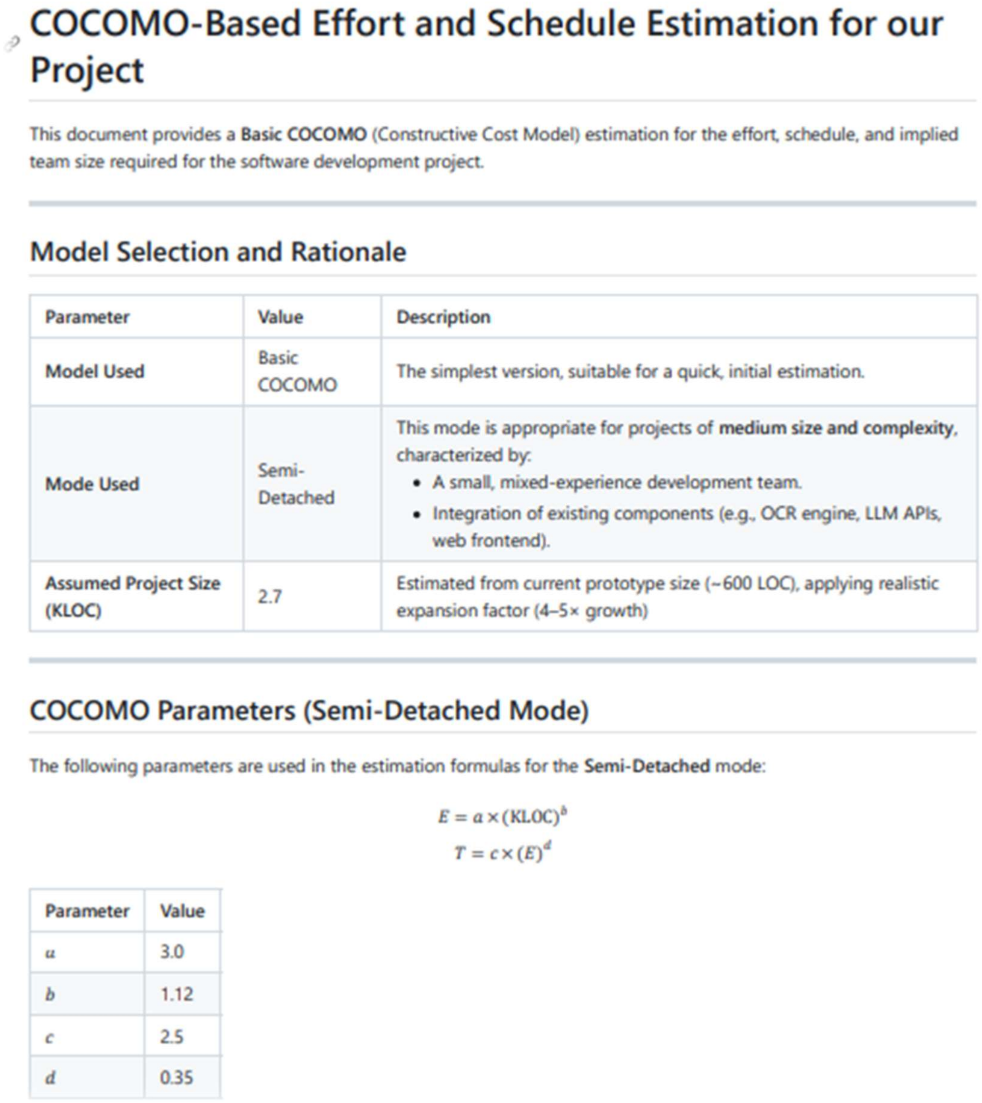


COCOMO-Based Effort and Schedule Estimation for our Project

This document provides a **Basic COCOMO** (Constructive Cost Model) estimation for the effort, schedule, and implied team size required for the software development project.

Model Selection and Rationale

Parameter	Value	Description
Model Used	Basic COCOMO	The simplest version, suitable for a quick, initial estimation.
Mode Used	Semi-Detached	This mode is appropriate for projects of **medium size and complexity**, characterized by: • A small, mixed-experience development team. • Integration of existing components (e.g., OCR engine, LLM APIs, web frontend).
Assumed Project Size (KLOC)	2.7	Estimated from current prototype size (~600 LOC), applying realistic expansion factor (4–5× growth)

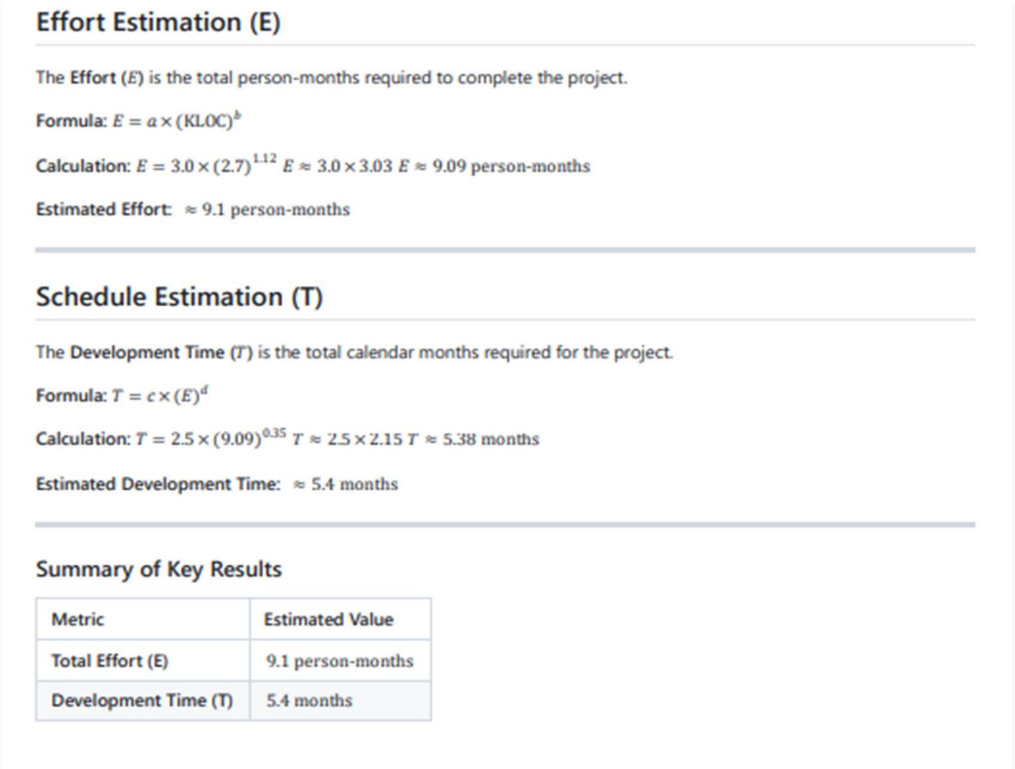
COCOMO Parameters (Semi-Detached Mode)

The following parameters are used in the estimation formulas for the **Semi-Detached** mode:

$$E = a \times (\text{KLOC})^{b}$$

$$T = c \times (E)^{d}$$

Parameter	Value
a	3.0
b	1.12
c	2.5
d	0.35



Effort Estimation (E)

The **Effort** (E) is the total person-months required to complete the project.

Formula: $E = a \times (\text{KLOC})^{b}$

Calculation: $E = 3.0 \times (2.7)^{1.12}$ $E \approx 3.0 \times 3.03$ $E \approx 9.09$ person-months

Estimated Effort: ≈ 9.1 person-months

Schedule Estimation (T)

The **Development Time** (T) is the total calendar months required for the project.

