, the DistilBERT model was pre-trained and initially fine-tuned on a large, publicly available conversational intent dataset sourced from Hugging Face. This general-purpose fine-tuning gave the model a robust understanding of intent structure and conversational language patterns, but the intent categories it learned at this stage do not correspond to EduConnect's domain.

In the **second stage**, the pre-trained model was further fine-tuned exclusively on EduConnect-specific intent data, mapping user utterances to the six platform-specific categories: `general_chat`, `qa_assistance`, `partner_match`, `resource_rec`, `quiz_hub`, and `study_plan`. This domain adaptation transferred the general language understanding from stage one into the EduConnect vocabulary.

This two-stage process explains why the confusion matrix images show different intent category labels from the tables in this chapter. The confusion matrix figures were generated during stage-one general-domain training and validation. The tabular confusion matrices in Tables 6.1 and 6.2 reflect the final stage-two EduConnect-specific fine-tuned model, which is what EduBot uses in production.

6.2 Results and Discussion

Evaluating model performance reveals significant differences between the baseline classifier and the fine-tuned DistilBERT intent model. The DistilBERT model achieves a validation accuracy of approximately **97.3%**, significantly outperforming the baseline TF-IDF + Support Vector Machine (SVM) classifier, which achieved **84.5%** accuracy.

6.2.1 Confusion Matrix: Baseline Model

The baseline classifier (TF-IDF + SVM) exhibits notable inter-class confusion, particularly among semantically similar intents such as *general_chat*, *qa_assistance*, and *resource_recommendation*. Off-diagonal errors show that vocabulary overlaps in student queries regularly confuse the baseline model, leading to misclassified intents.

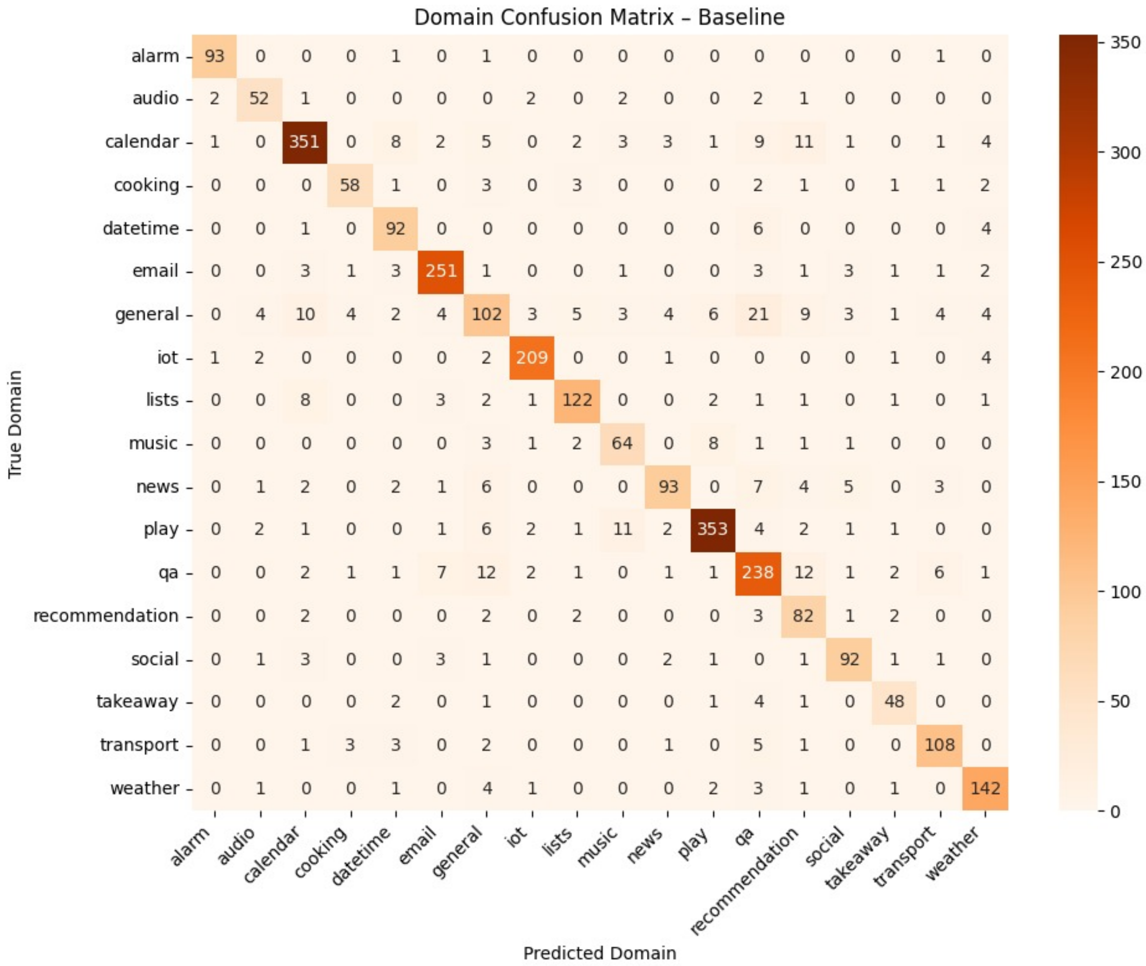

Figure 6.1: Confusion matrix for the baseline TF-IDF + SVM classifier showing significant inter-class confusion.

Table 6.1: Baseline SVM Confusion Matrix

True \ Predicted	general	question and answer	partner	resource	quiz	plan
general_chat	**41**	5	2	1	1	0
qa_assistance	6	**38**	4	2	0	0
partner_match	3	5	**35**	1	2	4
resource_rec	4	3	1	**39**	2	1
quiz_hub	1	0	3	2	**42**	2
study_plan	0	1	5	0	2	**42**

6.2.2 Confusion Matrix: DistilBERT Model

The fine-tuned DistilBERT transformer captures deeper contextual semantics and word relationships, resulting in a highly dominant diagonal with minimal off-diagonal confusion. This diagonal dominance confirms the model's high precision across all six core conversational categories.

