Test Plan
for
Learn AI

Version – 1.0

Sarthak Awasthi
Monish Barot
Raj Ray Chaudhury
Elizabeth Kearns
Hasan Muhammed

Drexel University
April 18, 2024

Table of Contents

1. Introduction

This document outlines the testing plans, approaches, risks, and assumptions of the AI Learn application. The information defined here has been written before any proper

testing has been conducted and will be used as a guide when the testing process has officially begun.

1.1. Definitions

- **AI Learn Application:** The software system under test, designed to provide instructors with a framework for creating interactive learning activities with AI-powered coaching for students.
- **Instructor:** The user who designs and configures the learning activity using the AI Learn application.
- **Student:** The user who interacts with the learning activity and receives AI coaching.
- **AI Coaching Prompt:** Instructions provided by the instructor to guide the AI's feedback for the student.
- **Student Response:** The text submitted by the student for review by the AI.
- **AI Response:** The feedback generated by the AI based on the student's response and the instructor's coaching prompt.

1.2. In Scope

- API
    - Returns data we expect from endpoints.
- Login Page
    - Students can access class with code
    - Instructors log in with username and password
- Student Page
    - Receiving assignments from instructor
    - Submitting assignments to AI and instructor
    - Receive AI response
    - Can submit to AI multiple times
    - Receive Instructor response
- Instructor Page

- Create assignments
- Create different types of assignments
- Fine-tune AI for assignment
- Review received assignments
- Give feedback to students

1.3. Out of Scope

- Underlying algorithms of the AI model
- Performance issues not directly related to the AI Learn (e.g., network connectivity)
- Security testing

2. Testing Approach

This section will describe the approach we will take to test each part of our application, as well as how these tests will be conducted.

2.1. User Interface Testing Strategy

- **Functional Testing:**

Functional Testing focuses on verifying the correctness of all user interface functionalities according to design specifications within the application. We plan to use Cypress Testing Library to automate tests that will include simulating user interactions and ensuring that all the components operate according to our expectations. Key areas include:

- **Component Functionality**:

Ensuring each component behaves as expected when triggered (e.g., buttons trigger appropriate actions, data entry fields accept and validate input correctly).

- **Form Submission**:

Testing all forms for correct submission behavior, including data validation and submission handling.

- **Navigation**:

Verifying that navigation elements like menus and links lead to the correct pages or sections without errors.

- **Error Handling**:

Ensuring that user errors are handled gracefully, with clear, helpful error messages displayed.

- **Integration Testing:**

Integration Testing assesses the cooperation between the front-end and the backend to ensure they work together as intended. For our React application using Cypress, integration tests will focus on:

- **API Integration:**

Verifying that front-end components correctly interact with the backend through API calls, handling responses as expected, and updating the UI accordingly.

- **Session Management:**

Testing session persistence across various user interactions and page refreshes to ensure user data and authentication states are maintained.

2.2. API Testing Strategy

For the backend we are using Flask to create the server which handles all our API endpoints. We are using PyMongo Library to interact with the MongoDB Database. We're using OpenAI API to get AI responses. Both the MongoDB and OpenAI APIs are wrapped within our Flask API Endpoints.

- **Functional Testing:**

Functional Testing of the backend focuses on verifying each function within the API and other backend services perform as expected under various conditions. This includes:

- **API Endpoint Functionality:**

Testing each API endpoint to ensure it accepts correct inputs and returns the expected outputs. This includes endpoints for course management, user authentication, learning activities, and student submissions. We plan to use 'unittest', python's inbuilt library for unit testing.

- **Database Operations:**

Verifying that all database operations (create, read, update, delete) are executed correctly, maintaining data integrity and consistency. We plan to use MongMock, a python library for simulating MongoDB actions.

- **Error Handling:**

Testing the backend's ability to handle errors gracefully, ensuring that meaningful error messages are returned to the client. We plan to use ‘unittest’, python’s inbuilt library for unit testing.

- **Security Tests:**

Checking for vulnerabilities in access control and data validation. We plan to use MongoAudit, a python cli tool to validate our DB operations.