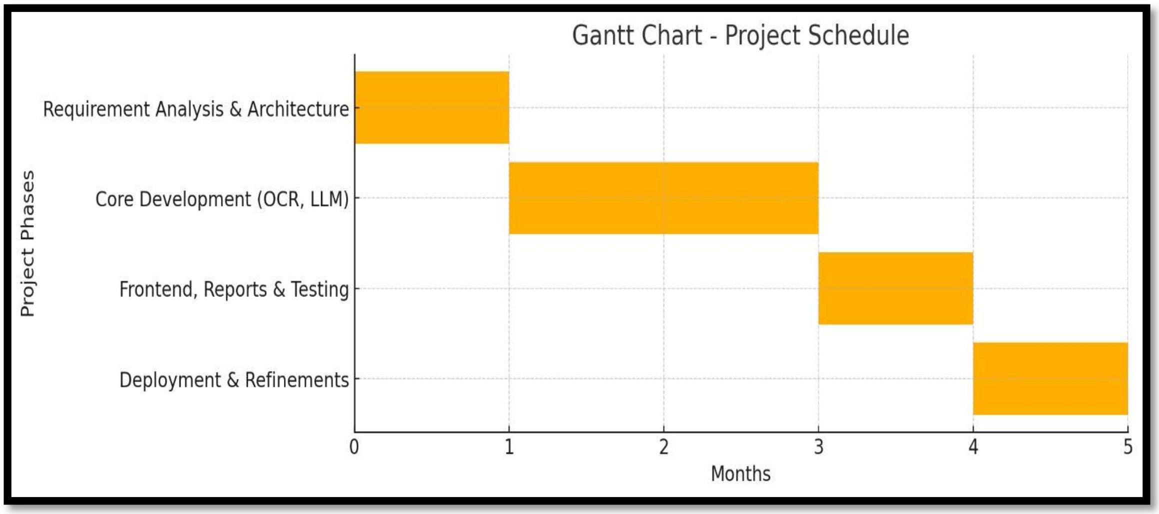
Formula: $T = c \times (E)^{d}$

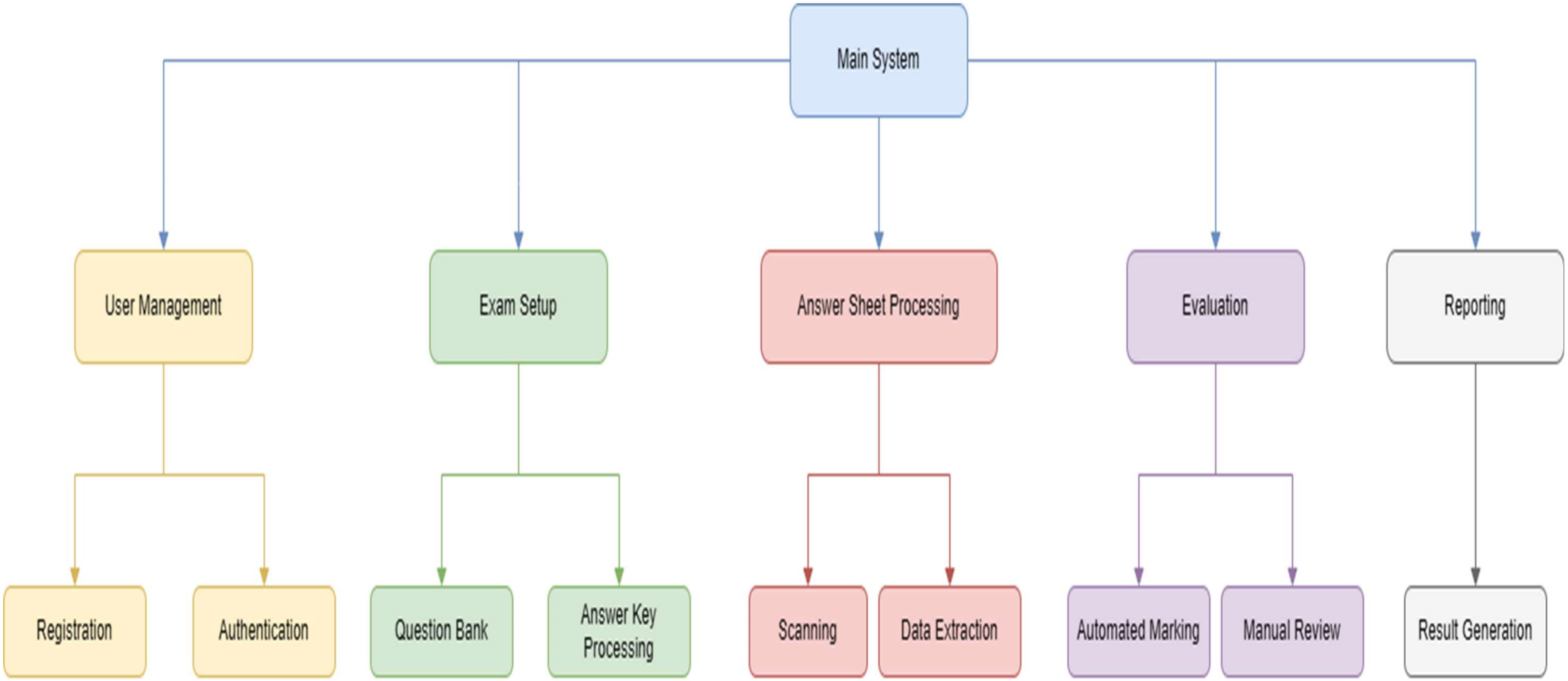
Calculation: $T = 2.5 \times (9.09)^{0.35}$ $T \approx 2.5 \times 2.15$ $T \approx 5.38$ months

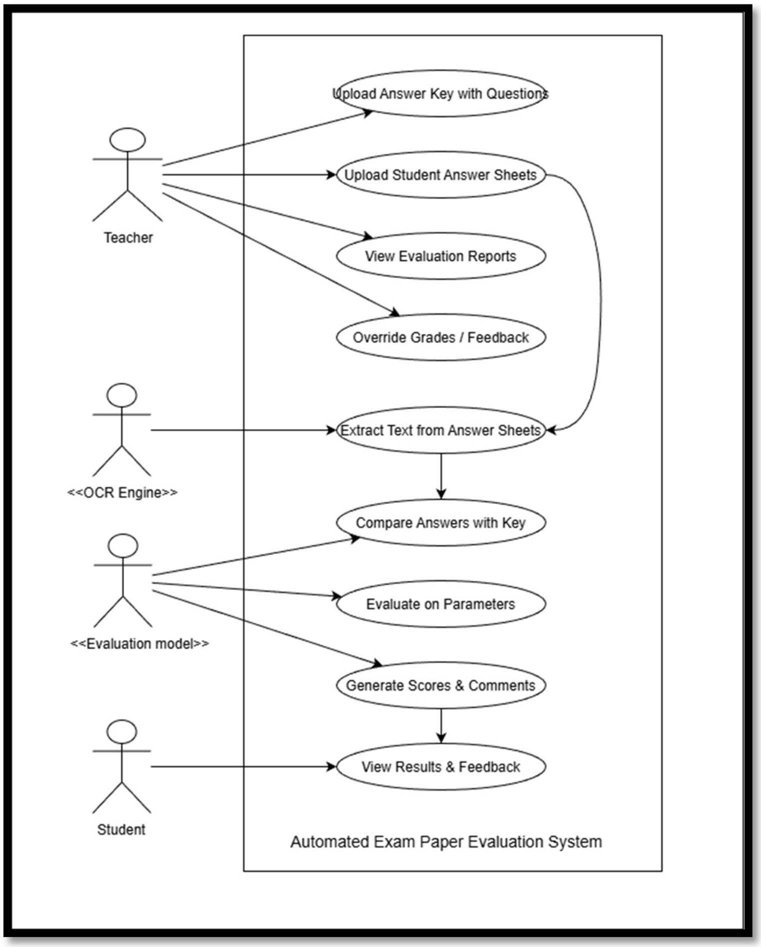
Estimated Development Time: ≈ 5.4 months