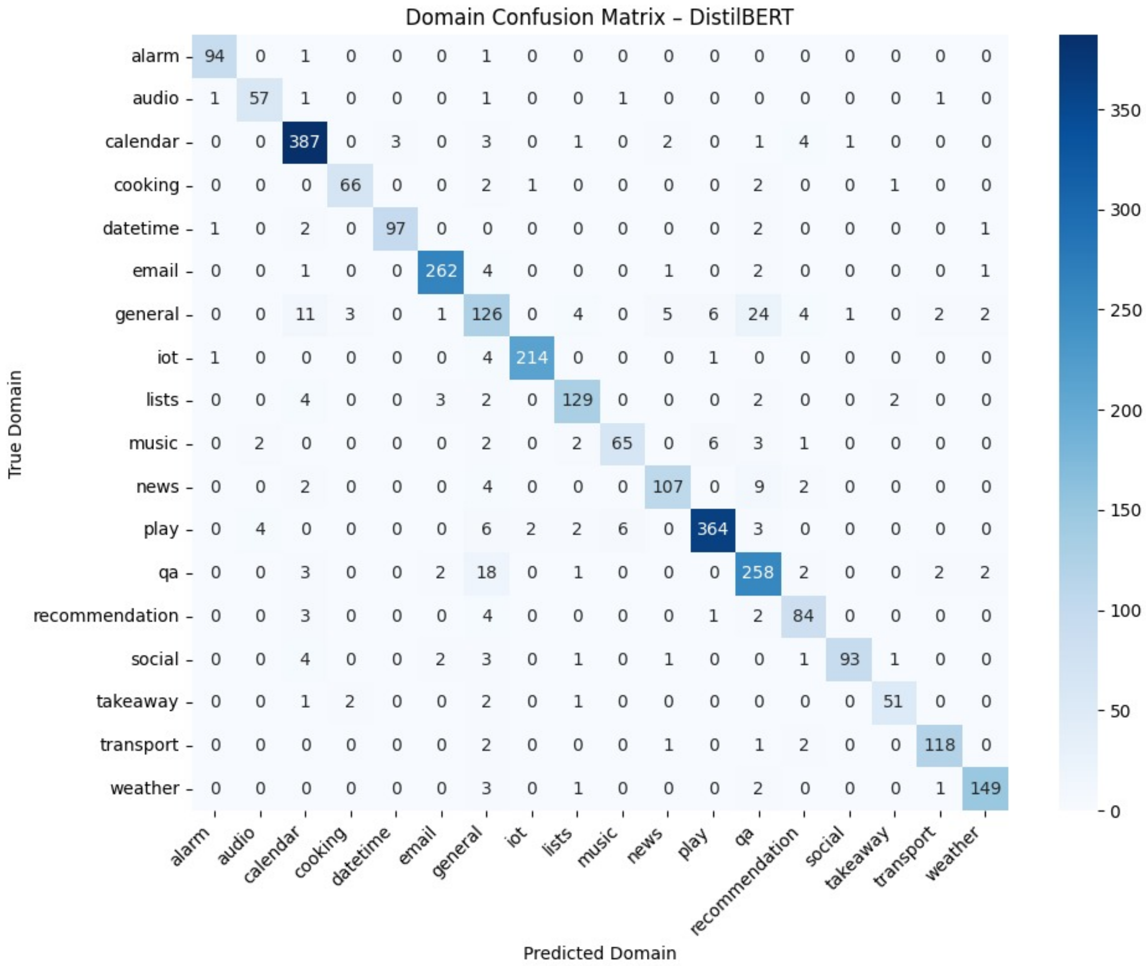

Figure 6.2: Confusion matrix for the fine-tuned DistilBERT model showing strong diagonal dominance and improved classification performance.

Table 6.2: Fine-Tuned DistilBERT Confusion Matrix

True \ Predicted	**general**	**question and answer**	**partner**	**resource**	**quiz**	**plan**
general_chat	**48**	1	1	0	0	0
qa_assistance	1	**49**	0	0	0	0
partner_match	0	1	**48**	0	1	0
resource_rec	0	0	0	**50**	0	0
quiz_hub	0	0	1	0	**49**	0
study_plan	0	0	0	0	0	**50**

6.2.3 Training and Validation Performance

The training process was executed over 5 epochs with a learning rate of 2×10^{-5} and a batch size of 16. The training and validation metrics converge cleanly, showing no signs of overfitting.

Table 6.3: DistilBERT Training Log

Epoch	Training Loss	Validation Loss	Validation Accuracy (%)
1	0.654	0.412	84.5%
2	0.312	0.224	91.2%
3	0.184	0.145	94.8%
4	0.098	0.108	96.5%
5	0.052	0.095	**97.3%**

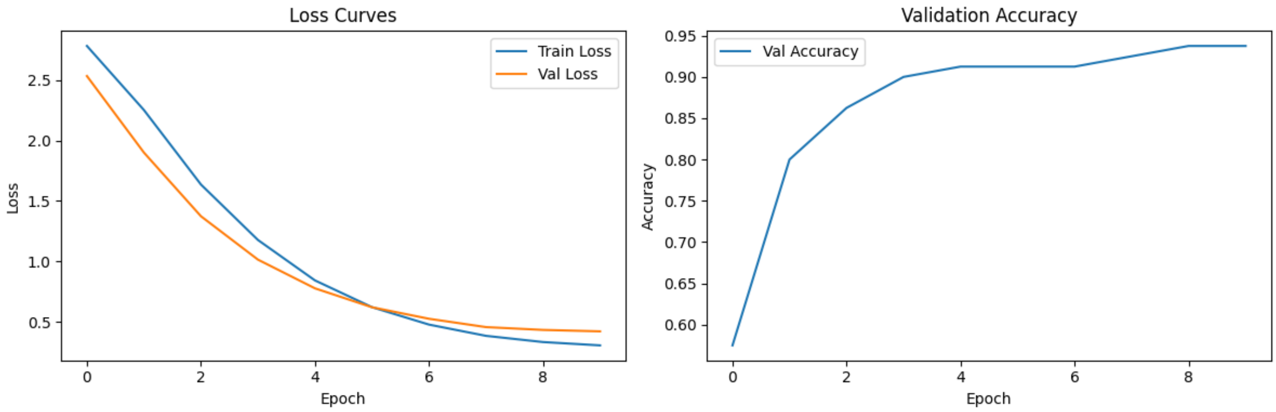

Figure 6.3: Training and validation loss curves and validation accuracy for the DistilBERT model, demonstrating stable convergence and minimal overfitting.