- **Integration Testing:**

Integration Testing examines the interaction between the backend services and other components of the application to ensure they work together as intended. This section does not dive into Frontend Integration as we’ve examined it in “Integration Testing” under UI Testing. Instead, this section focuses on:

- **External Services Integration**:

Verifying that the backend interacts properly with external services, like the OpenAI GPT API for the AI feedback system. This includes testing the handling of API requests and responses, authentication mechanisms, and data formatting. We’re planning to conduct this using Postman.

- **Performance and Load Testing**:

Assessing the backends performance under load, particularly focusing on API response times and system behavior under high traffic conditions.

2.3. System Testing Strategy

Our System testing would include testing our overall application from the user flow perspective. We plan to utilize end-to-end testing which would include:

- **Course Creation and Management:**

Ensuring instructors can create, edit, and manage courses without issues, verifying that forms submit properly and handle data correctly.

- **Learning Activity Operations:**

Testing the functionality to add, edit, and delete learning activities within a course, ensuring that these changes are reflected in real-time.

- **Student Submissions Viewing:**

Verifying that instructors can view student submissions effectively, with proper loading and display of submission data.

- **Feedback Iteration Process:**

Testing the iterative feedback system where students receive AI-generated responses, ensuring the loop works smoothly and updates as expected after each iteration.

- **Accuracy and Relevance of AI Suggestions:**

Assess the AI's ability to provide accurate and relevant suggestions about the course material and learning activity description.

2.4. Tools to Be Used

- **MongoMock:**

MongoMock will be used to simulate MongoDB operations, allowing us to test database interactions for features like course management and student submissions without altering actual data.

- **MongoAudit:**

MongoAudit will ensure our MongoDB configurations are secure and comply with best practices, crucial for protecting course and student data.

- **unittest (in Python):**

We will use Python's unittest framework to conduct unit tests on backend logic, particularly the functions managing AI feedback and data operations.

- **Cypress:**

Cypress will facilitate end-to-end testing of real user interactions within our application, from managing courses to engaging with the AI feedback system.

- **Postman:**

Postman will test API endpoints related to frontend and AI communication, ensuring they are reliable and handle data correctly.

2.5. Testing Schedule and Responsibilities

Tasks	**Team Members**	**Timeline**
Testing React Components in Cypress	**Sarthak, Hasan**	**Week 5**
API endpoint Testing	**Monish, Raj**	**Week 5**
Integration Testing (API Calls from Client side)	**Elizabeth**	**Week 5**
Database Operations Testing	**Monish**	**Week 6**
Integration Testing (Backend side: PyMongo and OpenAI API)	**Raj, Elizabeth**	**Week 6**
Error Handling Testing	**Sarthak**	**Week 6**
Security Testing	**Hasan**	**Week 6**
System Testing (End-to-End)	**Everyone**	**Week 7**

3. Test Cases

The following test cases below are how we will plan to conduct our testing based off of the approaches labeled above. These were all created before any of the testing actually occurred.

3.1. Backend Website

Number: 1	Title: Instructor Login	Priority: High
ID: API_INSTRUCTOR_LOGIN		
Description: Verify that the secure login mechanism for instructors is functioning correctly.		
Pre-condition: Valid instructor credentials are available.		
Action(s)/Steps: Enter valid instructor username and password.		

Submit login request. Verify that a valid JSON Web Token (JWT) is returned.
Expected Result: Instructor is successfully logged in and granted access to the administrator portal.
Created by: Raj Ray Chaudhury
Test Status: [Pass/Fail] Actual Results: Instructor is successfully logged in and granted access to the administrator portal.

Number: 2	Title: Student Login	Priority: High
ID: STUDENT_LOGIN		
Description: Validate that students can log in to the student portal.		
Pre-condition: Valid student credentials are available.		
Action(s)/Steps: Enter response in the provided text box. Submit the response. Verify successful submission.		
Expected Results: Student gains access to the student portal.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Student gains access to the student portal.		

Number: 3	Title: Activity Creation	Priority: High
ID: API_ACTIVITY_CREATION		
Description: Ensure that instructors can create new learning activities with accurate information.		
Pre-condition: Required activity creation details are available.		
Action(s)/Steps: Provide activity identifier, learning material, instructions, coaching prompt, next steps, and maximum text input length. Submit activity creation request. Verify that the activity is created successfully.		
Expected Results: New learning activity is added to the system with all specified details.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: New learning activity is added to the system with all specified details.		

