Capstone Project Concept Note and Implementation Plan

Project Title: Personalized Learning Management System

Team Members

1. Mussie Leake Beraki
2. Liya Tegared Bitew
3. Yordanos Teku Gutema
4. Mekdes Birhanu Abera
5. Dagimawit

Concept Note

1. Project Overview

Project Overview

This capstone project focuses on developing a **Personalized Learning Management System (PLMS)** designed to support Ethiopian Grade 12 students in preparing for the **Ethiopian University Entrance Examination (EUEE)**. The system uses machine learning techniques to analyze student learning styles, study habits, and performance data, then delivers personalized study resources and recommendations accordingly. The goal is to shift away from a "one-size-fits-all" approach toward a tailored educational experience that boosts engagement and academic success.

The project aligns with **SDG 4: Quality Education**, by promoting inclusive and equitable education and ensuring access to quality learning resources for all students, particularly those in underserved and rural areas. By improving student performance and engagement through data-driven personalization, the project also contributes to **reducing educational inequalities** and increasing the national EUEE pass rate, which remains critically low. Ultimately, this system has the potential to empower students to reach their full academic potential and bridge gaps in Ethiopia's secondary education system.

2. Objectives

The primary objective of this project is to design and implement a **Personalized Learning Management System (PLMS)** that leverages machine learning to improve academic performance among Ethiopian Grade 12 students preparing for the Ethiopian University Entrance Examination (EUEE). The specific objectives include:

1. **Analyze Student Learning Patterns**
    - Collect and examine data on students' learning styles, study habits, quiz scores, and past performance to understand key factors influencing academic outcomes.
2. **Develop Predictive Models**
    - Apply machine learning algorithms (e.g., Naïve Bayes, Gradient Boosting, and collaborative filtering) to predict student performance and recommend personalized learning resources.
3. **Provide Personalized Recommendations**
    - Deliver targeted study materials (e.g., Khan Academy, YouTube tutorials, local resources like Fetena) based on each student's unique profile and predicted needs.
4. **Enhance Learning Outcomes**
    - Help students improve their understanding of core subjects and increase their EUEE pass rates through a customized learning experience.
5. **Support Underserved Communities**
    - Ensure accessibility and inclusivity by tailoring the system for students from rural and under-resourced schools, helping bridge the educational gap.

By meeting these objectives, the project will provide an evidence-based solution to a critical national challenge—low EUEE success rates—and support **SDG 4: Quality Education** by making personalized learning tools more accessible to those who need them most.

3. Background

In Ethiopia, the Ethiopian University Entrance Examination (EUEE) plays a critical role in determining students' access to higher education. However, a significant proportion of Grade 12 students struggle to pass the exam due to limited access to personalized learning resources, insufficient guidance, and a mismatch between teaching methods and individual learning needs. Traditional classroom instruction often adopts a one-size-fits-all approach, which fails to address diverse student capabilities and learning styles, particularly in under-resourced schools.

While several educational platforms and tutoring programs exist, most are not tailored to the specific curriculum or contextual challenges faced by Ethiopian students. Initiatives such as Khan Academy and YouTube channels provide general support, but they do not offer customized content based on student performance or targeted feedback. Furthermore, many existing systems lack mechanisms for data-driven decision-making, which limits their ability to adapt to individual learner progress.

A **machine learning approach** introduces a data-centric solution to this challenge. By analyzing student interaction data, study habits, quiz results, and performance trends, ML models can uncover hidden patterns and make accurate predictions about student outcomes. Algorithms such as Naïve Bayes and Gradient Boosting have already demonstrated effectiveness in educational prediction tasks in Ethiopian contexts (e.g., Belachew & Gobena, 2017; Tadesse et al., 2020). Incorporating recommendation techniques like collaborative filtering (Wu et al., 2015) enables the delivery of personalized learning materials aligned with the Ethiopian high school curriculum.

By integrating these methods, this project aims to create a **Personalized Learning Management System (PLMS)** that empowers students with tailored academic support, thereby improving their chances of success in the EUEE and promoting equitable access to higher education.

