IntelliQuiz

Milestone 1 Report

CPSC 2350 - Software Practices

Instructor: Parsa Rajabi

Langara College

Authors:

Harpreet Singh

Miguel Fierro

Patrick Fang

Utsav Monga

Github

Table of Contents

Overview

IntelliQuiz is an AI-powered quiz maker web application, designed for students who wish to streamline study sessions and instructors who want to simplify quiz creation for their classes.

The app is able to take text from the user interface or PDF documents as input. The type and number of questions are set by the user, and can be downloaded as PDF or attempted from within the application. Quiz questions are presented one at a time during attempts, each will have a "Hint" option and an "Explanation" option in case the user does not understand what is being asked. Finally, in addition to the attempt score, feedback is provided at the end of the quiz pointing toward topics that need more review. This final summary can also be downloaded as a report PDF.

SDLC

The team decided to use the Kanban SDLC framework because of its flexibility and pull system, which helps manage workflow and reduces bottlenecks. The number of tasks in progress are limited to ensure that each task is given proper attention and doesn't overwhelm the team. Additionally, the Github Actions feature already incorporates a similar system, making it simple to implement for the team.

User Stories

OpenAI

1. As a student, I need feedback on my performance at the end of the quiz, highlighting the areas that need improvement and statistics of right and wrong answers.
2. As a user, I sometimes need hints for challenging questions or explanations for questions that are hard to understand.

ILovePDF

1. As a teacher, I need to download quizzes as pdf to help students assess their knowledge
2. As a teacher, I need to keep the pdfs password protected so that students can see the quiz only when I release the password.

Technology Stack

Front-End	APIs	Testing	Deployment
React	OpenAI	Jest	Render
CSS/Bootstrap	ILovePDF	React Testing Library	

Chosen APIs

The Quiz Generator application will make use of the OpenAI and ILovePDF APIs, the features chosen for each API are as follows:

OpenAI

1. Take text input and generate a list of questions based on it. The list will be returned as a JSON object with the following format:

```
{
"questions": [
{
"question": "Who was the primary author of the Declaration of Independence?",
"options": ["George Washington", "Thomas Jefferson", "Benjamin Franklin", "John Adams"],
"answer": "Thomas Jefferson"
},
]
}
```

2. Explain a question and give hints
   If the user does not understand what is being asked, the LLM can provide guidance without spoiling the answer, similarly it can provide hints to the user.

3. Quiz grade feedback
   At the end of each quiz, the LLM will generate feedback to accompany the grade of the quiz, to provide insights and further study material to the user.

ILovePDF

1. Extract text from a PDF file.
   The app will allow the user to submit raw text as input, but also submit a PDF file from which the API will extract the text.

2. Generate a PDF from Quiz results.
   The user can download a PDF file with the questions and answers before attempting the quiz and can download their attempt summary/feedback as a PDF report as well.

3. Lock PDF behind a password.
   In case the PDF document is to be shared digitally (i.e. by a teacher to their students), the quiz PDF can be locked behind a password before downloading.

Work Breakdown Structure

Task Number	Task	Assigned To	Estimated Hours	Actual Hours
1	Requirement Gathering			
1.1	Technical Specification			
1.1.1	Choose the programming language	Harpreet	1	1
1.1.2	Choose the APIs	Miguel	2	3
1.1.3	Define project scope	Team	2	1
1.1.4	Set up GitHub repository	Patrick	0.5	0.5
1.1.5	Determine test cases	Utsav	1	1.5
1.2	Team requirements			
1.2.1	Assign team roles	Harpreet	1	0.5

1.2.2	Define team communication channels	Harpreet	0.5	0.5
1.3	Document requirements			
1.3.1	Determine Git flow	Team	1	0.5
1.3.2	Define team communication schedule	Harpreet	0.5	0.5
1.4	Milestone 1 report			
1.4.1	Overview	Utsav	1	1
1.4.2	SDLC	Miguel	1.5	1
1.4.3	User Stories	Utsav	1	0.5
1.4.4	Technology Stack	Harpreet	1	0.5
1.4.5	Chosen APIs	Team	1	1
1.4.6	WBS	Team	2	2
1.4.7	Project Schedule	Miguel & Harpreet	0.5	0.5
1.4.8	Wireframes	Patrick	1.5	2
1.4.9	Data Flow Diagram	Miguel	1.5	2
1.4.10	Polish/Edit	Patrick	2	1.5
1.5	Video presentation			
1.5.1	Editing	Miguel & Utsav	2	3
1.5.2	Upload video	Miguel	1	1
1.6	**Complete Requirement Gathering**		**25**	**25**