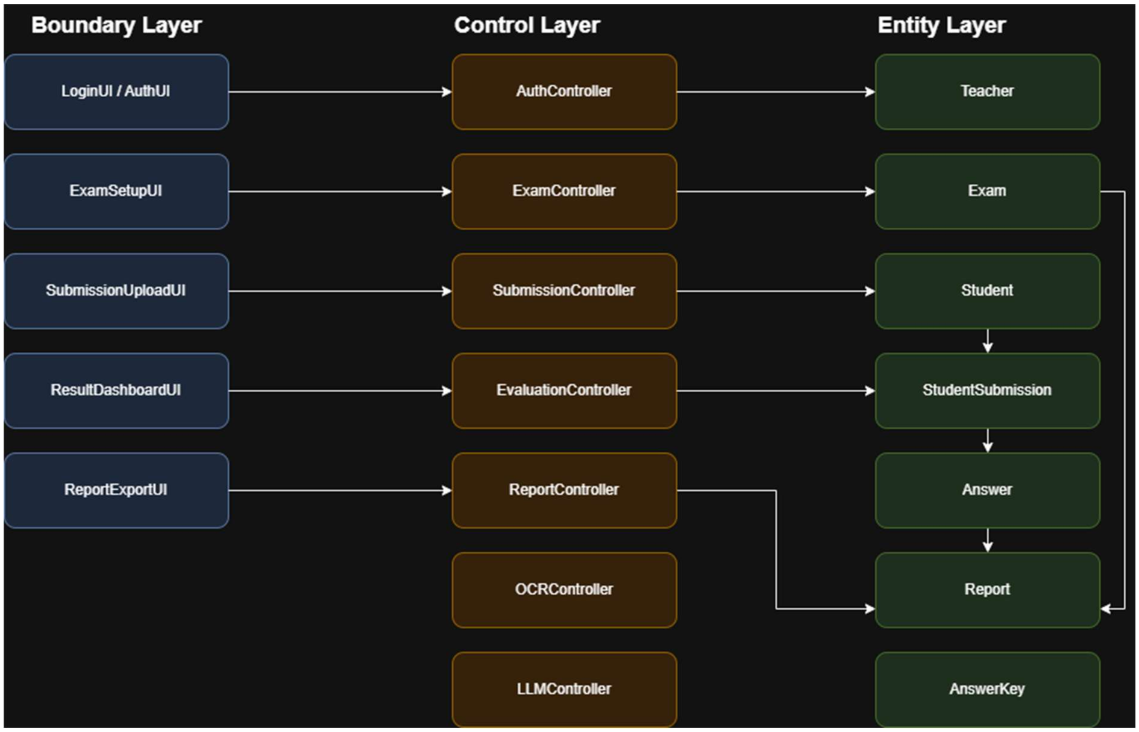
Summary of Key Results

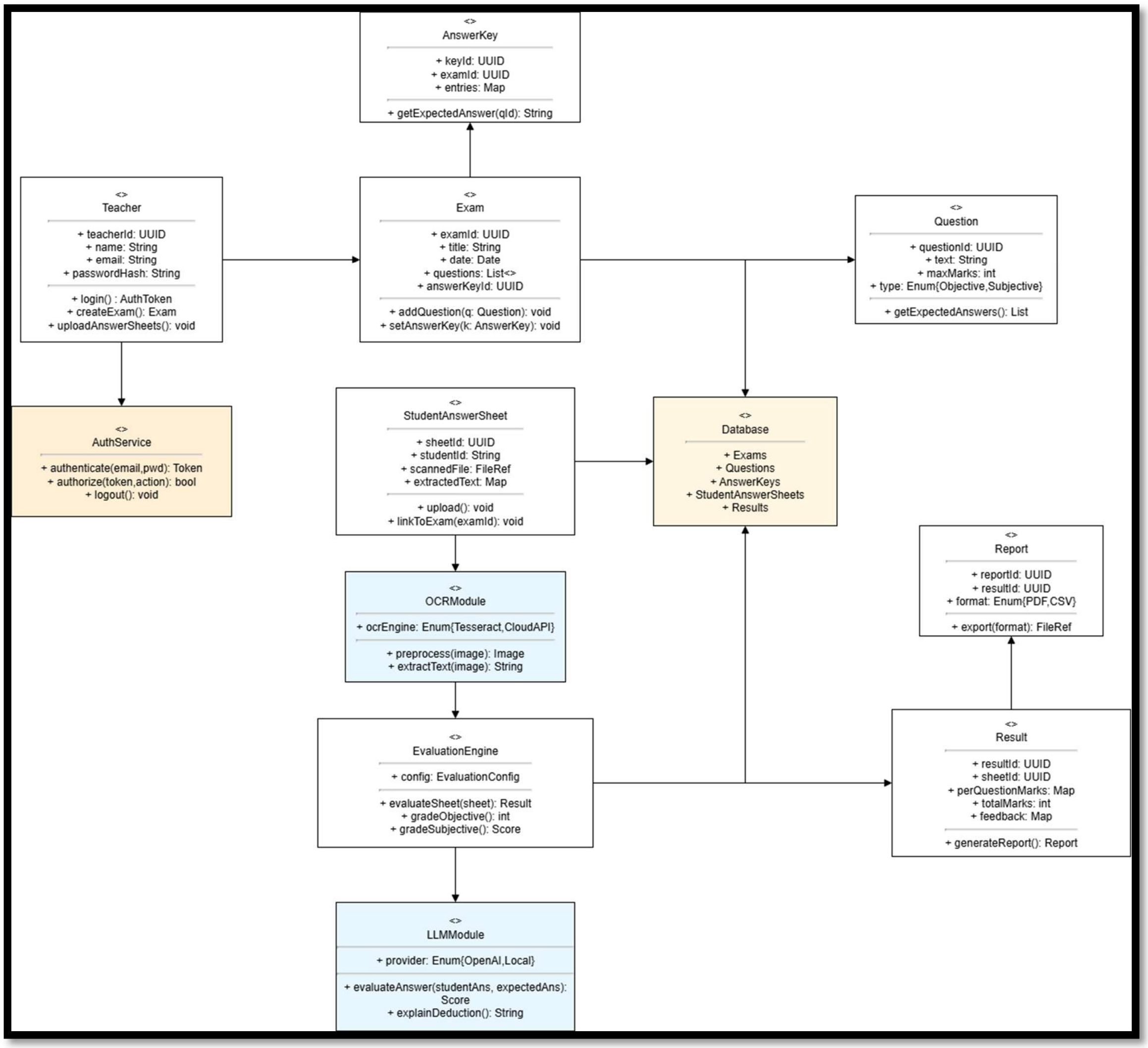
Metric	Estimated Value
Total Effort (E)	9.1 person-months
Development Time (T)	5.4 months


Gantt Chart - Project Schedule
Project Phases
Requirement Analysis & Architecture
Core Development (OCR, LLM)
Frontend, Reports & Testing
Deployment & Refinements
0
1
2
3
4
5
Months

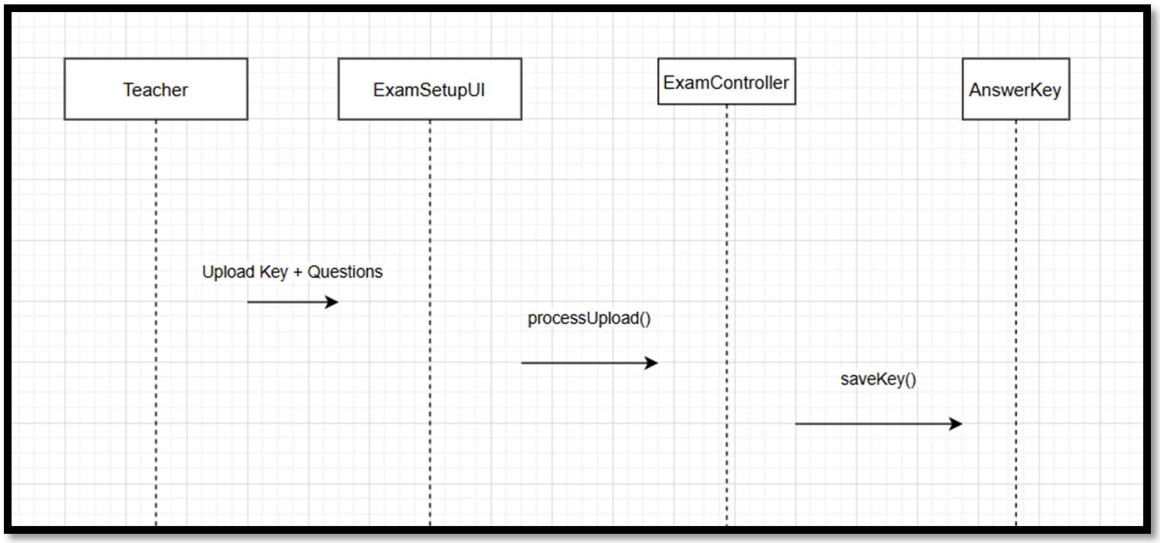

Main System
User Management
Exam Setup
Answer Sheet Processing
Evaluation
Reporting
Registration
Authentication
Question Bank
Answer Key Processing
Scanning
Data Extraction
Automated Marking
Manual Review
Result Generation

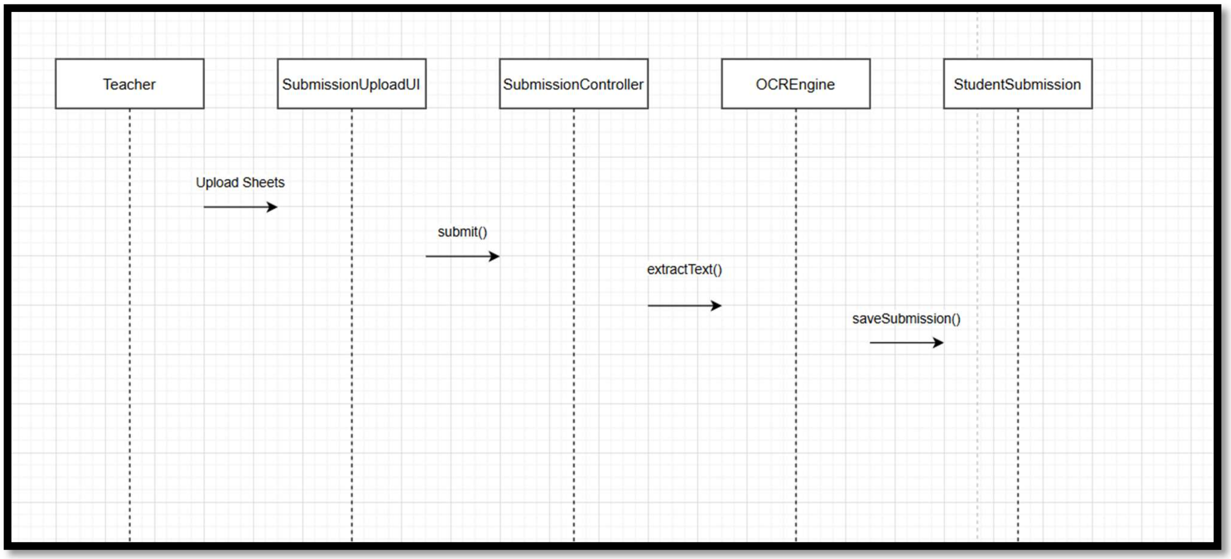

Upload Answer Key with Questions
Teacher
Upload Student Answer Sheets
View Evaluation Reports
Override Grades / Feedback
<<OCR Engine>>
Extract Text from Answer Sheets
Compare Answers with Key
<<Evaluation model>>
Evaluate on Parameters
Generate Scores & Comments
View Results & Feedback
Student
Automated Exam Paper Evaluation System

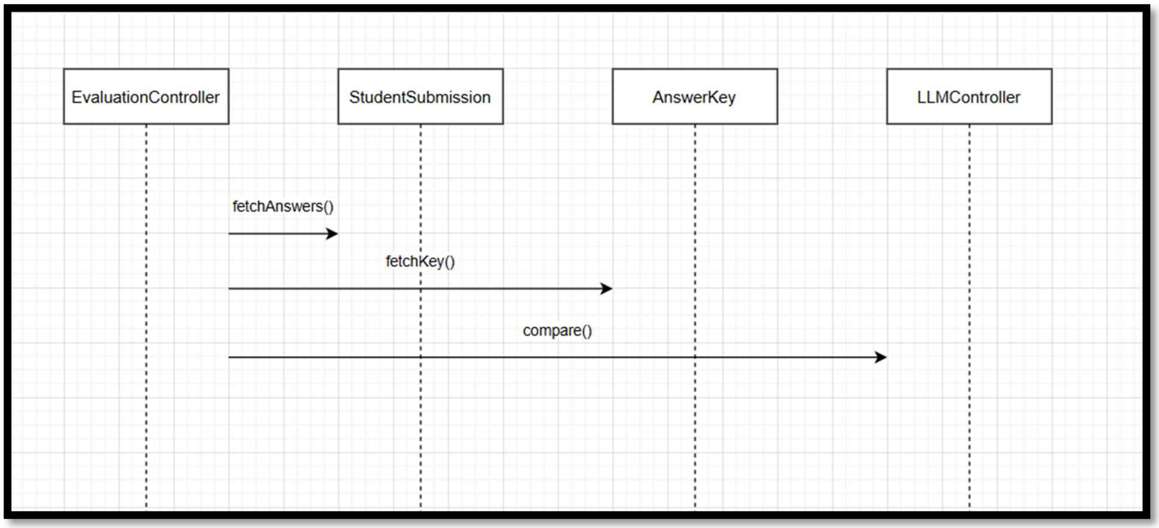

Boundary Layer
Control Layer
Entity Layer
LoginUI / AuthUI
AuthController
Teacher
ExamSetupUI
ExamController
Exam
SubmissionUploadUI
SubmissionController
Student
ResultDashboardUI
EvaluationController
StudentSubmission
ReportExportUI
ReportController
Answer
OCRController
Report
LLMController
AnswerKey

