Heaven’s Light Is Our Guide

Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

A report on “Project Title”

Course Code: CSE 3200

Course Title: Software Development Project II

Authors

MD. Rakibul Hasan Rifat

2103057

section A

Supervisor

(Supervisor’s Name)

(Supervisor's Designation)

Dept. of CSE, RUET

Date: dd/mm/yyyy

Heaven's Light Is Our Guide

Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

Certification

This is to certify that the project titled: "***Insert Project Title Here***" submitted by ***[Student Name(s)]***, Roll Number(s) ***[Insert Roll. No.]***, for the fulfillment of the requirements of the course CSE 3200 (Software Development Project II), has been examined and approved by the undersigned.

The project has been carried out under my supervision, and it is recommended for submission and evaluation.

Supervisor:
Name: ______________________________
Designation: _______________________
Department: ________________________
Institution: _________________________
Date: ______________________________

Signature: ________________________

Contents

Page No.

Abstract

(250–300 words)

Summarizes problems, method, results, and contribution in one concise paragraph.

List of Figures

Auto-generated list of all figures.

List of Tables

Auto-generated list of all tables.

Chapters start from here

Chapter 1
Introduction

1.1 Problem Statement

(Define the exact problem the project solves.)

1.2 Motivation

(Describe the broader domain where the problem exists.)

1.3 Objectives

(List the measurable goals the project aims to achieve.)

1.4 Contribution Summary

(*Highlight the key contributions of your work.)*

1.5 Report Structure

(Briefly describe what each chapter contains.)

Chapter 2
Background & Literature Review

2.1 Theoretical Foundation

LearnTube's design is grounded in established educational and technological theories. Personalized learning tailors experiences to individual needs, pace, and preferences through adaptive learning paths, learner modeling, and content recommendation (DOE, 2017). The platform implements these via AI-generated roadmaps and category-based content curation.

Recommendation systems employ multiple approaches: content-based filtering (keyword matching), collaborative filtering (user patterns), knowledge-based systems (AI roadmaps), and hybrid systems (Gemini + Perplexity integration). Large Language Models enable contextual understanding, generative content creation, and conversational tutoring—Gemini generates keywords while Perplexity powers recommendations and chat interactions.

Mobile learning (m-Learning) theory supports ubiquitous access, microlearning, and context-aware delivery. React Native and Expo enable cross-platform experiences. Backend-as-a-Service (BaaS) architecture provides serverless computing, API-first design, and real-time synchronization through Appwrite for authentication, database, and storage.

2.2 Related Work Review

Video-Based Learning Platforms: YouTube offers vast educational content but lacks structured paths and optimizes for engagement over learning. Coursera provides structured university courses with certifications but is primarily paid and rigid. Khan Academy delivers free mastery-based learning with progress tracking but limited to specific subjects without external content integration. Udemy offers diverse courses but with variable quality and no adaptive features.

AI-Powered Learning Assistants: Duolingo uses adaptive AI for language learning with gamification but limited to languages. Carnegie Learning provides intelligent math tutoring with real-time analysis but requires institutional adoption. Socratic by Google offers visual problem recognition and explanations but focuses on homework help rather than structured learning.

Research Findings: Studies demonstrate personalized content sequencing improves outcomes (Brusilovsky & Millán, 2007), context-aware mobile learning increases engagement (Chen & Huang, 2012), video length impacts learning effectiveness (Thai et al., 2017), AI enhances personalization and assessment (Zawacki-Richter et al., 2019), and transformer models enable educational content generation (Ouyang & Jiao, 2021).

2.3 Comparative Analysis

(Compare existing methods and highlight their limitations.)

2.4 Gap Analysis

(Clearly identify what the current literature fails to solve.)

Chapter 3
Requirements & Team Workflow

3.1 Functional Requirements

(List what the system must do from a user perspective.)

3.2 Non-functional Requirements

(Specify performance, security, usability, and other constraints.)

3.3 System Constraints

(Mention hardware, software, legal, or environmental constraints.)

3.4 Technology Stack Justification

(Explain why specific tools, languages, or frameworks were chosen.)

3.5 Team Structure & Individual Responsibilities

(Define each member's role and actual work contribution.)

3.6 Collaboration Workflow

(Present Trello/Jira boards showing task assignments and sprint progress.)

Chapter 4
Project Management & Finance

4.1 Work Breakdown Structure (WBS)

(Break the entire project into hierarchical tasks.)

4.2 Project Schedule — Gantt Chart

(Show timeline, deadlines, milestones, and dependencies.)

4.3 Budget & Financial Cost Analysis

(Document all expenses, estimations, and financial planning.)

4.4 Risk Analysis & Mitigation Plan

(Identify risks and describe how they are handled.)

4.5 Summary of Management Decisions

(Highlight the important decisions taken during project execution.)

Chapter 5
System Design & Architecture

5.1 Methodological Structure

(Present the top-level structure of the full system.)

5.2 High-Level Architecture Diagram

(Show how different modules communicate or exchange data.)

5.3 Module Interaction Diagram

(Describe the broader domain where the problem exists.)

5.4 Database Schema / ER Diagram

(Present database structure and relationships.)

5.5 Interface Specifications / API Contracts

(Define endpoints, request-response formats, and APIs.)

5.6 Security & Performance Design

(Explain measures taken for secure and efficient operation.)

Chapter 6
Implementation

6.1 Development Environment Setup

(List the environments, OS, compilers, and configurations used.)

6.2 Coding Standards & Version Control Strategy

(Explain naming conventions, formatting, and Git workflow model.)

6.3 Module-Wise Implementation

(Describe each module's logic, flow, and code snippets.)

6.3.x Module Ownership Block

(Include developer's name, tasks completed, and Git commit evidence.)

6.4 System Integration Workflow

(Explain how all modules were integrated and tested together.)

6.5 Performance, Optimization, and Security Techniques

(Discuss performance tuning and security improvements applied.)

Chapter 7
Testing, Results & Evaluation

7.1 Testing Strategy

(State whether unit, integration, system, or acceptance testing was used.)

7.2 Test Case Tables

(List inputs, expected outputs, and actual outputs.)

7.3 Performance Evaluation

(Present speed, accuracy, throughput, or resource usage metrics etc.)

7.4 Comparison with Baseline or Existing Systems

(Compare your system's performance with alternatives.)

7.5 Screenshots & Execution Results

(Show working screenshots of the actual system.)

Chapter 8
Discussion & Analysis

8.1 Interpretation of Findings

(Explain what the results mean in a real context.)

8.2 Strengths of the Project

(Discuss what your system does well compared to others.)

8.3 Limitations

(State clearly what the system cannot do yet.)

8.4 Recommendations for Future Development

(Suggest realistic improvements for next iterations.)

8.5 Reflection on Design & Implementation Decisions

(Explain why your design choices were justified.)

Chapter 9
Life-long Learning Impact

9.1 Technical & Professional Skills Acquired

(List long-term competencies learned during the project.)

9.2 Learning New Technologies & Tools

(Describe how the team adapted to unfamiliar tools or methodologies.)

9.3 Future Growth & Directions

(Provide a forward vision for the work.)

Chapter 10
Conclusion

10.1 Summary of Work

(Recap what the project achieved overall.)

10.2 Achievement of Objectives

(Show how each objective was met.)

10.3 Final Remarks

(Close with the significance and value of the project.)