2	Front-End			
2.1	Layout Design			

2.1.1	Design wireframes	Patrick	2	
2.1.2	Setup React/CSS/Bootstrap design	Utsav & Patrick	4	
2.2	User Interface			
2.2.1	Input Interface (Textbox/File upload)	Harpreet	2	
2.2.2	Number of questions, question types	Utsav & Miguel	1.5	
2.2.3	Navigation buttons	Miguel	1	
2.2.4	Quiz cards	Harpreet	1	
2.3	API integration			
2.3.1	Parse JSON to display questions	Miguel	0.5	
2.4	**Complete Front-End**		**13**	

3	API Calls			
3.1	Get questions from OpenAI			
3.1.1	Get OpenAI credentials	Utsav	0.5	
3.1.2	Send text to API	Patrick	1	
3.1.3	Retrieve questions as JSON	Team	1	
3.2	Get hint/explanation from OpenAI			
3.2.1	Send question to API	Miguel	1	
3.2.2	Get hint for question	Team	1	
3.2.3	Get explanation for question	Team	1	
3.3	Extract text from PDF file			
3.3.1	Get ILovePDF API keys	Miguel	0.5	

3.3.2	Send PDF file to API	Patrick	2	
3.3.3	Retrieve text from API	Team	1	
3.4	Generate PDF file from quiz attempt			
3.4.1	Compile quiz attempt information	Harpreet	3	
3.4.2	Send attempt info to API	Harpreet	1	
3.4.3	Lock PDF file with password	Team	2	
3.5	**Complete API calls**		**15**	

4	Deployment			
4.1	Deployment platform			
4.1.1	Choose deployment platform	Team	1	
4.1.2	Deploy GitHub repository	Team	3	
4.2	**Complete Deployment**		**4**	

5	Documentation			
5.1	SDLC Lab			
5.1.1	Brainstorm Ideas and Answers	Team	2	
5.1.2	Create Document and Report	Team	2	
5.2	Project Plan			
5.2.1	Project Description	Miguel & Utsav	0.5	
5.2.2	WBS	Team	3	
5.2.3	Project Limitation	Team	1	

5.2.4	Sample output	Miguel	1	
5.2.5	Edit README file	Patrick & Utsav	0.5	
5.2.6	Code Documentation	Miguel & Harpreet	2	
5.2.7	Polish/Edit	Team	2	
5.3	Project Report			
5.3.1	Project Description	Team	0.5	
5.3.2	WBS	Team	2	
5.3.3	Project Limitations	Team	1	
5.3.4	Sample output	Patrick & Harpreet	1	
5.3.5	Presentation Preparation	Team	2	
5.3.6	Code Documentation	Harpreet & Miguel	3	
5.3.7	Meeting Documentation	Utsav & Patrick	1	
5.4	**Complete Documentation**		**24.5**	

Project Schedule

1. **Requirement Gathering: Feb 4- Feb 19**

   1.1 Technical Specification: Feb 4- Feb 10

   1.2 Team requirement: Feb 11

   1.3 Document requirement: Feb 12

   1.4 Milestone 1 report: Feb 13- Feb 19

   1.5 Video Presentation: 19 Feb

   1 day for any risk mitigation any failure in Requirement gathering or time mismanagement