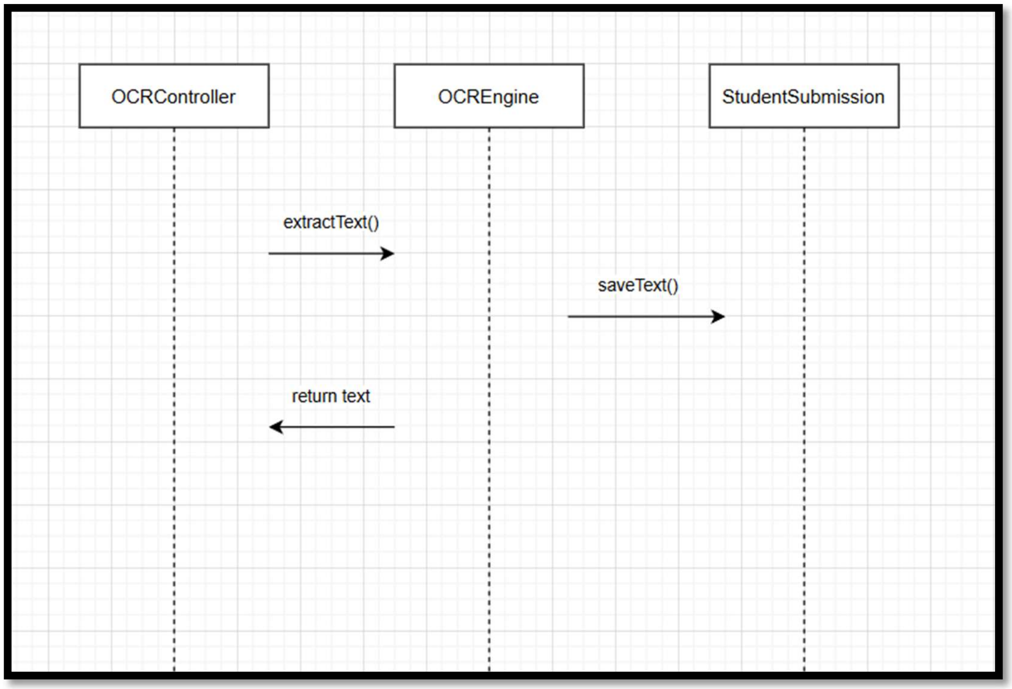


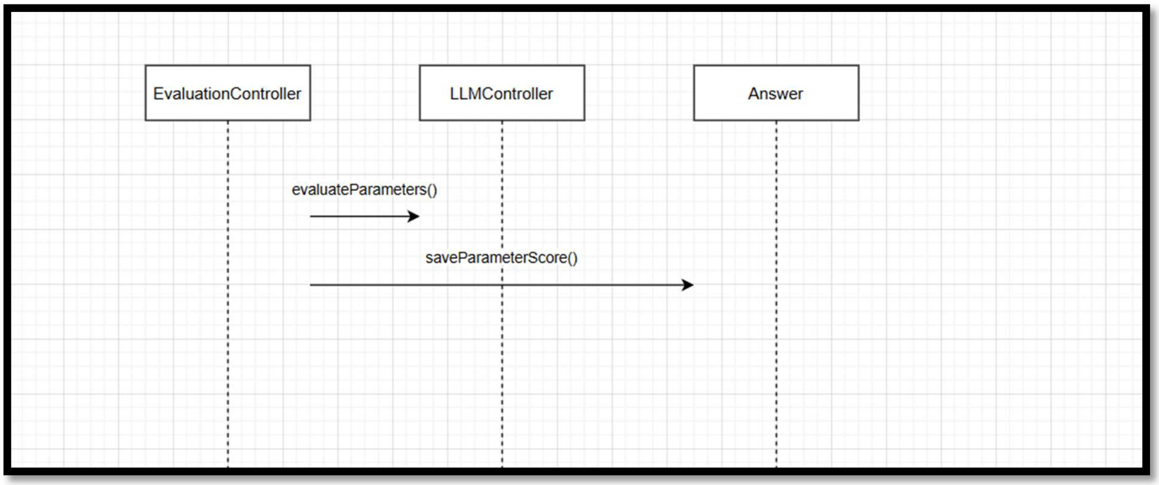
<>
AnswerKey
+ keyId: UUID
+ examId: UUID
+ entries: Map
+ getExpectedAnswer(qId): String
<>
Teacher
+ teacherId: UUID
+ name: String
+ email: String
+ passwordHash: String
+ login() : AuthToken
+ createExam(): Exam
+ uploadAnswerSheets(): void
<>
Exam
+ examId: UUID
+ title: String
+ date: Date
+ questions: List<>
+ answerKeyId: UUID
+ addQuestion(q: Question): void
+ setAnswerKey(k: AnswerKey): void
<>
Question
+ questionId: UUID
+ text: String
+ maxMarks: int
+ type: Enum{Objective,Subjective}
+ getExpectedAnswers(): List
<>
AuthService
+ authenticate(email,pwd): Token
+ authorize(token,action): bool
+ logout(): void
<>
StudentAnswerSheet
+ sheetId: UUID
+ studentId: String
+ scannedFile: FileRef
+ extractedText: Map
+ upload(): void
+ linkToExam(examId): void
<>
Database
+ Exams
+ Questions
+ AnswerKeys
+ StudentAnswerSheets
+ Results
<>
Report
+ reportId: UUID
+ resultId: UUID
+ format: Enum{PDF,CSV}
+ export(format): FileRef
<>
OCRModule
+ ocrEngine: Enum{Tesseract,CloudAPI}
+ preprocess(image): Image
+ extractText(image): String
<>
EvaluationEngine
+ config: EvaluationConfig
+ evaluateSheet(sheet): Result
+ gradeObjective(): int
+ gradeSubjective(): Score
<>
Result
+ resultId: UUID
+ sheetId: UUID
+ perQuestionMarks: Map
+ totalMarks: int
+ feedback: Map
+ generateReport(): Report
<>
LLMModule
+ provider: Enum{OpenAI,Local}
+ evaluateAnswer(studentAns, expectedAns): Score
+ explainDeduction(): String

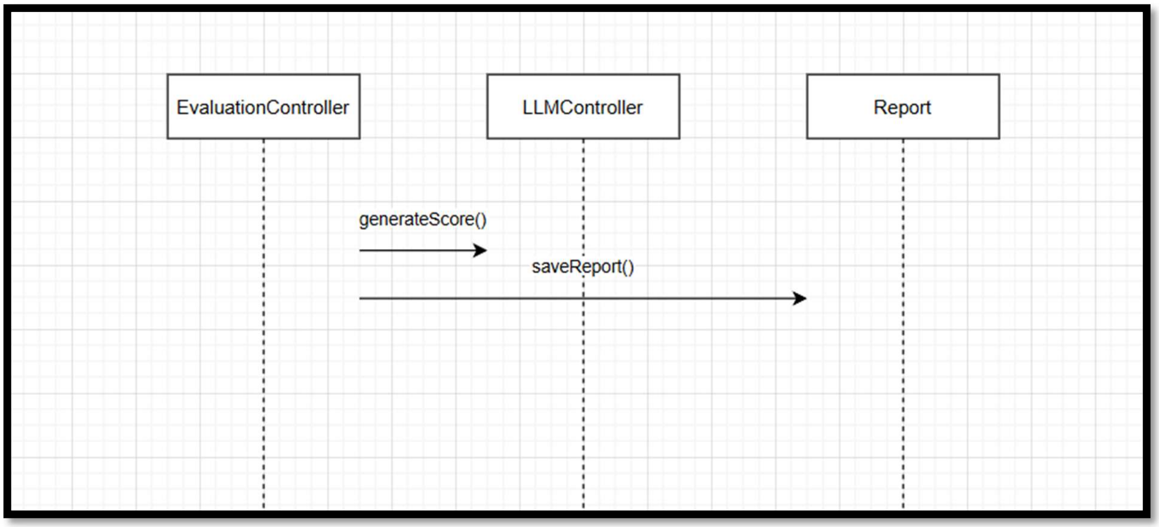

Teacher
ExamSetupUI
ExamController
AnswerKey
Upload Key + Questions
processUpload()
saveKey()

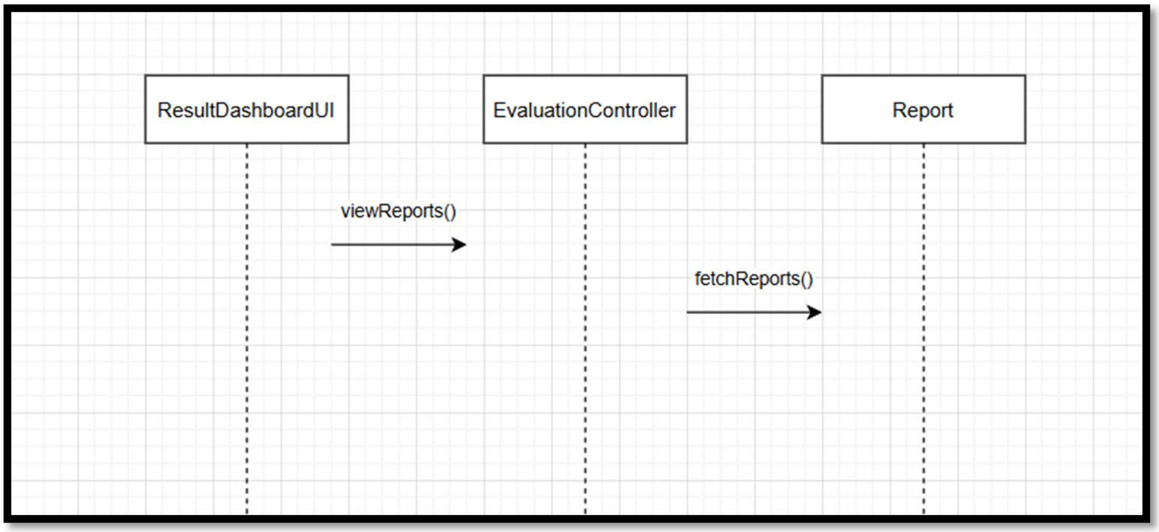

Teacher
SubmissionUploadUI
SubmissionController
OCREngine
StudentSubmission
Upload Sheets
submit()
extractText()
saveSubmission()