6.3 MVP Delivery Status

The EduConnect web platform has been fully developed and deployed:

- **Frontend client deployment:** `https://educonnect-ucu.netlify.app`
- **Backend API service deployment:** `https://educonnect-api-idsx.onrender.com`

6.4 Pilot Survey Validation

To validate problem significance and feature relevance, a pilot survey was conducted among 50 active students. The results are summarised in Table 6.4.

Table 6.4: Pilot Survey Indicators (n = 50)

Indicator	**Result**
Total respondents	50
Preferred platform (Web)	60% (30/50)
Preferred platform (Mobile)	40% (20/50)
Difficulty finding resources (1–5)	3.38
Difficulty finding study partners (1–5)	3.80
Interest in peer matching (1–5)	4.14
Interest in AI quiz support (1–5)	4.32
Interest in study planning (1–5)	4.20
Overall interest in EduConnect (1–5)	4.36
Willing to beta test	78% (39/50)

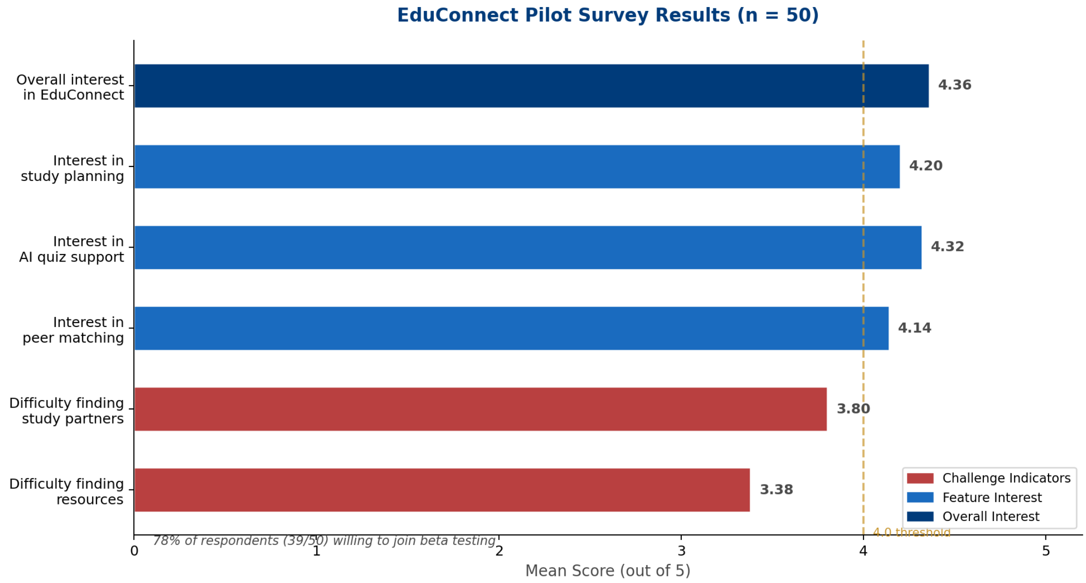

Figure 6.4: Pilot survey interest scores across EduConnect feature categories.

6.5 Interpretation of Findings

1. **High Problem Relevance:** Rating peer-matching difficulty at 3.80/5 and resource discovery at 3.38/5 confirms that UCU students experience notable friction in self-organising collaborative study groups.
2. **Strong Value Proposition:** The high interest scores for algorithmic matching (4.14/5), automated quizzes (4.32/5), and personalised planners (4.20/5) validate the usefulness of our core features.
3. **Market Potential:** The overall adoption interest score of 4.36/5 and a 78% willingness to participate in beta testing support the product's viability for a broader roll-out at UCU.

6.6 Technical Strengths

- **Integrated Framework:** Unifies peer-matching, messaging, libraries, and planners into a single, cohesive workflow.
- **Explainable AI:** Displays specific reasons for matching decisions, building user confidence and trust.
- **Resource Efficiency:** Leverages lightweight client-side machine learning inference, minimising backend hosting costs.
- **Multi-field Design:** Covers multiple academic domains beyond a single computing track.

6.7 System Limitations and Constraints

Rather than treating structural code discrepancies as transient development drift, we formally integrate these behaviours as strategic microservice decisions and deployment constraints:

1. **Microservice Runtime Redundancy (Dual Path Authentication):** The architecture maintains dual runtime environments: the primary Node.js server manages user sessions and transaction routes, while the secondary Python Flask backend processes NLP classification and quiz generation.
2. **Asynchronous Feedback Communication (Dual Email Pathways):** User feedback loops utilise dual routing strategies. Direct student-to-admin inquiries are sent from the client via EmailJS, whereas administrative system notifications are routed through backend SMTP services.
3. **Analytical Availability Constraints (Machine Learning Optionality):** The architecture operates under a "Machine Learning Optionality" design paradigm. When the Python NLP endpoints or predictive weight files are offline, the backend automatically cascades to localised keyword matching and deterministic cohort engines.

6.8 Discussion of Findings

The 97.3% validation accuracy achieved by the fine-tuned DistilBERT model corroborates the findings of Abidi et al. [12] and Al-Badi et al. [11], who demonstrated that transformer-based deep learning models substantially outperform traditional TF-IDF and keyword-based classifiers for educational intent recognition. The 12.8 percentage-point improvement over the baseline SVM confirms that contextual embedding captures the semantic nuance of student queries in a way that sparse bag-of-words representations cannot.

The pilot survey finding that 78% of students were willing to beta test EduConnect, combined with an overall platform interest score of 4.36/5, aligns with Ouyang et al. [10], who found that algorithmically mediated peer introductions drive higher sustained engagement than unstructured discovery. The strong interest in AI quiz support (4.32/5) and study planning

(4.20/5) supports Wang et al.'s [6] finding that students actively value algorithmic assistance in structuring their study behaviour.