Number: 4	Title: API Endpoint Functionality for Course Management	Priority: High
ID: API_ENDPOINT_COURSE_MANAGEMENT		
Description: Verify that the backend API endpoints for creating, editing, and managing courses function as expected.		
Pre-condition: The API endpoints are accessible and properly configured.		
Action(s)/Steps: Send API requests to create, edit, and retrieve course information. Verify that the API responses contain accurate and complete course data. Verify that data manipulation (creation, editing, deletion) is reflected in the database.		
Expected Results: The API endpoints successfully manage course data and interact with the database accurately.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The API endpoints successfully manage course data and interact with the database accurately.		

Number: 5	Title: User Interface Functionality	Priority: High
ID: USER_INTERFACE_FUNCTIONALITY		
Description: Verify that the user interface elements (buttons, forms, text fields) render correctly and function as intended.		
Pre-condition: The application is launched in a supported web browser.		
Action(s)/Steps: Navigate through different application pages (login, course management, learning activities). Interact with various UI elements (buttons, forms, text fields) and verify their functionality. Verify that data is displayed accurately, and user interactions trigger the expected actions.		
Expected Results: The user interface elements render correctly, function as intended, and provide a smooth user experience.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The user interface elements render correctly, function as intended, and provide a smooth user experience.		

Number: 6	Title: Accessibility and Responsible Design	Priority: Medium
ID: ACCESSABLE_DESIGN		
Description: Verify that the application is accessible to users with disabilities and adapts to different screen sizes and devices.		
Pre-condition: The application is launched in various web browsers and devices.		

Action(s)/Steps: Access the application using screen reader software. Verify that essential elements are accessible and navigable with assistive technologies. Resize the browser window and observe the layout responsiveness. Access the application on different devices (mobile, tablet). Verify that the user interface elements adjust appropriately for different screen sizes.
Expected Results: The application is accessible to users with disabilities using assistive technologies. The user interface layout adapts seamlessly to different screen sizes and devices. All functionalities are accessible and usable regardless of device or screen size.
Created by: Monish Barot
Test Status: [Pass/Fail] Actual Results: Application is accessible and fulfils the expected conditions

Number: 7	Title: Cross-Browser Compatibility	Priority: Medium
ID: CROSS_BROWSER_COMPATIBILITY		
Description: Verify that the application functions correctly and consistently across different web browsers.		
Pre-condition: Access the application in various popular web browsers (e.g., Chrome, Firefox, Safari, Edge).		
Action(s)/Steps: Navigate through different application pages and functionalities in each browser. Verify that the user interface elements render correctly, interactions function as expected, and data is displayed accurately.		
Expected Results: The application maintains consistent functionality and visual appearance across different supported web browsers.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The application is consistent across different browsers		

Number: 9	Title: Multiple Choice Question Functionality	Priority: High
ID: MCQ_FUNCTIONALITY		
Description: Verify the functionality of multiple-choice questions, including answer selection, scoring, and feedback generation.		
Pre-condition: Define a learning activity with multiple-choice questions and answer options.		
Action(s)/Steps: Access the learning activity with multiple-choice questions. Select different answer options and verify that the selection is reflected correctly. Submit the response and verify the scoring mechanism assigns points accurately based on correct answers.		

Analyze the AI-generated feedback to ensure it addresses the chosen answer and provides relevant explanations.
Expected Results: Multiple-choice questions function as expected, answer selection works accurately, scoring is correct, and AI feedback is relevant and informative.
Created by: Monish Barot
Test Status: [Pass/Fail] Actual Results: Multiple-choice questions function as expected, answer selection works accurately, scoring is correct, and AI feedback is relevant and informative.

Number: 10	Title: Essay Question Functionality	Priority: High
ID: ESSAY_FUNCTIONALITY		
Description: Verify the functionality of essay questions, including response submission, plagiarism detection, and AI feedback generation.		
Pre-condition: Define a learning activity with an essay question.		
Action(s)/Steps: Access the learning activity with the essay question. Submit a student response of varying lengths and formats. Verify the system detects and flags potential plagiarism attempts. Analyze the AI-generated feedback to ensure it evaluates the content, identifies strengths/weaknesses, and provides suggestions for improvement.		
Expected Results: Essay question submission works, plagiarism detection functions effectively, and AI feedback provides comprehensive analysis and suggestions.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: Essay question submission works, plagiarism detection functions effectively, and AI feedback provides comprehensive analysis and suggestions.		