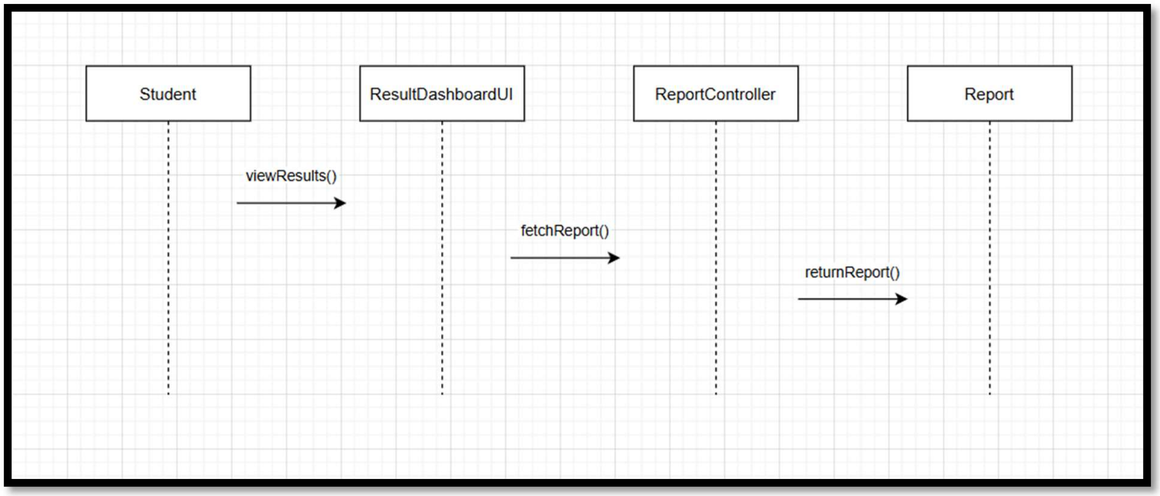

EvaluationController
StudentSubmission
AnswerKey
LLMController
fetchAnswers()
fetchKey()
compare()

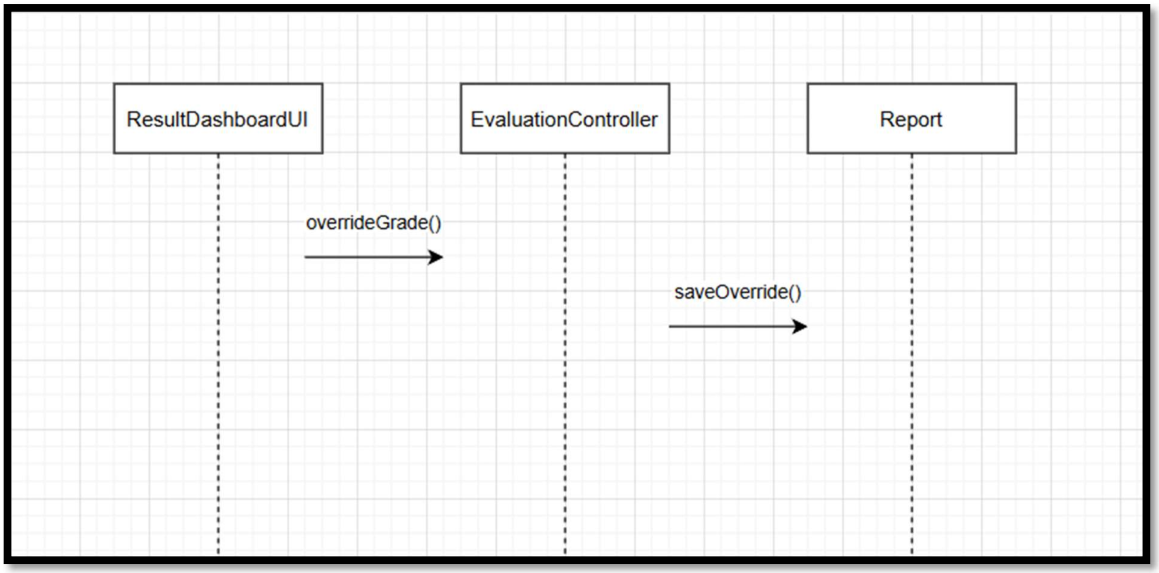

OCRController
OCREngine
StudentSubmission
extractText()
saveText()
return text

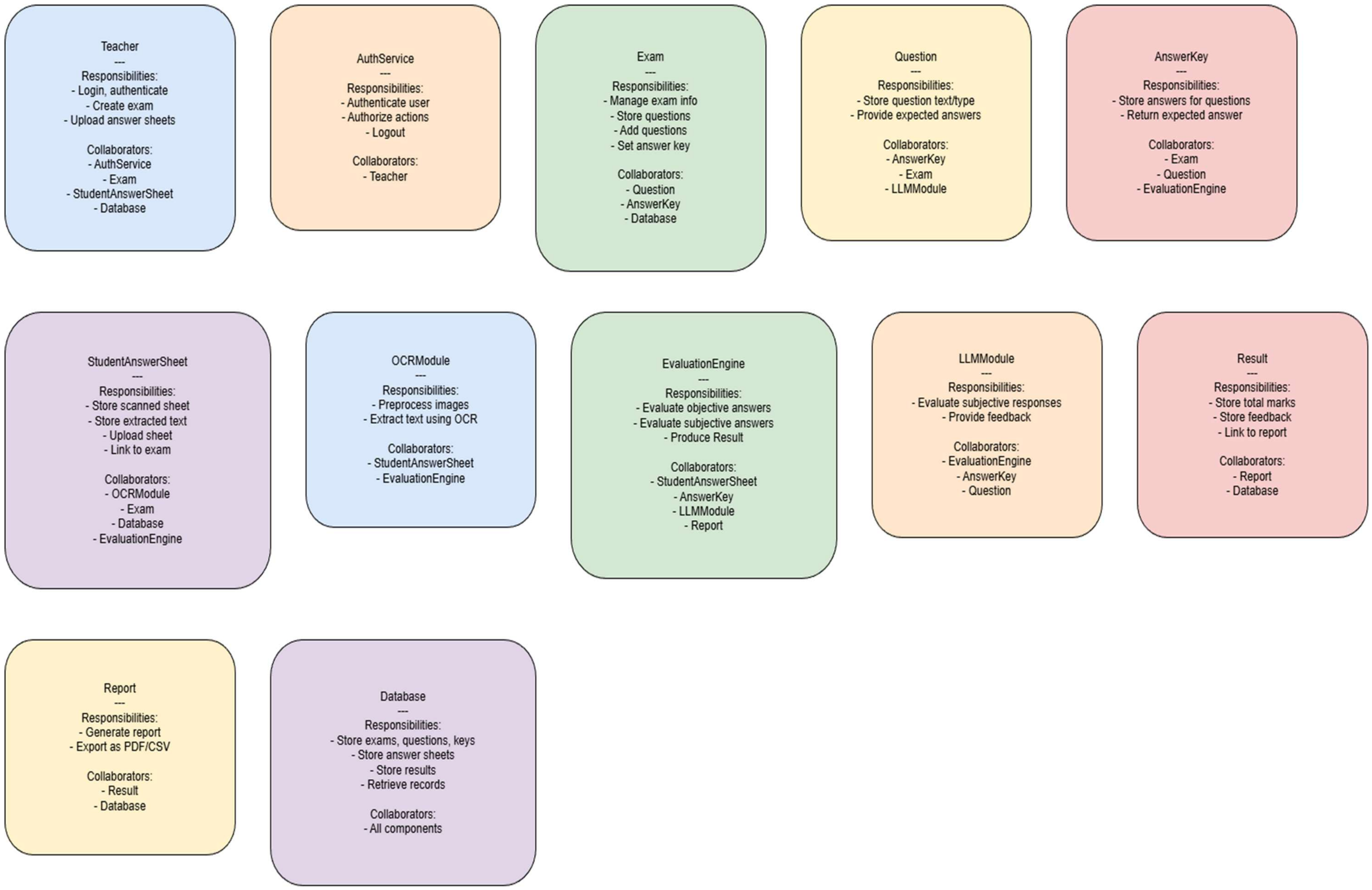

EvaluationController
LLMController
Answer
evaluateParameters()
saveParameterScore()


EvaluationController
LLMController
Report
generateScore()
saveReport()


ResultDashboardUI
EvaluationController
Report
viewReports()
fetchReports()


Student
ResultDashboardUI
ReportController
Report
viewResults()
fetchReport()
returnReport()


ResultDashboardUI
EvaluationController
Report
overrideGrade()
saveOverride()



Teacher

Responsibilities:
- Login, authenticate
- Create exam
- Upload answer sheets

Collaborators:
- AuthService
- Exam
- StudentAnswerSheet
- Database

AuthService

Responsibilities:
- Authenticate user
- Authorize actions
- Logout

Collaborators:
- Teacher

Exam

Responsibilities:
- Manage exam info
- Store questions
- Add questions
- Set answer key