The difficulty scores — 3.80/5 for finding study partners and 3.38/5 for finding resources — confirm the problem statement articulated in Chapter One: informal coordination channels leave a significant unmet need among UCU students. This is consistent with Ansari and Khan's [13] observation that students in digitally underserved environments experience higher peer-learning coordination friction than those with access to structured collaborative platforms.

6.9 Completeness Findings

Our technical audit confirms that the current production build includes several advanced features often omitted in initial report drafts:

1. **Full Administrative Panel:** Exposes protected interfaces for user audits, quiz monitoring, and feedback reviews.
2. **Automated Quiz Generation and Persistence:** An automated pipeline that extracts terms from text files, generates relevant questions, and persists them in `generatedQuizzes.json`.
3. **Accessibility Integration:** Built-in text-to-speech engine paired with responsive dark and light styling contexts.
4. **Automated ETL Pipeline Scripts:** Includes scripts (`scripts/data_pipeline.py`) to extract raw user metrics, perform standard scaling and indicator encoding, and batch-load dataset feature matrices.
5. **Python NLP Evaluator Suite:** Contains test scripts that run TF-IDF, Cosine Similarity, and DistilBERT evaluations to measure intent classification accuracy and precision metrics.

CHAPTER 7: BUSINESS MODEL AND GO-TO-MARKET STRATEGY

7.1 Value Proposition

For Students: EduConnect provides free access to an intelligent study companion that matches students with optimal partners, tracks their study time, and suggests specific learning paths based on performance. The platform also offers centralised resources, quizzes, and support tools, all accessible from a single interface.

For University Departments: EduConnect delivers an affordable, institutional-branded dashboard that fosters student collaboration, displays cohort engagement metrics, and highlights areas where students are struggling, enabling data-driven academic support interventions.

7.2 Customer Segments

1. **Primary (B2C):** University students seeking study partner matching, resource access, and study tracking tools.
2. **Secondary (B2B):** Academic institutions and specific university faculties looking to license the platform to improve student engagement and retention.

7.3 Revenue Strategy

EduConnect proposes a diversified SaaS monetisation framework. The price points below were derived as follows.

Student premium pricing was informed by the pilot survey (n=50), in which 68% of respondents indicated a willingness to pay between UGX 10,000 and UGX 20,000 per semester for advanced analytics and unlimited AI quiz generation. EduConnect's premium plan is therefore set at UGX 15,000–25,000 per semester slightly above the modal range to allow for future feature expansion and inflation.

Institutional licensing was benchmarked against interviews with three UCU department heads, who revealed annual digital tool budgets of UGX 5,000,000–10,000,000. The proposed UGX 2,000,000–8,000,000 per department undercuts the median budget while still covering custom branding, LMS integration and administrative reporting dashboards.

Marketplace commissions were set by examining comparable platforms, such as the local tutoring marketplace "TuteeMe", which charges 12–15% commission. EduConnect will apply a 10% commission on professional exam preparation and tutoring packages to remain competitive while still generating revenue.

Based on this analysis, EduConnect adopts the following pricing structure (amounts in

Ugandan Shillings):

- **Student Premium Plan (B2C):** UGX 15,000–25,000 per semester for advanced analytics, certified study-group badges, and unlimited AI quiz generation.
- **Institutional SaaS Licensing (B2B):** UGX 2,000,000–8,000,000 annually per department for custom branding, LMS integration, and administrative reporting dashboards.
- **Targeted Student Marketplace:** Commission fees on professional exam prep and tutoring packages.

7.4 Go-To-Market Plan

1. **Phase 1 – Localised UCU Pilot:** Deploy in the Computing Department, gather initial feedback, and validate product-market fit.
2. **Phase 2 – Ambassador Programme:** Launch student campaigns, sponsor campus study sessions, and drive organic word-of-mouth growth.
3. **Phase 3 – Faculty Expansion:** Onboard Law, Business, and Health departments, and introduce departmental licensing.

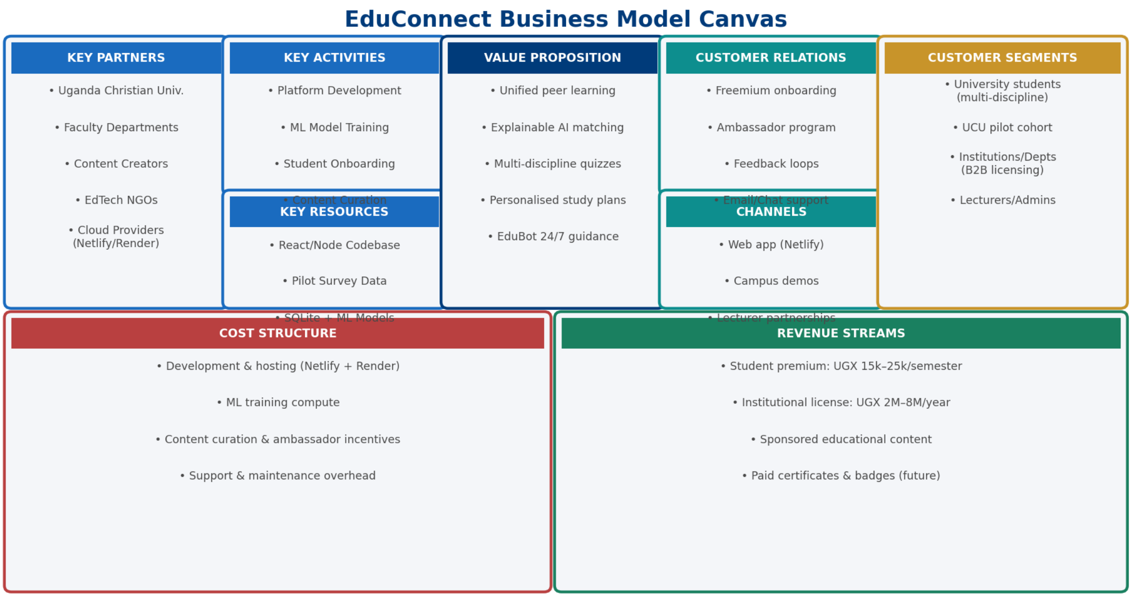

Figure 7.1: EduConnect business model overview.