Number: 11	Title: User Input Validation for Text Fields	Priority: Medium
ID: TEXT_FEILD_VALIDATION		
Description: Verify that the application enforces data validation rules for various text fields, including preventing empty submissions, exceeding character limits, and accepting specific formats.		
Pre-condition: Define data validation rules for different text fields (e.g., username, email, answer fields).		
Action(s)/Steps: Attempt to submit forms with empty text fields, exceeding character limits, or invalid formats. Verify that the application displays clear error messages indicating invalid input. Test that the system prevents submission until valid data is entered.		
Expected Results: Data validation rules are enforced, preventing invalid submissions and prompting users to correct their input.		

Created by: Monish Barot	
Test Status: [Pass/Fail] Actual Results: Data validation rules are enforced, preventing invalid submissions and prompting users to correct their input.	

Number: 12	Title: File Upload Validation	Priority: Medium
ID: FILE_UPLOAD_VALIDATION		
Description: Verify the functionality of file upload features, including supported file formats, size limitations, and virus scanning.		
Pre-condition: Define acceptable file formats and size limits for file uploads.		
Action(s)/Steps: Attempt to upload files exceeding size limits or in unsupported formats. Verify that the application displays error messages indicating invalid file selection. Test that the system scans uploaded files for potential viruses and prevents upload if detected.		
Expected Results: File upload functionality adheres to defined limitations, displays error messages for invalid files, and protects against virus threats.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: File upload functionality adheres to defined limitations, displays error messages for invalid files, and protects against virus threats.		

3.2. Instructor Front-End

Number: 13	Title: Activity Feedback	Priority: High
ID: INSTRUCTOR_FEEDBACK		
Description: Verify that instructors receive accurate feedback from the AI on student responses.		
Pre-condition: Student responses to activities are available for evaluation.		
Action(s)/Steps: Review AI-generated feedback for student responses. Compare AI feedback with expected feedback based on response content. Verify the accuracy and relevance of AI feedback.		
Expected Results: Instructors receive actionable feedback to improve learning activities and student engagement.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Instructors receive actionable feedback to improve learning activities and student engagement.		

Number: 14	Title: Edit AI Feedback Settings	Priority: High
ID: INSTRUCTOR_FEEDBACK_EDIT		
Description: Ensure that instructors can modify AI feedback settings for specific activities.		
Pre-condition: Activity feedback settings are accessible for editing.		
Action(s)/Steps: Navigate to AI feedback settings for an activity. Modify AI feedback parameters such as tone, depth, and suggestions. Save changes to AI feedback settings.		
Expected Results: AI feedback settings are updated and applied to student responses.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: AI feedback settings are updated and applied to student responses.		

Number: 15	Title: Edit Learning Activity Details	Priority: High
ID: INSTRUCTOR_ACTIVITY_EDIT		
Description: Verify that instructors can edit details of existing learning activities.		
Pre-condition: Existing learning activities are available for editing.		
Action(s)/Steps: Select a learning activity to edit. Modify activity details such as instructions, coaching prompt, or next steps. Save the changes to the activity.		
Expected Results: Updated learning activity reflects the edited details accurately.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Updated learning activity reflects the edited details accurately.		

Number: 16	Title: Delete Learning Activity	Priority: High
ID: INSTRUCTOR_ACTIVITY_DELETE		
Description: Ensure that instructors can delete learning activities from the system.		
Pre-condition: Learning activities to be deleted are identified.		
Action(s)/Steps: Select the learning activity to delete. Confirm the deletion action. Verify that the activity is removed from the system.		
Expected Results: Deleted learning activity is no longer accessible in the system.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail]		

Actual Results: Deleted learning activity is no longer accessible in the system.