Collaborators:
- Question
- AnswerKey
- Database

Question

Responsibilities:
- Store question text/type
- Provide expected answers

Collaborators:
- AnswerKey
- Exam
- LLMModule

AnswerKey

Responsibilities:
- Store answers for questions
- Return expected answer

Collaborators:
- Exam
- Question
- EvaluationEngine

StudentAnswerSheet

Responsibilities:
- Store scanned sheet
- Store extracted text
- Upload sheet
- Link to exam

Collaborators:
- OCRModule
- Exam
- Database
- EvaluationEngine

OCRModule

Responsibilities:
- Preprocess images
- Extract text using OCR

Collaborators:
- StudentAnswerSheet
- EvaluationEngine

EvaluationEngine

Responsibilities:
- Evaluate objective answers
- Evaluate subjective answers
- Produce Result

Collaborators:
- StudentAnswerSheet
- AnswerKey
- LLMModule
- Report

LLMModule

Responsibilities:
- Evaluate subjective responses
- Provide feedback

Collaborators:
- EvaluationEngine
- AnswerKey
- Question

Result

Responsibilities:
- Store total marks
- Store feedback
- Link to report

Collaborators:
- Report
- Database

Report

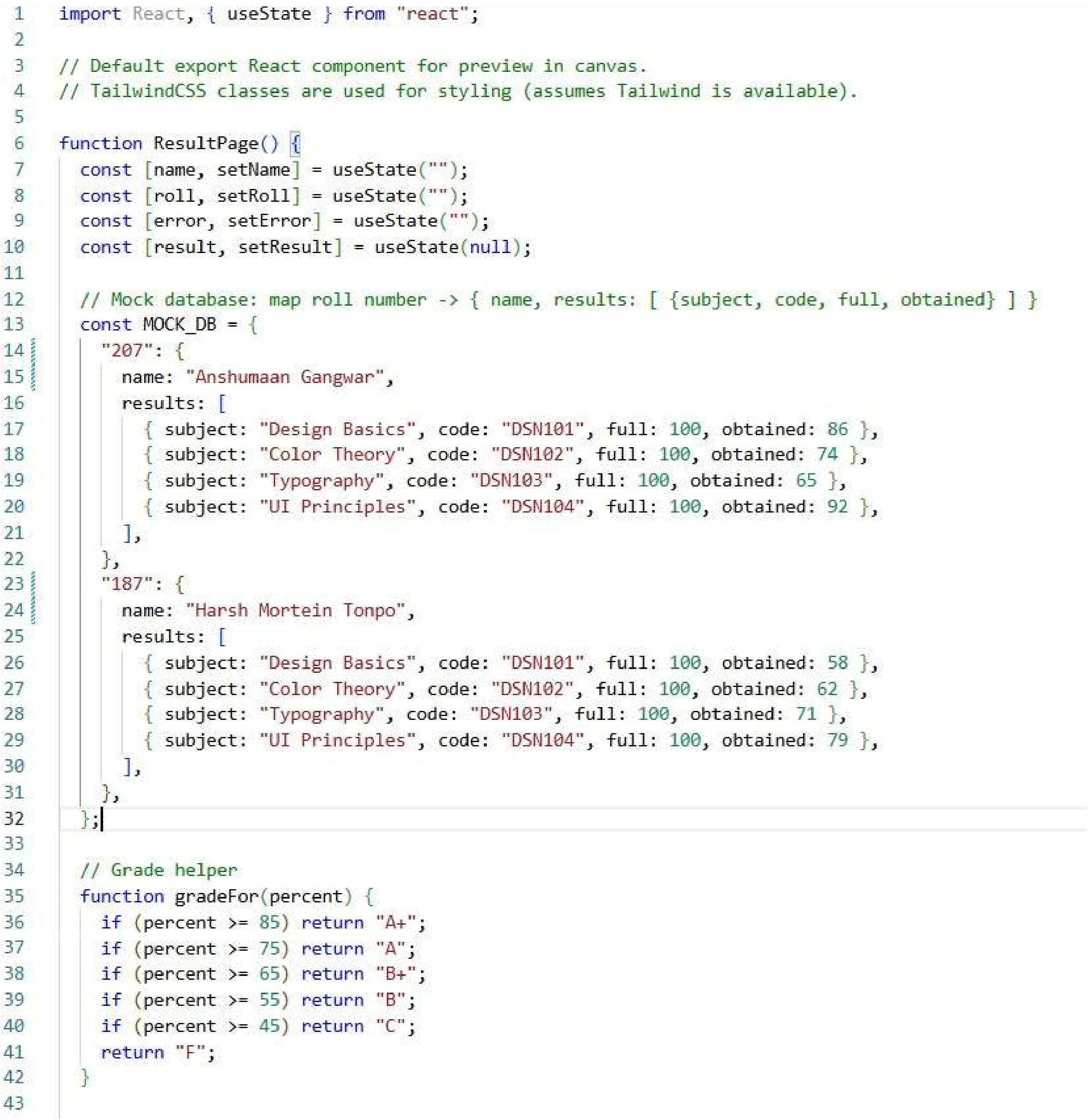
Responsibilities:
- Generate report
- Export as PDF/CSV

Collaborators:
- Result
- Database

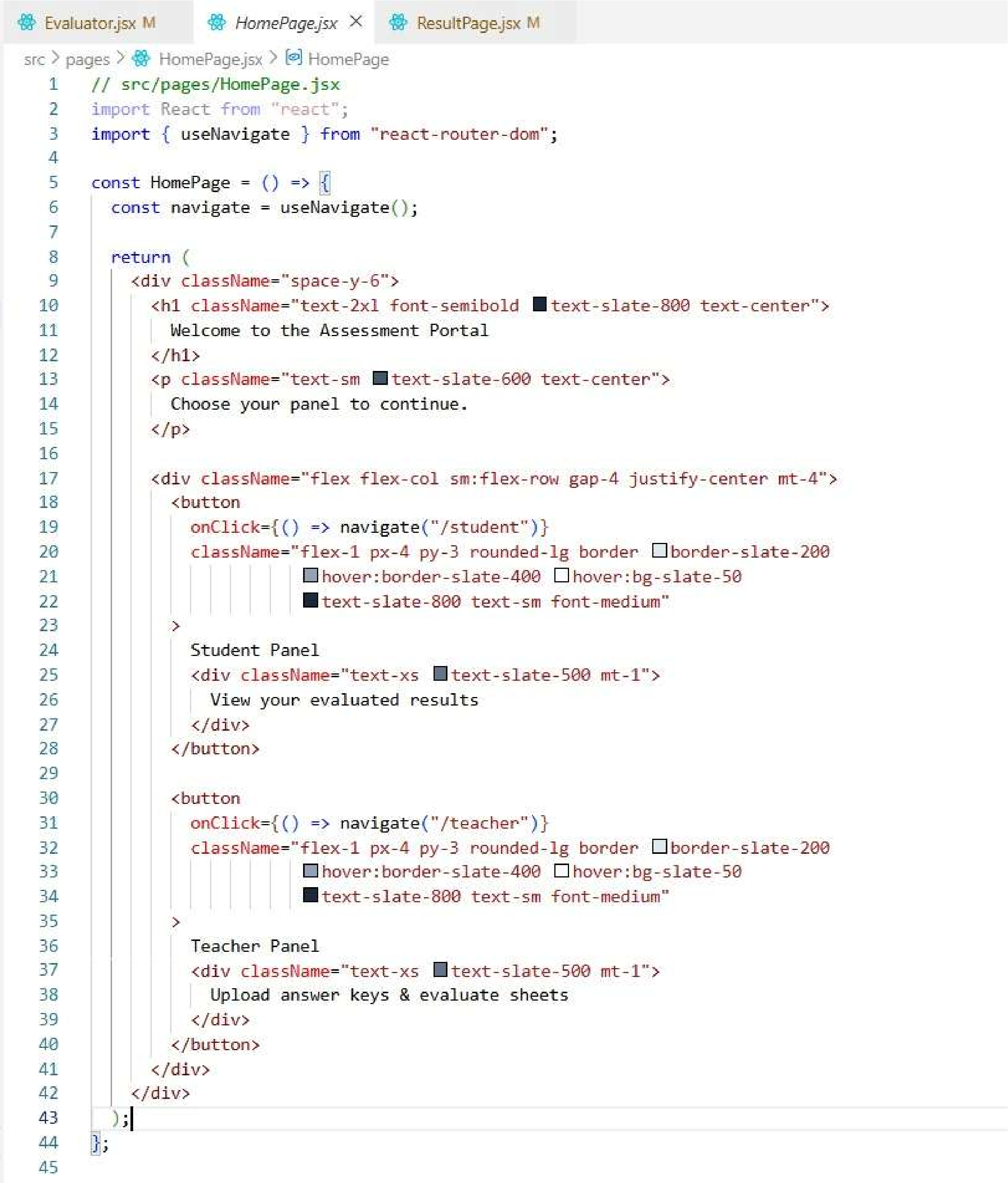
Database

Responsibilities:
- Store exams, questions, keys
- Store answer sheets
- Store results
- Retrieve records