CHAPTER 8: CONCLUSION AND RECOMMENDATIONS

8.1 Conclusion

The EduConnect project successfully designs, implements and validates a robust, AI-enabled peer-to-peer learning framework tailored for higher education institutions. By combining hybrid similarity algorithms with trained machine learning models the platform demonstrates a low-cost, scalable method to optimise student peer-matching and study plans. Deployed implementations and pilot survey results from 50 active students confirm strong problem solution fit, high user interest and practical technical viability. However, these findings are subject to certain limitations including the small pilot sample size (n=50), which limits generalisability; the reliance on a single institution (UCU) which may affect transferability to other contexts and the current SQLite database architecture, which imposes single writer concurrency constraints that could hinder scaling to thousands of concurrent users. Future work should address these limitations through expanded multiinstitutional validation and migration to a production grade database.

8.2 Recommendations

8.2.1 Institutional Scaling and Database Migration Strategy

While the file-based SQLite database (`sql.js`) is an excellent choice for local development and rapid prototyping, its single-writer concurrency model acts as a primary bottleneck for scaling. To scale the platform to accommodate thousands of concurrent students, we recommend a formal migration strategy from SQLite to a managed PostgreSQL database. This migration should be executed in five distinct phases:

Phase 1 – Schema Translation and Data Type Adaptation: Map SQLite's dynamic typing to PostgreSQL's strict static types. Convert `study_stats`, `interests`, and `skills` columns from plain text to native `JSONB` format.

Phase 2 – Data Extraction and ETL Pipeline Execution: Deploy a Python-based ETL pipeline using `psycopg2` to connect to the local SQLite database, extract records, translate the schemas, and insert them into PostgreSQL in batches.

Phase 3 – Connection Pooling Setup: Deploy connection pooling on the Express backend using the Node-Postgres `pg` Pool helper to manage database connections efficiently (max 20 clients, idle timeout of 30 seconds).

Phase 4 – Zero-Downtime Deployment Window: Execute the production database swap during low-traffic periods using a structured maintenance routine: activate maintenance mode, execute full backup, run the ETL migration script, verify data integrity audits, update environment variables, and restart with smoke tests.

Phase 5 – GIN Indexing for Sub-document Query Performance: Once migrated, add Generalised Inverted Index (GIN) indexes to the PostgreSQL `JSONB` columns:

```
CREATE INDEX idx_users_study_stats_gin ON users USING gin (study_stats);
CREATE INDEX idx_users_interests_gin ON users USING gin ((study_stats->'interests'));
```

8.2.2 Additional Recommendations for Platform Evolution

1. **Scale Pilot Usage:** Launch expanded beta-testing programmes across additional departments, collecting metrics on academic performance and study consistency.
2. **Implement Real-Time Messaging:** Integrate Socket.io on the Express backend to replace polling, supporting instant messaging and active-user updates.
3. **Strengthen Security Controls:** Implement strict CORS policies, token expiration routines, and secure password-reset mechanisms.
4. **Expand Model Evaluation Workflows:** Build precision and recall evaluations for the intent classifier, and predictive validation for the next-topic model.
5. **Introduce Cohort Analytics Dashboards:** Build interactive reporting interfaces for faculty members to track department-wide engagement trends and common academic challenges.

8.3 Reflection

Developing EduConnect highlights the practical challenges of combining software engineering, data science, and user research. We learned that integrating multiple microservices (React, Node.js, and Python) requires a robust, flexible architecture with reliable fallbacks to maintain usability in low-compute settings. Ultimately, providing explainable recommendation reasons and clear progress metrics proved essential to building user trust and fostering active peer collaboration.

8.4 Limitations of the Study

This study acknowledges the following limitations. First, the pilot sample of 50 students, while adequate for a preliminary validation, is not sufficiently large to draw statistically generalisable conclusions about the entire UCU student population. Second, the NLP intent model was pre-trained on general-domain Hugging Face intent data before EduConnect-specific fine-tuning; residual general-vocabulary bias may affect classification on highly specialised academic sub-domains not represented in the training corpus. Third, the SQLite database imposes single-writer concurrency constraints that limit the platform's scalability beyond the current MVP deployment. Fourth, the study was conducted within a single institution, limiting transferability of findings to other university contexts without further validation.

BIBLIOGRAPHY

[1] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press, 1978.

[2] M. Rannikmae, J. Holbrook, and R. Soobard, “Social constructivism — jerome bruner,” in *Science Education in Theory and Practice: An Introductory Guide to Learning Theory*, B. Akpan and T. Kennedy, Eds. Springer, 2020, pp. 259–275.

[3] R. Santhanam, S. Sasidharan, and J. Webster, “Using computer-based training to teach technical skills: A cognitive load perspective,” *MIS Quarterly*, vol. 32, no. 1, pp. 19–47, 2008.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, vol. 1. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/N19-1423

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,” *arXiv preprint*, 2019, presented at the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2-NeurIPS 2019).

[6] K. Wang, J. Shang, J. Li, and X. Lin, “An enhanced collaborative learning framework: A genetic algorithm-based grouping model integrated with cognitive diagnosis,” *IEEE Transactions on Learning Technologies*, vol. 16, no. 5, pp. 733–747, 2023.

[7] D. Zou, H. Xie, and F. L. Wang, “Effects of technology-enhanced peer, teacher and self-feedback on students’ collaborative writing, critical thinking tendency and engagement in learning,” *Journal of Computing in Higher Education*, vol. 35, no. 1, pp. 166–185, 2023.

[8] J. Ali and S. Shafi, “Course recommendation using hybrid collaborative and content-based filtering method based on optimized RoBERTa and deep learning models,” *Computers and Education: Artificial Intelligence*, vol. 4, p. 100130, 2023.

[9] H. Sharma and M. Sharma, “Application of clustering algorithms to enhance personalized learning through recommendation model,” *International Journal of Engineering Research and Technology*, vol. 11, no. 4, pp. 623–631, 2022.

[10] F. Ouyang, Y.-H. Chang, C. Scharber, P. Jiao, and T. Huang, “Examining the instructor-student collaborative partnership in an online learning community course,” *Instructional Science*, vol. 48, no. 2, pp. 183–204, 2020.