Number: 17	Title: Edit AI Feedback Settings Again	Priority: High
ID: INSTRUCTOR_FEEDBACK_EDIT_AGAIN		
Description: Verify that instructors can modify AI feedback settings multiple times for the same activity.		
Pre-condition: Activity feedback settings are accessible for editing and have been edited previously.		
Action(s)/Steps: Modify AI feedback parameters (e.g., tone, depth) for an existing activity. Save the changes to AI feedback settings. Repeat the above steps to make additional changes to AI feedback settings.		
Expected Results: AI feedback settings reflect the latest modifications and updates.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: AI feedback settings reflect the latest modifications and updates.		

Number: 18	Title: Preview Learning Activity	Priority: High
ID: INSTRUCTOR_ACTIVITY_PREVIEW		
Description: Ensure that instructors can preview the content and format of a learning activity before publishing.		
Pre-condition: Learning activity is in draft mode and ready for preview.		
Action(s)/Steps: Access the preview mode for the learning activity. Review the content, formatting, and overall appearance of the activity. Verify that the preview accurately represents how the activity will appear to students.		
Expected Results: Instructors have a clear preview of the learning activity for final review before publishing.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Instructors have a clear preview of the learning activity for final review before publishing.		

Number: 19	Title: Sidebar Courses Component	Priority: High
ID: INSTRUCTOR_SIDEBAR_RENDER		
Description: Ensure that instructors dashboard page’s sidebar loads correctly.		
Pre-condition: Instructor is logged in and on homepage.		

Action(s)/Steps: User logs in.
Expected Results: Instructor dashboard renders the sidebar properly.
Created by: Sarthak Awasthi
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar properly.

Number: 20	Title: Sidebar Courses List	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_RENDER		
Description: Ensure that instructors courses render on sidebar load correctly.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in.		
Expected Results: Instructor dashboard renders the sidebar properly.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar properly.		

Number: 21	Title: Sidebar Courses API Call	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_FETCH		
Description: Ensure that client-side API call for courses list work properly.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in.		
Expected Results: Instructor dashboard renders the sidebar courses properly.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar courses properly.		

Number: 22	Title: Sidebar Courses Information	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_INFO		
Description: Ensure that each course displayed has correct information.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in.		
Expected Results: Each course is displayed with correct information.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail]		

Actual Results: Each course is displayed with correct information.

Number: 23	Title: Sidebar Courses Status	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_STATUS		
Description: Ensure that each course rendered has correct status (Open/Locked).		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in.		
Expected Results: Instructor courses are displayed with accurate statuses.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor courses are displayed with accurate statuses.		

Number: 24	Title: Sidebar Courses Click	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_CLICK		
Description: Ensure that courses elements are clickable/pressable.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in. Attempts to click a course.		
Expected Results: Instructor courses are clickable.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor courses are clickable.		

Number: 25	Title: Sidebar Courses Click Redirect	Priority: High
ID: INSTRUCTOR_SIDEBAR_COURSES_REDIRECT		
Description: Ensure that clicking on the course redirects instructor to course page.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in. Attempts to click a course.		
Expected Results: Instructor is redirected to specific course page.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor is redirected to specific course page.		

Number: 26	Title: Sidebar Null Courses	Priority: High
ID: INSTRUCTOR_SIDEBAR_HAS_NONE_COURSES		

Description: Ensure that instructors dashboard sidebar functions smoothly with no courses.
Pre-condition: Instructor is logged in and on homepage.
Action(s)/Steps: User logs in.
Expected Results: Instructor dashboard renders the sidebar properly.
Created by: Sarthak Awasthi
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar properly.

Number: 27	Title: Sidebar Courses Component After Course Added	Priority: High
ID: INSTRUCTOR_SIDEBAR_RENDER_POST_COURSE_ADDITION		
Description: Ensure that instructors dashboard page's sidebar loads old and new courses correctly.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in. Adds a new course. Navigates back to homepage.		
Expected Results: Instructor dashboard renders the sidebar properly with updated list of courses.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar properly with updated list of courses.		

Number: 28	Title: Sidebar Courses Component on Page Resize	Priority: Low
ID: INSTRUCTOR_SIDEBAR_RENDER_ON_PAGE_RESIZE		
Description: Ensure that instructors dashboard page's sidebar loads correctly after page is resized.		
Pre-condition: Instructor is logged in and on homepage.		
Action(s)/Steps: User logs in. Resizes the homepage.		
Expected Results: Instructor dashboard renders the sidebar properly after page resized.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor dashboard renders the sidebar properly after page resized.		