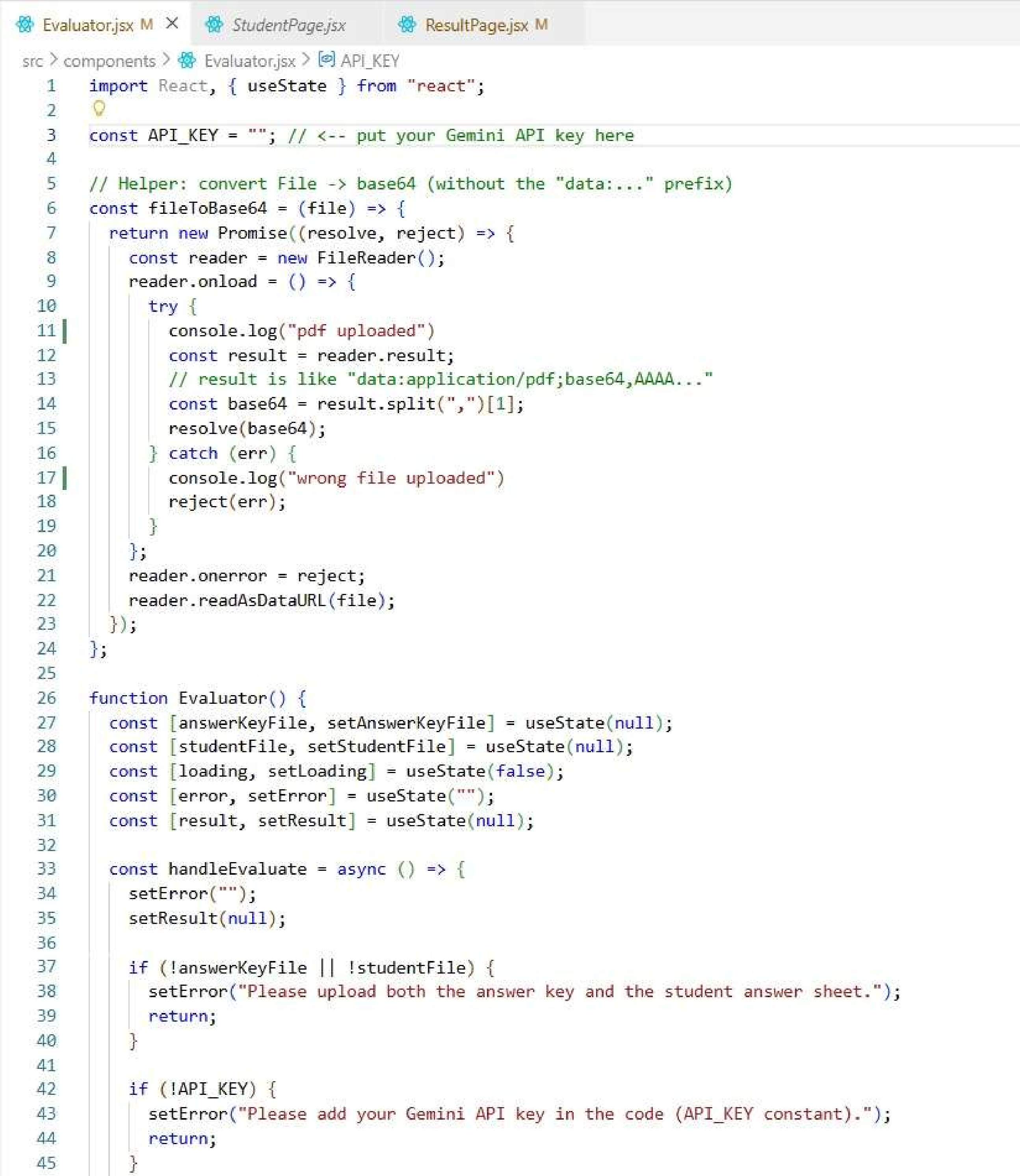
Collaborators:
- All components



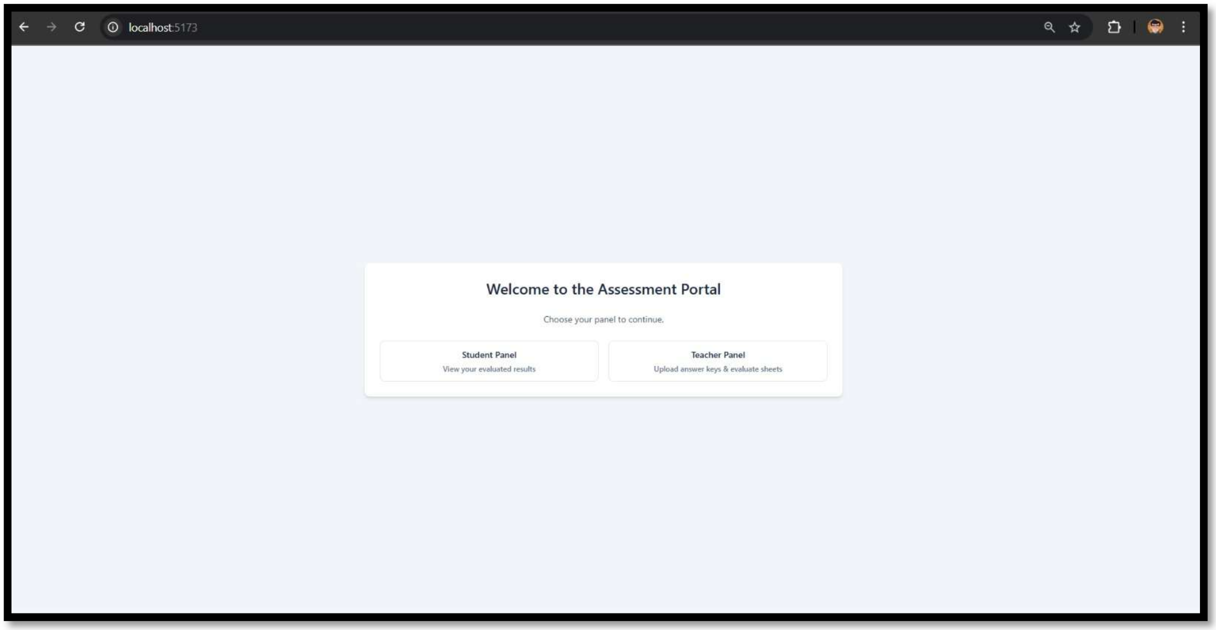
```
import React, { useState } from "react";

// Default export React component for preview in canvas.
// TailwindCSS classes are used for styling (assumes Tailwind is available).

function ResultPage() {
  const [name, setName] = useState("");
  const [roll, setRoll] = useState("");
  const [error, setError] = useState("");
  const [result, setResult] = useState(null);

  // Mock database: map roll number -> { name, results: [ {subject, code, full, obtained} ] }
  const MOCK_DB = {
    "207": {
      name: "Anshumaan Gangwar",
      results: [
        { subject: "Design Basics", code: "DSN101", full: 100, obtained: 86 },
        { subject: "Color Theory", code: "DSN102", full: 100, obtained: 74 },
        { subject: "Typography", code: "DSN103", full: 100, obtained: 65 },
        { subject: "UI Principles", code: "DSN104", full: 100, obtained: 92 },
      ],
    },
    "187": {
      name: "Harsh Mortein Tonpo",
      results: [
        { subject: "Design Basics", code: "DSN101", full: 100, obtained: 58 },
        { subject: "Color Theory", code: "DSN102", full: 100, obtained: 62 },
        { subject: "Typography", code: "DSN103", full: 100, obtained: 71 },
        { subject: "UI Principles", code: "DSN104", full: 100, obtained: 79 },
      ],
    },
  };

  // Grade helper
  function gradeFor(percent) {
    if (percent >= 85) return "A+";
    if (percent >= 75) return "A";
    if (percent >= 65) return "B+";
    if (percent >= 55) return "B";
    if (percent >= 45) return "C";
    return "F";
  }

```



```jsx
// src/pages/HomePage.jsx
import React from "react";
import { useNavigate } from "react-router-dom";

const HomePage = () => {
  const navigate = useNavigate();

  return (
    <div className="space-y-6">
      <h1 className="text-2xl font-semibold text-slate-800 text-center">
        Welcome to the Assessment Portal
      </h1>
      <p className="text-sm text-slate-600 text-center">
        Choose your panel to continue.
      </p>

      <div className="flex flex-col sm:flex-row gap-4 justify-center mt-4">
        <button
          onClick={() => navigate("/student")}
          className="flex-1 px-4 py-3 rounded-lg border border-slate-200
                     hover:border-slate-400 hover:bg-slate-50
                     text-slate-800 text-sm font-medium"
        >
          Student Panel
          <div className="text-xs text-slate-500 mt-1">
            View your evaluated results
          </div>
        </button>

        <button
          onClick={() => navigate("/teacher")}
          className="flex-1 px-4 py-3 rounded-lg border border-slate-200
                     hover:border-slate-400 hover:bg-slate-50
                     text-slate-800 text-sm font-medium"
        >
          Teacher Panel
          <div className="text-xs text-slate-500 mt-1">
            Upload answer keys & evaluate sheets
          </div>
        </button>
      </div>
    </div>
  );
};

```



```jsx
import React, { useState } from "react";

const API_KEY = ""; // <-- put your Gemini API key here

// Helper: convert File -> base64 (without the "data:..." prefix)
const fileToBase64 = (file) => {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.onload = () => {
      try {
        console.log("pdf uploaded")
        const result = reader.result;
        // result is like "data:application/pdf;base64,AAAA..."
        const base64 = result.split(",")[1];
        resolve(base64);
      } catch (err) {
        console.log("wrong file uploaded")
        reject(err);
      }
    };
    reader.onerror = reject;
    reader.readAsDataURL(file);
  });
};

function Evaluator() {
  const [answerKeyFile, setAnswerKeyFile] = useState(null);
  const [studentFile, setStudentFile] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState("");
  const [result, setResult] = useState(null);

  const handleEvaluate = async () => {
    setError("");
    setResult(null);

    if (!answerKeyFile || !studentFile) {
      setError("Please upload both the answer key and the student answer sheet.");
      return;
    }

    if (!API_KEY) {
      setError("Please add your Gemini API key in the code (API_KEY constant).");
      return;
    }
```