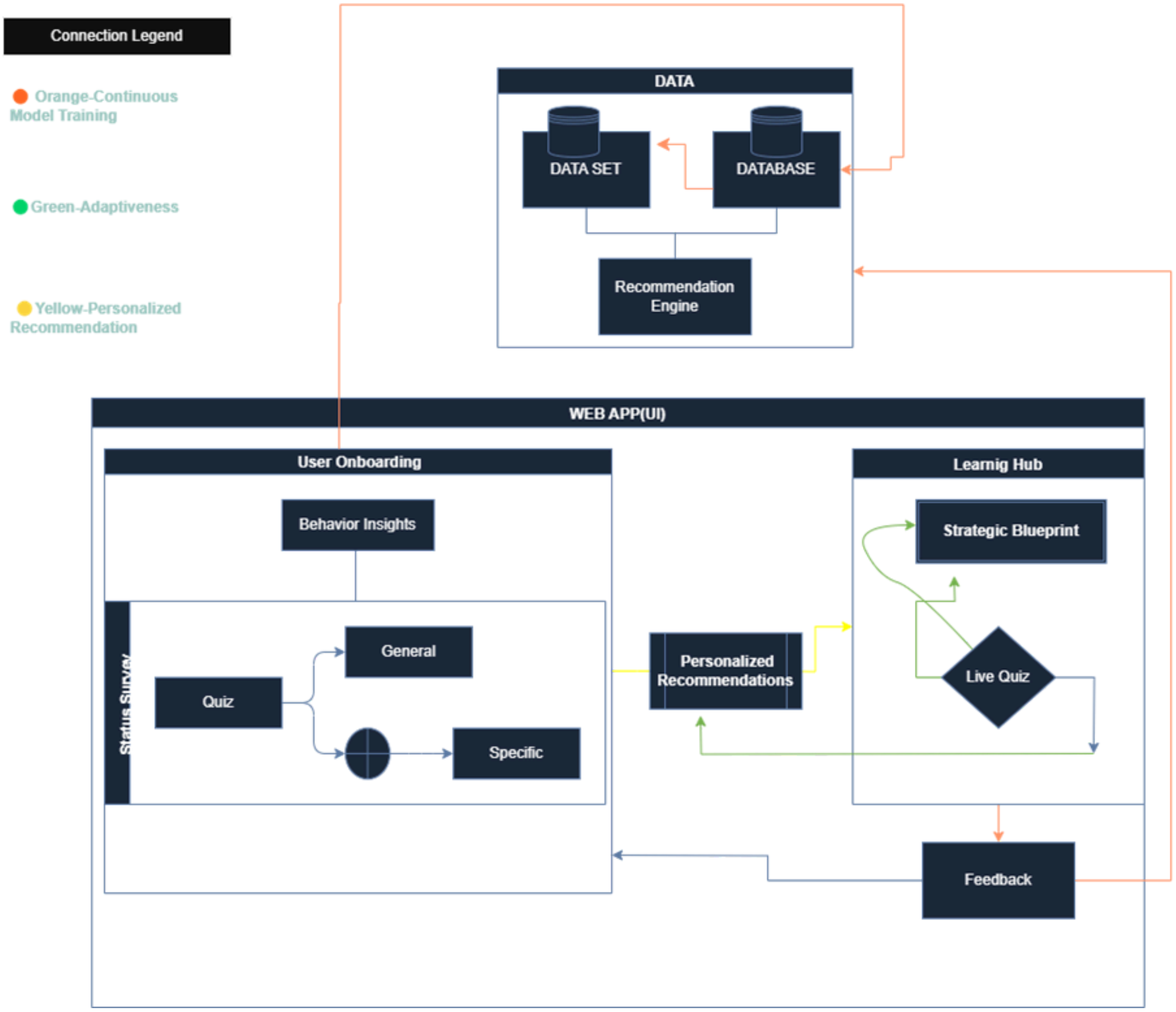
4. Methodology

Our PLMS leverages machine learning techniques to personalize learning experiences and improve student outcomes. Key methodologies include:

- **Decision Trees**: To classify students by learning styles (visual, auditory, kinesthetic) and predict academic performance.
- **Collaborative Filtering (K-Nearest Neighbors)**: To recommend personalized learning materials based on user behavior.
- **Behavioral Analytics**: To track interaction data (e.g., study time, quiz engagement) and provide real-time personalization.

5. Architecture Design Diagram

Personalized Adaptive Learning Management System (PALMS)



COMPONENTS DESCRIPTION

Data Layer

Dataset: Likely includes the historical baseline data (50 Grade 12 students' profiles: learning styles, past experiences, academic outcomes).