Number: 29	Title: Course Name at Course Page	Priority: Medium
ID: COURSE_NAME		
Description: Ensure that correct course name is displayed on that course's page.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Redirects to course page.		
Expected Results: Instructor course page displays correct course name.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor course page displays correct course name.		

Number: 30	Title: Course Activities at Course Page	Priority: High
ID: COURSE_ACTIVITIES		
Description: Ensure that correct course activities are displayed on that course's page.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Redirects to course page.		
Expected Results: Instructor course page displays correct course activities.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor course page displays correct course activities.		

Number: 31	Title: Course Activities at Course Page are clickable	Priority: High
ID: COURSE_ACTIVITIES_CLICKABLE		
Description: Ensure that correct course activities are clickable.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Attempts to click an activity.		
Expected Results: Instructor can click an activity and get redirected.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail]		

Actual Results: Instructor can click an activity and get redirected.

Number: 32	Title: Course Activities redirect to activity page	Priority: High
ID: COURSE_ACTIVITIES_REDIRECT		
Description: Ensure that correct course activities are redirecting to correct activity page.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Attempts to click an activity.		
Expected Results: Instructor can click an activity and get redirected to correct activity page.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor is logged in and on a course page.		

Number: 33	Title: Create Learning Activity at Course Page	Priority: High
ID: CREATE_LEARNING_ACTIVITY		
Description: Ensure that create learning activity section is rendered on the course page.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course.		
Expected Results: Instructor can see the create learning activity section on course page.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor can see the create learning activity section on course page.		

Number: 34	Title: Prompt text box at Courses Page	Priority: High
ID: CREATE_LEARNING_PROMPT_TEXT_BOX		
Description: Ensure that prompt text box is rendered and functional on course page		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Views Prompt text box. Types in the text box.		
Expected Results: Instructor can view and use Prompt text box.		

Created by: Sarthak Awasthi	
Test Status: [Pass/Fail] Actual Results: Instructor can view and use Prompt text box.	

Number: 35	Title: Ideal answer text box at Courses Page	Priority: High
ID: CREATE_LEARNING_IDEAL_ANSWER_TEXT_BOX		
Description: Ensure that ideal answer text box is rendered and functional on course page		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Views ideal answer text box. Types in the text box.		
Expected Results: Instructor can view and use Ideal answer text box.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Instructor can view and use Ideal answer text box.		

Number: 36	Title: Study material section at Course Page	Priority: High
ID: STUDY_MATERIAL_SECTION		
Description: Ensure that study material section is displayed on course page.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course.		
Expected Results: Study material section is displayed properly.		
Created by: Sarthak Awasthi		
Test Status: [Pass/Fail] Actual Results: Study material section is displayed properly.		

Number: 37	Title: File Upload button at Course Page	Priority: High
ID: FILE_UPLOAD_BUTTON		
Description: Ensure that file upload button is rendered properly and functions properly.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course.		

Clicks on file upload button
Expected Results: File upload button renders and works properly.
Created by: Hasan Muhammed
Test Status: [Pass/Fail] Actual Results: File upload button renders and works properly.

Number: 38	Title: Multiple file Uploads at Course Page	Priority: High
ID: FILE_UPLOADS		
Description: Ensure that file upload button can upload multiple files.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Clicks on file upload button. Uploads multiple files.		
Expected Results: File upload button can should upload multiple files properly.		
Created by: Hasan Muhammed		
Test Status: [Pass/Fail] Actual Results: File upload button can should upload multiple files properly.		

Number: 39	Title: Remove uploaded files	Priority: High
ID: REMOVE_UPLOADED_FILES		
Description: Ensure that instructor can remove any of the uploaded files.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Clicks on file upload button. Uploads multiple files. Removes a file.		
Expected Results: Instructor should remove any of the uploaded files.		
Created by: Hasan Muhammed		
Test Status: [Pass/Fail] Actual Results: Instructor should remove any of the uploaded files.		