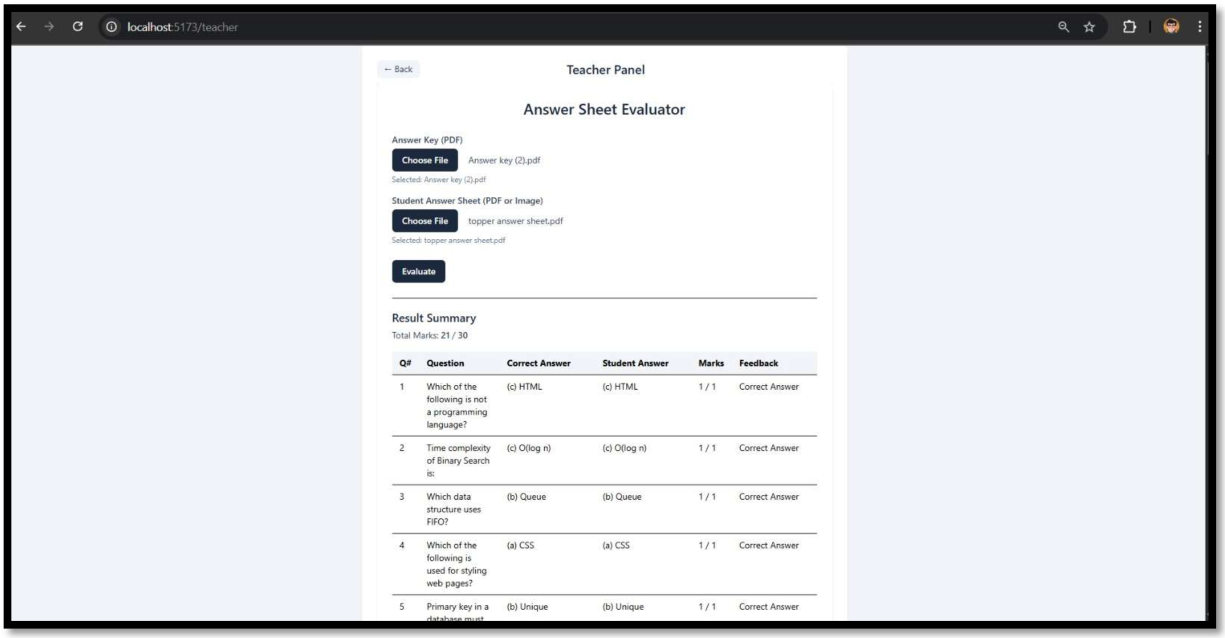


Welcome to the Assessment Portal

Choose your panel to continue.

Student Panel
View your evaluated results

Teacher Panel
Upload answer keys & evaluate sheets



Teacher Panel

Answer Sheet Evaluator

Answer Key (PDF)

Answer key (2).pdf

Selected: Answer key (2).pdf

Student Answer Sheet (PDF or Image)

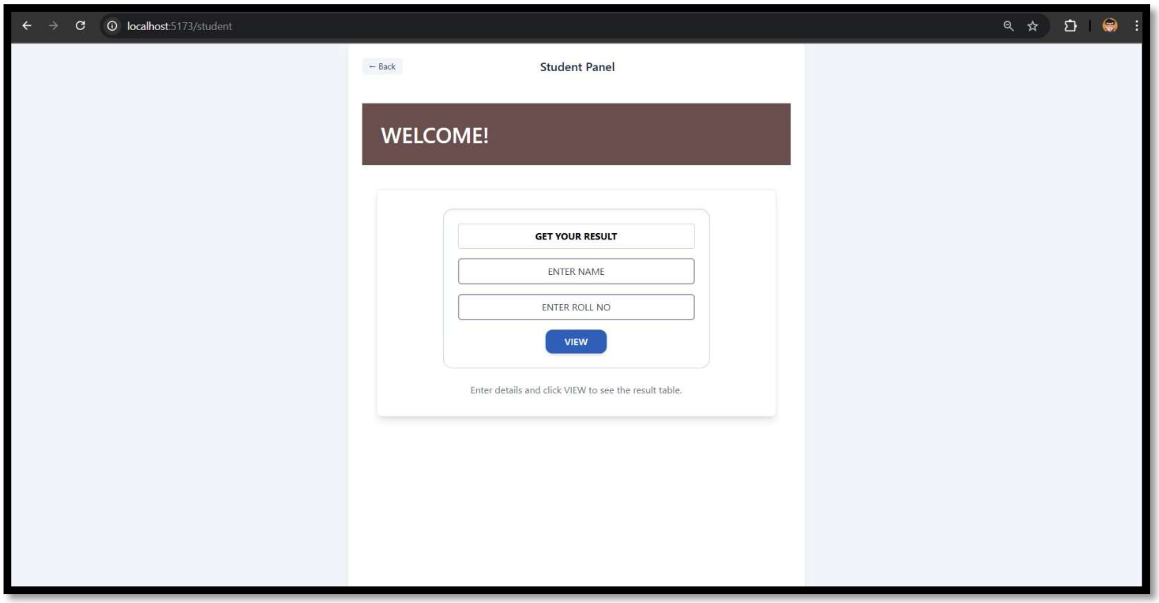
topper answer sheet.pdf

Selected: topper answer sheet.pdf

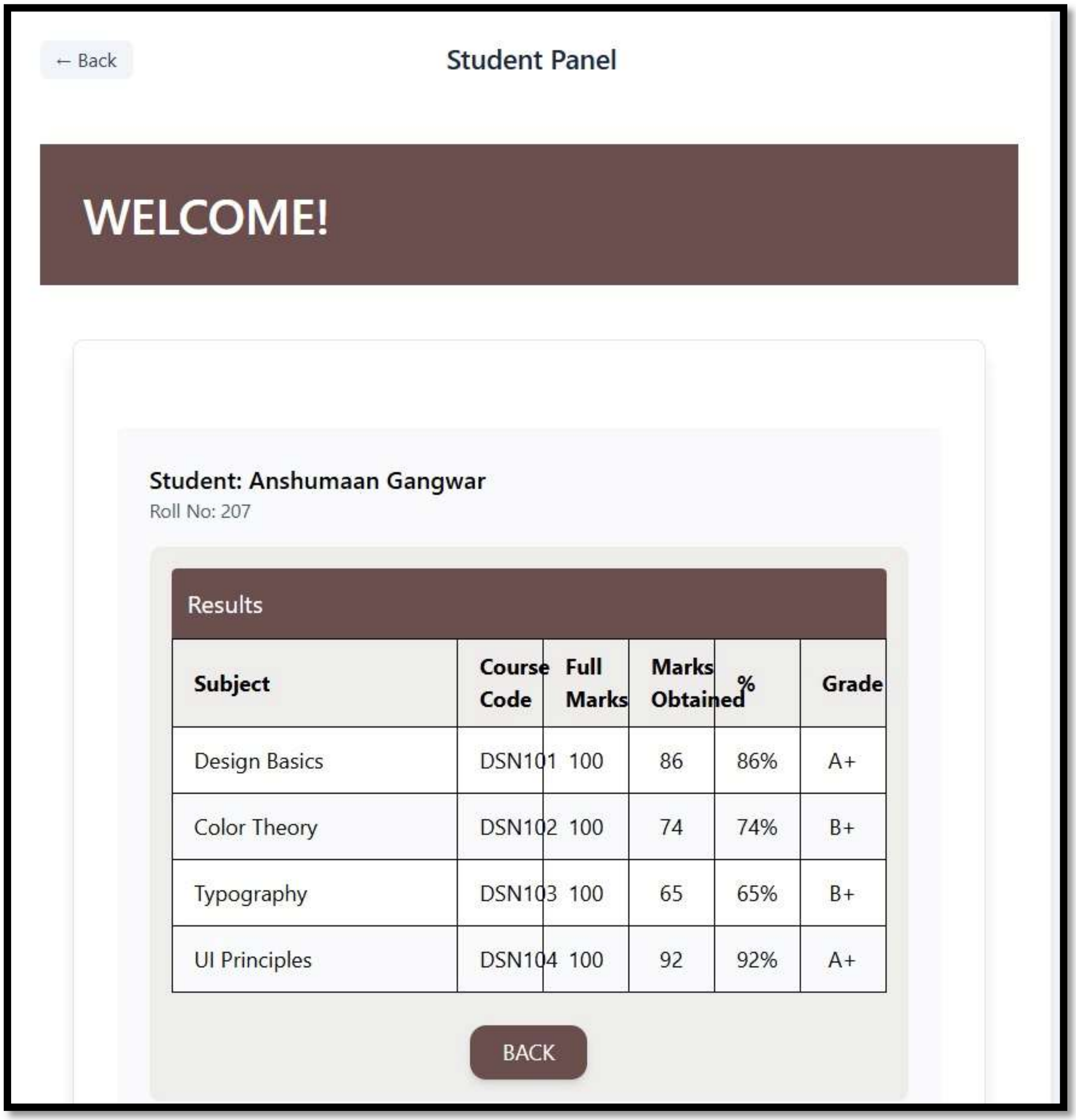
Result Summary

Total Marks: **21 / 30**

Q#	Question	Correct Answer	Student Answer	Marks	Feedback
1	Which of the following is not a programming language?	(c) HTML	(c) HTML	1 / 1	Correct Answer
2	Time complexity of Binary Search is:	(c) O(log n)	(c) O(log n)	1 / 1	Correct Answer
3	Which data structure uses FIFO?	(b) Queue	(b) Queue	1 / 1	Correct Answer
4	Which of the following is used for styling web pages?	(a) CSS	(a) CSS	1 / 1	Correct Answer
5	Primary key in a database must	(b) Unique	(b) Unique	1 / 1	Correct Answer



Enter details and click VIEW to see the result table.



Student: Anshumaan Gangwar

Roll No: 207

Results

Subject	Course Code	Full Marks	Marks Obtained	%	Grade
Design Basics	DSN101	100	86	86%	A+
Color Theory	DSN102	100	74	74%	B+
Typography	DSN103	100	65	65%	B+
UI Principles	DSN104	100	92	92%	A+