Database: Stores new study materials (e.g., Khan Academy videos, YouTube EUEE prep videos) and possibly user data (fresh data from 10-25 app users).

Recommendation Engine: This engine uses data to generate personalized recommendations, likely by implementing ML approaches like collaborative filtering (KNN) or decision trees.

Web App/UI Layer

User Onboarding: Collects initial student data (e.g., learning style, past grades) via a Status Quiz/Survey.

Behavior Insights: Tracks user interactions to provide insights (e.g., time spent on materials, quiz performance).

Personalized Recommendations: Delivers tailored resources, split into General (broad topics) and Specific (targeted materials).

Learning Hub: Includes a Strategic Blueprint (learning plan) and Live Quiz (mock EUEE-style quizzes) to guide students.

Feedback: Collects user feedback to refine the system.

Flow:

- Students start with a survey/quiz to input their profiles.
- The recommendation engine uses the dataset/database to suggest materials.
- Students engage with the Learning Hub (quizzes, resources), and their interactions (behavior insights, feedback) loop back to refine recommendations.

6. Data Sources

The primary data source for this project will be historical academic records and mock exam results collected from Grade 12 students across selected Ethiopian high schools. This dataset includes students' subject-wise scores, attendance records, demographic details (such as gender and school type), and self-reported study habits. This type of data is highly relevant, as it directly reflects the academic performance and learning behavior critical for building a personalized prediction and recommendation system. To support generalization, we may also utilize publicly available educational datasets or simulated student activity logs where necessary. Prior studies, such as Tadesse et al. (2020), successfully used similar data for performance prediction in Ethiopian secondary schools, validating its applicability. Before model training, the raw data will undergo preprocessing steps including data cleaning, normalization, encoding categorical variables, handling missing values, and balancing the dataset to avoid bias. These steps are crucial to ensure the quality and consistency of the data used for developing accurate and reliable machine learning models.

7. Literature Review

Several studies provide a strong foundation for the methodology we adopt in this project. Belachew & Gobena (2017) demonstrated the effectiveness of using Naïve Bayes to predict student performance with high accuracy (95.7%) in an Ethiopian university context, validating the use of machine learning for academic predictions in local settings. Similarly, Tadesse et al. (2020) employed Gradient Boosting to analyze secondary school student data, showing that incorporating study habits and exam results can enhance prediction quality—an approach we directly adopt for our own feature selection. Additionally, Wu et al. (2015) introduced a MOOC recommendation system using collaborative filtering, which informs our plan to personalize educational content based on student behavior and preferences. Our project extends these findings by focusing on Ethiopian Grade 12 students, combining predictive modeling with personalized learning recommendations to improve outcomes for EUEE preparation, especially for students in underserved and remote areas.

Implementation Plan

1. Technology Stack

Programming Language:

- Python

Backend Framework:

- Django: For server-side logic, user authentication, and ML model integration
- Django REST Framework (DRF): To create APIs for frontend-backend communication

Frontend Framework:

- Django: For a modern, interactive user interface

Machine Learning Libraries:

- Scikit-learn: For Decision Trees, KNN, and basic ML models
- LightFM *(optional)*: For hybrid recommendations
- Pandas / NumPy: For data handling and preprocessing
- Matplotlib / Seaborn: For visualizing student performance and patterns

Database:

- SQLite (for local development) or PostgreSQL (for production)

Version Control:

- Git / GitHub

Hosting (Free Options):

- Backend:
  - Render or Railway (free plans for hosting Django + PostgreSQL)
- Database:
  - Supabase or ElephantSQL (free PostgreSQL hosting)

2. Timeline

Refer to the Gantt chart here.

3. Milestones

Milestone 1: Project Planning and Research

- Finalize project scope and objectives
- Assign roles and responsibilities
- Conduct literature review and research on personalized learning
- Identify data sources (synthetic + potential real user feedback)
- Design system architecture and tech stack

Milestone 2: Dataset Preparation & ML Model Development

- Generate and clean synthetic dataset of students
- Extract features: learning styles, behaviors, scores
- Train Decision Tree model for learning style prediction
- Train KNN model for content recommendation
- Evaluate model performance with accuracy metrics

Milestone 3: Backend Development

- Set up Django project structure
- Implement user registration and login system
- Develop APIs for:
  - User profiles
  - Quiz management
  - Content delivery
  - Behavior logging
- Integrate ML model via API