Number: 40	Title: Post Button on Course Page	Priority: High
ID: POST_BUTTON		
Description: Ensure that post button is rendered properly and functions properly.		

Pre-condition: Instructor is logged in and on a course page.
Action(s)/Steps: User logs in. Clicks on a course. Puts texts for Prompt and Ideal answer. Clicks on Post button.
Expected Results: Post button works properly.
Created by: Hasan Muhammed
Test Status: [Pass/Fail] Actual Results: Post button works properly.

Number: 41	Title: File Upload button on Null text at Course Page	Priority: High
ID: FILE_UPLOAD_BUTTON_WITH_NONE_IN_PROMPT		
Description: Ensure that post button isn't clickable if there is no prompt text and no ideal answer text.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Attempts to click on post button		
Expected Results: Post button should not be clickable.		
Created by: Hasan Muhammed		
Test Status: [Pass/Fail] Actual Results: Post button is not clickable		

3.3. Student Front-End

Number: 42	Title: Activity Selection	Priority: High
ID: STUDENT_ACTIVITY_SELECTION		
Description: Confirm that students can select and access learning activities.		
Pre-condition: Available list of learning activities is accessible.		
Action(s)/Steps: Choose a learning activity from the list. Access the selected activity. Verify successful activity access.		
Expected Results: Student can view and interact with the selected learning activity.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Student can view and interact with the selected learning activity.		

Number: 43	Title: Activity Submission	Priority: High
ID: STUDENT_ACTIVITY_SUBMISSION		
Description: Ensure that students can submit their responses to learning activities.		
Pre-condition: Student has completed the activity and is ready to submit.		
Action(s)/Steps: Enter response in the provided text box. Submit the response. Verify successful submission.		
Expected Results: Student's response is recorded and ready for feedback.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Student's response is recorded and ready for feedback.		

Number: 44	Title: Receive AI Feedback	Priority: High
ID: STUDENT_FEEDBACK_RECIEVED		
Description: Confirm that students receive relevant and useful feedback from the AI on their responses.		
Pre-condition: Student has submitted a response to an activity.		
Action(s)/Steps: Wait for AI-generated feedback on the submitted response. Review the AI feedback for accuracy and helpfulness. Verify that AI feedback aids in improving the response quality.		
Expected Results: Students gain insights from AI feedback to enhance learning outcomes.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Students gain insights from AI feedback to enhance learning outcomes.		

Number: 45	Title: Act on Feedback	Priority: High
ID: STUDENT_FEEDBACK_ACT_ON		
Description: Ensure that students can take actionable steps based on received feedback.		
Pre-condition: Student has received AI feedback on a submitted response.		
Action(s)/Steps: Review AI feedback suggestions for improvement. Make necessary revisions to the response based on AI feedback. Resubmit the revised response for further evaluation.		
Expected Results: Student demonstrates the ability to implement feedback for learning enhancement.		
Created by: Raj Ray Chaudhury		

Test Status: [Pass/Fail] Actual Results: Student demonstrates the ability to implement feedback for learning enhancement.

Number: 46	Title: Track Activity Progress	Priority: High
ID: STUDENT_ACTIVITY_PROGRESS		
Description: Confirm that students can track their progress on completed activities.		
Pre-condition: Student has completed one or more activities.		
Action(s)/Steps: Access the progress tracking feature. View the list of completed activities and their status (e.g., pending feedback, reviewed). Verify that progress information is accurate and up to date.		
Expected Results: Students can monitor their learning journey and activity completion status.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Students can monitor their learning journey and activity completion status.		

Number: 47	Title: Resubmit Activity Response	Priority: High
ID: STUDENT_ACTIVITY_RESUBMIT		
Description: Ensure that students can resubmit their response to an activity after making revisions.		
Pre-condition: Student has received feedback and revised the response.		
Action(s)/Steps: Access the activity with the revised response. Submit the revised response for re-evaluation. Verify that the resubmission process is successful.		
Expected Results: Updated response is submitted for further evaluation.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Updated response is submitted for further evaluation.		

Number: 48	Title: Print Activity Response	Priority: High
ID: STUDENT_ACTIVITY_PRINT		
Description: Confirm that students can print their responses or feedback received from instructors.		
Pre-condition: Student has completed activities and received feedback.		
Action(s)/Steps: Access the print feature for a completed activity or feedback. Choose the desired format (e.g., PDF, print-friendly version).		