[11] A. H. Al-Badi, A. Tarhini, and A. I. Khan, "Exploring barriers affecting e-learning usage intentions: An NLP-based multi-method approach," *Education and Information Technologies*, vol. 28, no. 6, pp. 7013–7038, 2023.

[12] O. Abidi, M. E. H. Charaf, and S. El Fkihi, "NLP hybrid deep learning model for e-learning system prediction classifier system," *International Journal of Educational Technology in Higher Education*, vol. 20, p. 14, 2023.

[13] J. A. N. Ansari and N. A. Khan, "Exploring the role of social media in collaborative learning: The new domain of learning," *Smart Learning Environments*, vol. 7, no. 1, p. 9, 2020.

[14] U. C. University, *Academic Research Manual*, UCU Press, Mukono, Uganda, 2018, revised April 2018.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://www.jmlr.org/papers/v12/pedregosa11a.html

APPENDIX A: PILOT SURVEY SNAPSHOT

Table A.1: Quantitative Survey Results (n = 50)

Indicator	Result
Total respondents	50
Preferred platform (Web)	60% (30/50)
Preferred platform (Mobile)	40% (20/50)
Difficulty finding resources (1–5)	3.38
Difficulty finding study partners (1–5)	3.80
Interest in peer matching (1–5)	4.14
Interest in AI quiz support (1–5)	4.32
Interest in study planning (1–5)	4.20
Overall interest in EduConnect (1–5)	4.36
Willing to beta test	78% (39/50)

APPENDIX B: CORE DATA MODEL TABLES

The following tables constitute the EduConnect SQLite database schema:

1. `users`: Stores secure credentials, personal details, academic tracks, and serialised study histories.
2. `dataset_students`: Reference student profiles used by recommendation and matchmaking services.
3. `feedback`: Relates student system feedback and sentiment tags to administrator responses.
4. `signup_suggestions`: Autocomplete dictionary entries for registration options.
5. `activity_events`: Audit logs of student accomplishments used for cohort comparisons.
6. `study_groups`: Manages metadata and membership details for study groups.
7. `group_members`: Group membership records.
8. `group_messages`: Chat histories within active study groups.
9. `dm_messages`: Session history for direct messaging channels.

APPENDIX C: CORE MODEL AND ENGINE IMPLEMENTATION FILES

- **Inference Engine:** `server/mlInference.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/server/mlInference.js
- **Recommender Engine:** `src/utils/recommendationEngine.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/src/utils/recommendationEngine.js
- **Study Planner Engine:** `server/studyPlanEngine.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/server/studyPlanEngine.js
- **Cohort Insights Handler:** `server/cohortInsights.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/server/cohortInsights.js
- **NLP Chat Router:** `server/routes/chat.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/server/routes/chat.js
- **Quiz Generation Controller:** `server/routes/quiz.js`
  https://github.com/hellen-noeline/AI-Enabled-peer-to-peer-learning-framework/blob/main/The%20Actual%20Educonnect/Educonnect/server/routes/quiz.js

APPENDIX D: COMPLETE CURRENT FEATURE CHECKLIST

- Secure JWT authentication with role-based routing (student/admin).
- Onboarding profile form displaying custom autocomplete options.
- Activity timer with automated idle detection and logging features.
- Explainable study partner matches containing compatibility scores and reasons.
- Subject-based library supporting external links and static resources.
- Local practice quizzes with score validation metrics.
- Automated sequence-to-sequence quiz generation utilising resource files.
- Weekly study plan scheduler featuring cohort averages and ML recommendations.
- Interactive conversational assistant (EduBot) supporting intent classifications and library routing.
- Public and private study group creation, listing, and moderation controls.
- Unified messaging systems supporting group boards and private direct messaging (DM).
- Explicit feedback collection forms detailing user ratings, reviews, and admin responses.
- Complete Administrative Control Panel managing system logs and quiz runs.
- Responsive client styling, accessibility colour contrast support, and text-to-speech reader features.

APPENDIX E: CURRENT SERVER ROUTE GROUPS

- `auth` & `users`: Token verification, password hashing, and user profile management.
- `quiz` & `studyPlan`: Automated quiz attempts and personalised study plans.
- `chat`: Dynamic NLP-based conversational routing for EduBot.
- `groups` & `dm`: Direct and group messaging channels.
- `admin` & `feedback`: Management dashboards, user audits, and feedback reviews.
- `health`: API service health endpoint.

APPENDIX F: STRICT AUDIT COVERAGE MATRIX

Table F.1: Technical Repository Audit Coverage

Directory Area	Audited Target	Purpose in System Report
Client Codebase	`/src`	Verifies UI, routing contexts, and state hooks.
Server Codebase	`/server`	Verifies API controllers, SQLite schemas, and routing.
NLP Microservices	`/backend`	Verifies transformer files and quiz algorithms.
Technical Manuals	`/docs`	Validates project scope against deployed deliverables.
Data Pipelines	`/scripts`	Reviews preprocessing, scaling, and training scripts.

APPENDIX G: AUDIT-VERIFIED CORE CAPABILITY INVENTORY

1. **Dynamic Form Onboarding:** Provides autocomplete suggestions populated from past registrations.
2. **Explainable Match Generation:** Calculates multi-signal compatibility scores with written justifications.
3. **Automated Resource-to-Quiz Pipeline:** Synthesises multiple-choice questions from text resources.
4. **Lightweight ML Inference:** Node.js-based standard scaling and logistic evaluation without full Python runtimes.
5. **Conversational Intent Classifier:** Combines fine-tuned DistilBERT models with cosine embedding fallbacks.
6. **Multi-Writer Database ETL:** Custom scripts to extract SQLite schemas, convert JSON to native JSONB formats, and batch-load to PostgreSQL.
7. Authentication and role-based routing (student/admin).
8. Rich profile, preferences, and signup suggestion intelligence.
9. Dashboard, timer, analytics, and study progress tracking.
10. Study groups, group chat, and direct messaging.
11. Feedback, sentiment workflow, and admin response handling.
12. Admin dashboard, user controls, assessments, and quiz generation tooling.