Milestone 4: Frontend Development

- Design and build user interface using Django templates
- Build dashboard for users to view recommendations and track progress

- Implement quiz interface and content viewer
- Ensure responsive and user-friendly design

Milestone 5: System Integration & Testing

- Connect frontend with backend APIs
- Ensure communication between Django and ML services
- Conduct full system testing (unit, integration, and user testing)
- Gather feedback from real users

Milestone 6: Finalization & Deployment

- Optimize performance and fix bugs
- Prepare user documentation and user guide
- Deploy system on a test hosting platform
- Final presentation preparation (slides, demo, report)

Milestone 7: Final Report & Submission

- Submit complete documentation and final report
- Deliver project presentation and demo
- Conduct project reflection and evaluation

4. Challenges and Mitigations

Development: Limited Expertise in Django Fullstack Development

Building an adaptive web app using Django fullstack may be challenging for a beginner team; we'll leverage Django's documentation, tutorials, and pre-built templates (e.g., Django REST Framework for API integration) to simplify development of the UI and Recommendation Engine integration.

Development: Implementing Real-Time Adaptivity in Django

Enabling real-time recommendation updates (e.g., after quizzes) in Django is complex; we'll use Django's ORM for efficient database queries and batch updates (e.g., daily adjustments) to reduce complexity while ensuring adaptivity.

Development: Resource Tagging for Adaptivity

Tagging 10 resources (e.g., Khan Academy videos) with difficulty and purpose requires time and expertise; we'll manually tag based on Grade 12 EUEE needs (e.g., "exam prep" for YouTube videos) and prioritize difficulty tags if time-constrained.

Implementation: Small Dataset for ML Models

A small dataset (50 historical, 10-25 fresh users) may cause overfitting in ML models; we'll use a Kaggle dataset (e.g., "Student Performance") as a fallback and apply regularization (e.g., KNN with k=5, Decision Tree pruning) to improve robustness.

Implementation: Getting Accurate Data from Students

Students may provide inaccurate data in surveys (e.g., misreporting grades, social media use), affecting ML model performance; we'll validate responses by cross-checking with mock quiz results and encourage honesty through anonymized data collection.

Implementation: Internet Access for Students

Many Grade 12 students lack reliable internet to access PALMS resources; we'll provide offline options (e.g., downloadable PDFs) and optimize the Django app for low-bandwidth usage to ensure accessibility.

Implementation: Student Engagement with PALMS

Students may not engage with quizzes or feedback, limiting adaptivity; we'll gamify the app (e.g., badges for quiz completion) and simplify feedback prompts (e.g., Yes/No responses) to encourage participation.

5. Ethical Considerations

Data Privacy and Security: We'll anonymize student data (e.g., grades, quiz scores) by using unique IDs and secure the Django app with HTTPS and CSRF protection to safeguard privacy during storage and transmission.

Informed Consent: Students will receive a clear explanation of data usage via a consent form in English and Amharic during onboarding, ensuring participation is voluntary and withdrawable.

Fairness in Recommendations: To avoid bias in ML models, we'll ensure the dataset reflects diverse student profiles and monitor recommendations to promote equitable resource allocation for all learners.

Cultural Sensitivity and Accessibility: We'll include Amharic resources, use simple English in videos, and design the UI for low-bandwidth access, respecting Ethiopia's educational and cultural context.

6. References

1. Tadesse, B., Abera, M., & Gebre, S. (2020). *Prediction of Ethiopian Secondary School Students' Academic Performance Using Machine Learning Techniques*. Ethiopian Journal of Education and Sciences, 15(2), 23-35.

→ This study effectively used student academic records and study behavior data to develop predictive models using Gradient Boosting, highlighting the relevance of such data in an Ethiopian context.

2. Belachew, D., & Gobena, L. (2017). *Student Performance Prediction Using Naïve Bayes at Wolkite University*. Proceedings of the 9th Annual Conference on Computing and ICT Research, Addis Ababa.
   → Demonstrated high accuracy in predicting university student performance using minimal academic datasets, further confirming the utility of academic records for ML-based predictions in Ethiopia.

3. Kotsiantis, S. B., Pierrakeas, C., & Pintelas, P. (2004). *Predicting Students' Performance in Distance Learning Using Machine Learning Techniques*. Applied Artificial Intelligence, 18(5), 411–426.
   → A widely cited work that supports the use of academic performance data in building intelligent recommendation and prediction systems in education.