2. **Front End: 21 Feb- 2 March**

2.1 Layout Design: 21 Feb - 22 Feb

2.2 User Interface Implementation: 23 Feb - 26 Feb

2.3 API Integration: 27 Feb - 29 Feb

2 day for risk mitigation in any front end failure or time mismanagement

3. **API Calls: 3 March - 7 March**

   3.1 API Calls: 3 March - 6 March

   1 day for risk mitigation in any API Calls failure or time mismanagement

4. **Deployment: 8 March**

   4.1 Deploy Github repo: 8 March

5. **Documentation: 9 March - 14 March**

   5.1 SDLC lab: 9 March - 10 March

   5.2 Project Plan: 11 March - 12 March

   5.3 Project Report : 12 March - 13 March

   1 Day for risk mitigation in documenting report or time mismanagement

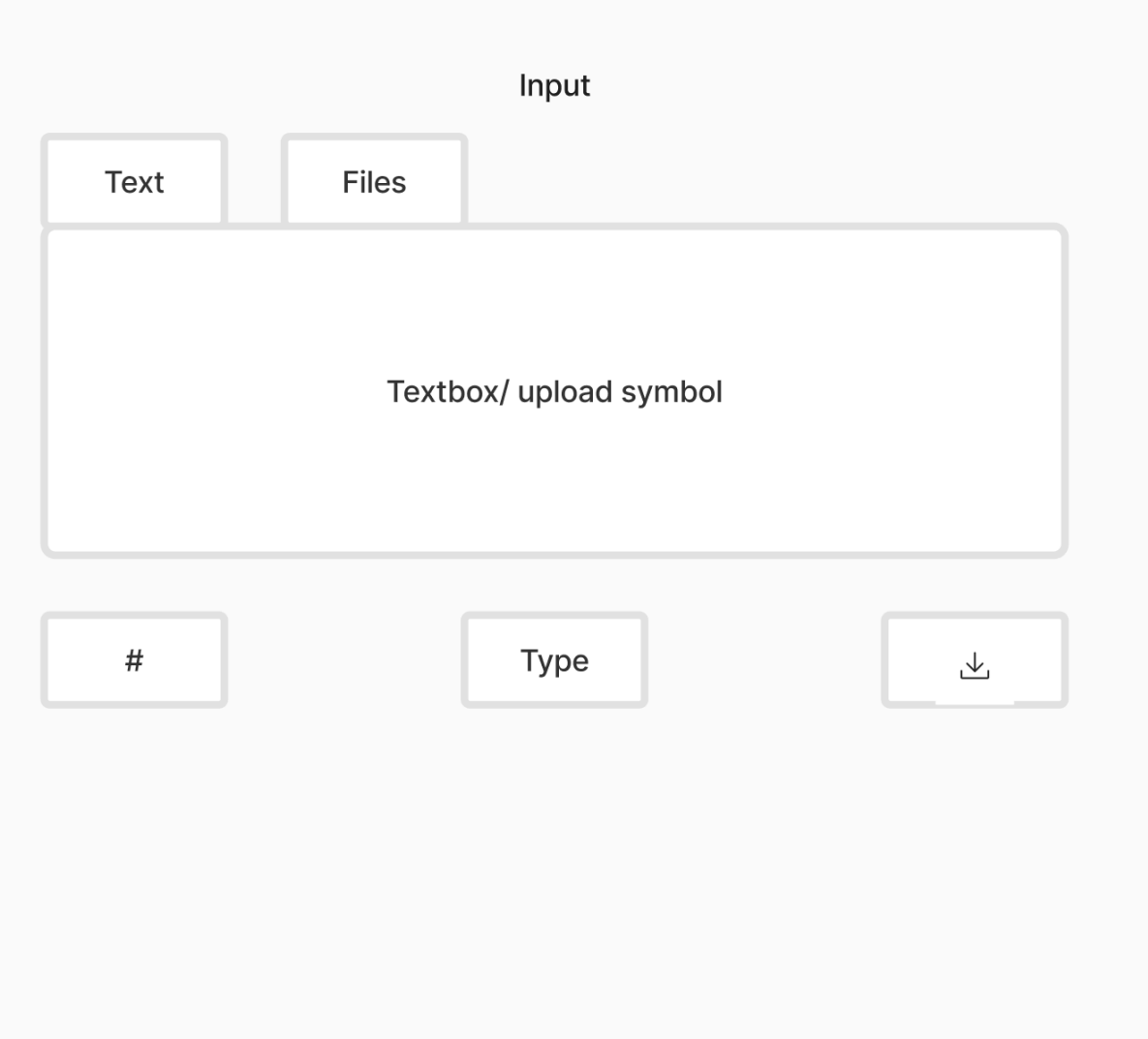
6. **Final Submission - 16 March**

   2 days in-between for any risk mitigation

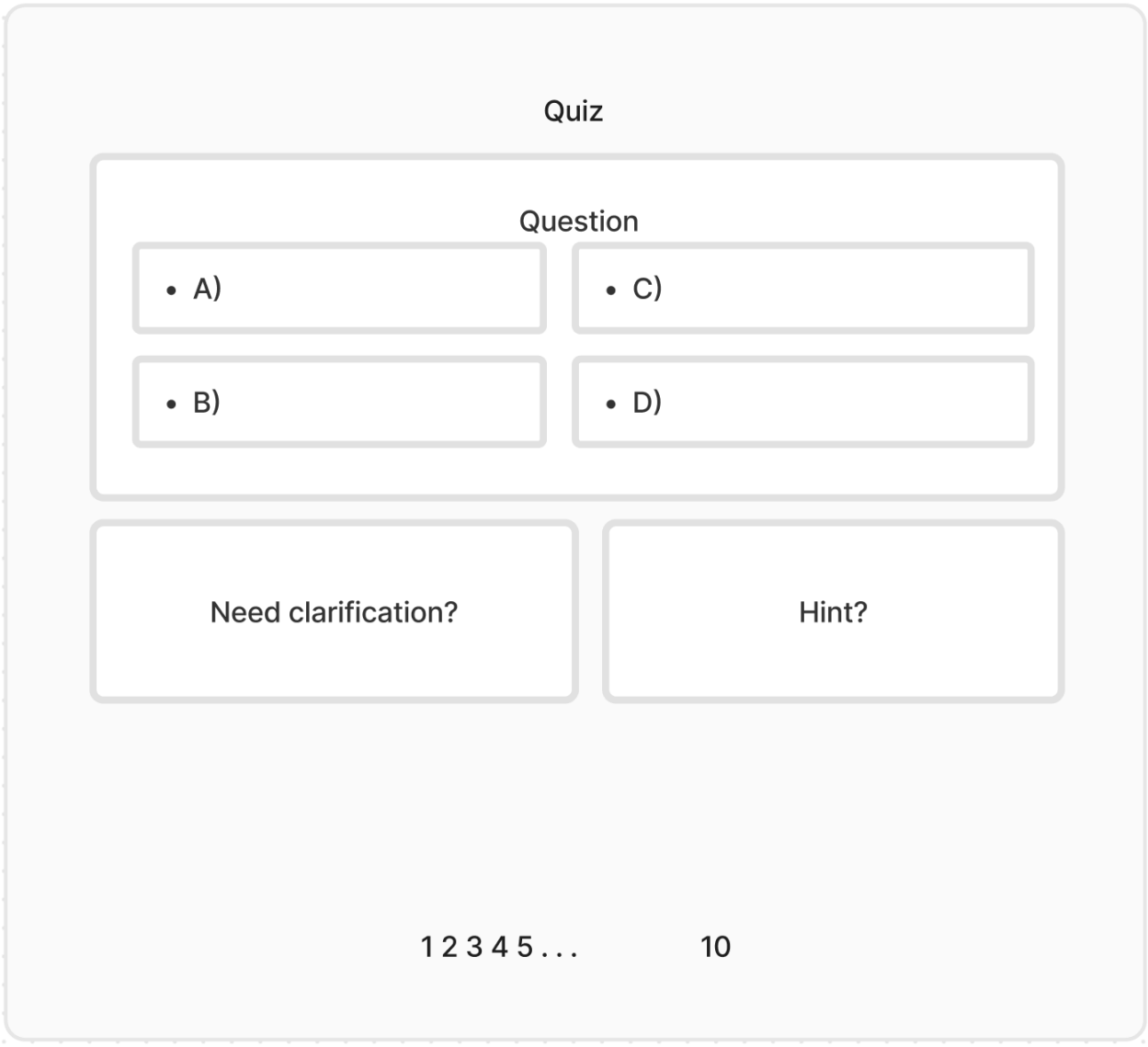
Meeting Schedule

- Weekly Team Meetings every Tuesday or Wednesday 11:00 am to 4:00 pm
- Bi-weekly Online Update Sessions every Monday on Discord 11:00 am to 1:00 pm

Wireframes



IntelliQuiz



IntelliQuiz

Score

79/100

Question 1

- A)
- C)
- B)
- D)

Short Explanation

Question 2

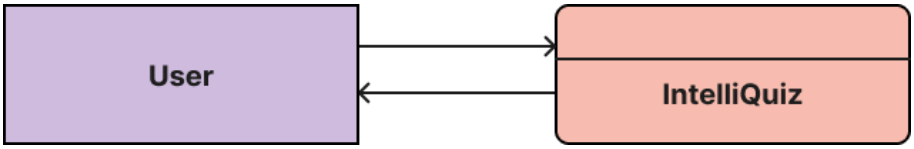
- A)
- B)
- C)
- D)

Short Explanation

Feedback

Data Flow Diagram

Level 0



Level 1