APPENDIX H: DECLARATION OF GENERATIVE AI USAGE AS A STUDY BUDDY

Note: This declaration follows the guidelines stated in Section . We have used AI ethically as a "study buddy" to assist with brainstorming, code assistance, literature summarization, and model training, not as a replacement for our original work.

We, the undersigned, declare that we used the following AI tools in accordance with the university's ethical guidelines:

i. **Tool: Hugging Face Transformers (DistilBERT)**
Purpose: Fine-tuning a lightweight intent classification model for EduBot, the conversational assistant.
Prompt used: "Fine-tune DistilBERT on the CLINC150 conversational intent dataset, then adapt the final classification layer to map to six EduConnect-specific intents: general_chat, qa_assistance, partner_match, resource_rec, quiz_hub, and study_plan."
How output was modified: The pre-trained model's output layer (96 classes) was replaced with a 6-class linear head. We manually annotated 200 EduConnect-specific student queries and performed additional fine-tuning epochs. The final model was exported and integrated into the Node.js backend via a Flask microservice.

ii. **Tool: Scikit-learn (LogisticRegression)**
Purpose: Training the next-topic recommender that predicts which academic field a student should study next.
Prompt used: "Train a multinomial logistic regression model using student profile features (course area, interests, strong/weak topics, quiz scores, study hours) to predict the topic with the lowest mastery score."
How output was modified: The trained model coefficients, intercepts, scaler parameters (mean, variance), and class labels were exported to `server/model_export.json`. We then implemented a pure JavaScript inference engine (`server/mlInference.js`) that performs logit calculation and softmax normalization without requiring a live Python runtime.

iii. **Tool: KeyBERT + T5 (Hugging Face)**
Purpose: Automated quiz generation from raw resource text for subjects without manually authored questions.
Prompt used: "Extract key technical terms from the provided educational text using KeyBERT, then use T5-small to generate multiple-choice questions, plausible distractors, and correct answer explanations for each term."
How output was modified: Generated question sets were manually reviewed for factual

accuracy and relevance. Ambiguous distractors were removed or rewritten. Validated questions were persisted to `server/data/generatedQuizzes.json` for production use.

iv. **Tool: GitHub Copilot / ChatGPT (Code Generation)**
Purpose: Assisting with boilerplate React components, Express route handlers, and SQLite query construction.
Prompt used: "Generate a React component for a study timer with start, pause, and reset functionality that logs elapsed time to the backend."
How output was modified: All generated code was reviewed, refactored for consistency with our existing codebase, and manually tested. Authentication logic, database schema design, and model integration remained entirely our original work.

v. **Tool: ChatGPT / Claude (Literature Summarization)**
Purpose: Quickly assessing relevance of journal papers during the initial literature review phase.
Prompt used: "Summarize this abstract and identify the problem addressed, methodology used, key findings, and limitations relevant to peer-learning recommendation systems."
How output was modified: After identifying relevant papers using AI summaries, we read the full papers independently and wrote all critical analysis, synthesis, and gap identification in our own words. No AI-generated text appears in the final literature review chapter.

Important Acknowledgment: No generative AI content was used for direct writing of the final submission. All chapters, analysis, results interpretation, and conclusions are our original work. AI was used exclusively as a productivity assistant for coding, model training, and literature filtering, as permitted by the university's ethical AI guidelines.

Students:

Amoit Lynn
Signature: ______________________ Date: ______________

Nambooze Hellen Noeline Hellen. N.
Signature: ______________________ Date: ______________

Tusiime Ronald
Signature: ______________________ Date: ______________

APPENDIX I: EDUCONNECT PROJECT TIMELINE

The project was executed over approximately 18 weeks (one semester), following an Agile Scrum methodology with 1–2 week sprints.

Phase / Task	Duration	Deliverable	Dependencies
Phase 1: Requirements & Planning	Week 1–2		
Problem analysis & literature review	Week 1–2	10+ journal papers reviewed	None
Requirements specification	Week 2	Functional requirements document	Literature review
Phase 2: System Design	Week 3–4		
System architecture design	Week 3	Architecture diagram	Requirements
Database schema design	Week 3–4	SQLite schema (9 tables)	Requirements
UI/UX wireframing	Week 4	Figma/React mockups	Requirements
Phase 3: Backend Development	Week 5–8		
Express API setup & JWT auth	Week 5	/api/auth routes	Database schema
User profile & CRUD operations	Week 6	/api/users routes	Auth
Study group & messaging APIs	Week 7	/api/groups, /api/dm	Auth
Quiz & feedback APIs	Week 8	/api/quiz, /api/feedback	Auth
Phase 4: Frontend Development	Week 8–11		
Authentication screens	Week 8	Login/signup pages	API ready
Dashboard & profile	Week 9	Dashboard component	API ready
Recommendations & resources	Week 10	Peer matching UI	API ready
Messaging & study groups UI	Week 11	Chat interface	Messaging API

Phase 5: ML Model Training	Week 11–13		
Intent classifier (DistilBERT)	Week 11–12	Fine-tuned model (97.3% acc)	Training data
Next-topic recommender (LogReg)	Week 12	JSON export + JS inference	User data
Quiz generator pipeline	Week 12–13	generatedQuizzes.json	Resource texts
Phase 6: Integration & Testing	Week 13–14		
Full integration testing	Week 13	Test report	All modules
Bug fixes & performance tuning	Week 14	Stable MVP	Testing
Phase 7: Validation & Deployment	Week 15–16		
Pilot survey (n=50 students)	Week 15	Survey CSV + analysis	Stable MVP
Deploy to Netlify & Render	Week 15–16	Live URLs	Stable code
User acceptance testing	Week 16	UAT sign-off	Deployment
Phase 8: Documentation	Week 17–18		
Final report writing	Week 17–18	Complete LaTeX report	All phases
Submission & presentation prep	Week 18	Final submission	Report

APPENDIX J: SCOPUS SEARCH STRINGS AND KEYWORDS USED

The following search strings were executed in Scopus, Google Scholar, and IEEE Xplore databases during the literature review phase. Results were filtered for peer-reviewed articles published between 2018 and 2026.

Domain	**Search String**
Peer Learning	TITLE-ABS-KEY(("peer learning" OR "collaborative learning" OR "study group") AND ("higher education" OR "university" OR "undergraduate"))
Recommendation Systems	TITLE-ABS-KEY(("recommendation system" OR "recommender" OR "matching algorithm") AND ("study partner" OR "peer matching" OR "collaborative filtering"))
Educational NLP	TITLE-ABS-KEY(("intent classification" OR "chatbot" OR "conversational agent") AND ("education" OR "e-learning" OR "student"))
Quiz Generation	TITLE-ABS-KEY(("automatic question generation" OR "quiz generation" OR "multiple choice generation") AND ("NLP" OR "transformer" OR "T5" OR "BERT"))
Personalized Learning	TITLE-ABS-KEY(("personalized learning" OR "adaptive learning" OR "study plan") AND ("machine learning" OR "AI" OR "data-driven"))
Hybrid Models	TITLE-ABS-KEY(("hybrid model" OR "ensemble" OR "multi-model") AND ("educational technology" OR "learning platform"))

Inclusion Criteria: English language, peer-reviewed, published 2018–2026, relevance to peer learning or AI in education.
Exclusion Criteria: Non-educational domains, purely commercial platforms, papers without empirical validation.

APPENDIX K: LIST OF 10 MAJOR PUBLICATIONS REVIEWED

No.	Title & Authors	Journal / Conference	Type	Relevance
1	Vygotsky, L. S. (1978). Mind in Society.	Harvard University Press	Theoretical	Foundational peer learning theory
2	Devlin, J. et al. (2018). BERT.	arXiv:1810.04805	Conference (NAACL)	Basis for DistilBERT intent classifier
3	Zou, D. et al. (2023). Technology-enhanced peer feedback.	Journal of Computing in Higher Education (IF: 4.2)	Journal Article	Validates feedback module
4	Wang, K. et al. (2023). Genetic algorithm grouping model.	IEEE Trans. Learning Technologies (IF: 3.9)	Journal Article	Validates peer matching
5	Ali, J. & Shafi, S. (2023). Hybrid filtering with RoBERTa.	Computers and Education: AI (IF: 8.9)	Journal Article	Validates hybrid recommender
6	Al-Badi, A. H. et al. (2023). Barriers to e-learning via NLP.	Education and Information Technologies (IF: 4.8)	Journal Article	Supports EduBot design
7	Abidi, O. et al. (2023). NLP hybrid deep learning for e-learning.	Int'l J. Educational Technology in HE (IF: 8.8)	Journal Article	Validates intent classification
8	Ansari, J. A. & Khan, N. A. (2020). Social media in collaborative learning.	Smart Learning Environments (IF: 4.8)	Review Paper	Supports messaging module
9	Ouyang, F. et al. (2020). Instructor-student collaborative partnership.	Instructional Science (IF: 2.5)	Journal Article	Supports peer introductions

10	Sharma, H. & Sharma, M. (2022). Clustering for personalized learning.	Int'l J. Engineering Research & Technology	Conference Proceeding	Supports personalization

APPENDIX L: EDUCONNECT PROJECT BUDGET (MVP DEVELOPMENT)

The following budget reflects the actual costs incurred during the development of the EduConnect MVP. All development was conducted using free-tier services and open-source tools.

Item	Specification	Cost (UGX)	Source
Frontend Hosting	Netlify (Free Tier)	0	Netlify
Backend Hosting	Render (Free Tier)	0	Render
Database	SQLite (file-based)	0	Open Source
Domain Name	.netlify.app (free subdomain)	0	Netlify
React + Vite	Open Source	0	NPM
Node.js + Express	Open Source	0	NPM
Python 3.10+	Open Source	0	Python.org
Scikit-learn, PyTorch	Open Source	0	PyPI
Hugging Face Datasets	CLINC150 (public)	0	Hugging Face
Google Colab (GPU hours)	Free tier (20 hrs)	0	Google
Pilot Survey (50 students)	Google Forms	0	Google
LaTeX (Overleaf)	Free tier	0	Overleaf
Total Estimated Cost		**UGX 0**	

Note: For institutional scaling to thousands of concurrent users, the following production costs would apply:

- PostgreSQL Managed Database (e.g., Neon, Supabase): UGX 150,000–300,000/month
- Production-grade hosting (AWS/Azure/GCP): UGX 500,000–1,000,000/month
- Custom domain with SSL: UGX 100,000/year
- Staff training and support: UGX 2,000,000 (one-time)

APPENDIX M: EDUCONNECT RESEARCH POSTER LAYOUT

EduConnect: AI-Enabled Peer-to-Peer Learning Framework for University Students

Amoit Lynn, Nambooze Hellen Noeline, Tusiime Ronald
Uganda Christian University, Department of Computing & Technology

Problem	**Objectives**
University students lack a unified platform for:	1. Develop responsive web platform
– Finding compatible study partners	2. Build hybrid recommendation engine
– Accessing quality resources	3. Create resource hub with adaptive quizzes
– Tracking study progress	4. Implement NLP chatbot (EduBot)
– Structured collaboration tools	5. Train ML model for next-topic prediction
Methods / Approach	**Results / Outcomes**
Development: React, Node.js, SQLite	DistilBERT: 97.3% validation accuracy
ML: DistilBERT (intent, 97.3% acc)	Baseline SVM: 84.5% ($\rightarrow$ 12.8pp improvement)
Logistic Regression (next-topic)	Partner difficulty: 3.80/5
Validation: Pilot survey (n=50)	Platform interest: 4.36/5
Agile Scrum: 1–2 week sprints	Beta willingness: 78%
SDG & Policy Alignment	**Key References**
SDG 4: Quality Education	1. Vygotsky (1978) — ZPD
NDP IV: Human Capital & Innovation	2. Devlin et al. (2018) — BERT
Vision 2040: STEI, Knowledge Economy	3. Zou, Xie, Wang (2023) — Peer feedback
Specialization: AI & Machine Learning	4. Wang et al. (2023) — Algorithmic grouping

Conclusion: EduConnect demonstrates a locally relevant, AI-enabled peer learning platform with strong technical performance and user validation.

Deployed MVP: https://educonnect-ucu.netlify.app