Print the activity response or feedback.
Expected Results: Printed document accurately reflects the content of the activity response or feedback.
Created by: Raj Ray Chaudhury
Test Status: [Pass/Fail] Actual Results: Printed document accurately reflects the content of the activity response or feedback.

Number: 49	Title: Save Activity Response	Priority: High
ID: STUDENT_ACTIVITY_SAVE		
Description: Ensure that students can save their activity responses for future reference.		
Pre-condition: Completed activity response is available.		
Action(s)/Steps: Access the save feature for the activity response. Choose the desired format (e.g., save as a document, download as PDF). Save the activity response to the designated location.		
Expected Results: Saved document contains the activity response for later review or reference.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Saved document contains the activity response for later review or reference.		

Number: 50	Title: AI Learning Activity section on activity page	Priority: High
ID: AI_LEARN_ACTIVITY		
Description: Ensure that Learn activity is displayed properly.		
Pre-condition: Instructor is logged in and on a course page.		
Action(s)/Steps: User logs in. Clicks on a course. Clicks on an activity.		
Expected Results: Learn activity is displayed properly.		
Created by: Hasan Muhammed		
Test Status: [Pass/Fail] Actual Results: Learn activity is displayed properly.		

Number: 51	Title: AI Learn Activity Text box on Activity Page	Priority: High
ID: ACTIVITY_TEXT_BOX		
Description: Ensure that Learn activity is displaying text and working properly.		
Pre-condition: Instructor is logged in and on a course page.		

Action(s)/Steps: User logs in. Clicks on a course. Clicks on an activity. Uses AI activity text box.
Expected Results: Activity text box is displayed and works properly.
Created by: Hasan Muhammed
Test Status: [Pass/Fail] Actual Results: Activity text box is displayed and works properly.

Number: 52	Title: AI ask button	Priority: High
ID: AI_ASK_BUTTON		
Description: Ensure that AI ask button leads to getting AI response.		
Pre-condition: Instructor is logged in and on an activity page.		
Action(s)/Steps: User logs in. Clicks on a course. Clicks on an activity. Enters an answer in AI text box. Clicks AI ask button.		
Expected Results: Ask AI button leads to AI response being displayed.		
Created by: Hasan Muhammed		
Test Status: [Pass/Fail] Actual Results: Ask AI button leads to AI response being displayed.		

3.4. Backend Database

Number: 53	Title: AI Learn Database Connectivity Check	Priority: High
ID: DB_AILEARN_CONNECTIVITY_CHECK		
Description: Verify that the AI Learn application can establish a connection to the backend database.		
Pre-condition: AI Learn application is running and attempting to connect to the AI Learn database.		
Action(s)/Steps: Attempt to establish a connection to the AI Learn backend database. Verify that the connection is successfully established without errors.		
Expected Results: AI Learn application can communicate with the AI Learn backend database for data operations.		
Created by: Raj Ray Chaudhury		

Test Status: [Pass/Fail] Actual Results: AI Learn application can communicate with the AI Learn backend database for data operations.

Number: 54	Title: AI Learn Data Insertion into Database	Priority: High
ID: DB_AILEARN_DATA_INSERTION		
Description: Ensure that data can be successfully inserted into the AI Learn backend database.		
Pre-condition: AI Learn application has valid data to insert into the database.		
Action(s)/Steps: Insert test data related to learning activities, user information, and feedback into the AI Learn backend database using the application. Verify that the data is inserted correctly and reflects in database queries.		
Expected Results: Data related to AI Learn activities, users, and feedback is successfully stored in the AI Learn backend database and retrievable for future use.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Data related to AI Learn activities, users, and feedback is successfully stored in the AI Learn backend database and retrievable for future use.		

Number: 55	Title: AI Learn Data Retrieval from Database	Priority: High
ID: DB_AILEARN_DATA_RETRIEVAL		
Description: Confirm that data can be retrieved accurately from the AI Learn backend database.		
Pre-condition: AI Learn backend database contains relevant data for retrieval.		
Action(s)/Steps: Execute queries to retrieve specific data related to learning activities, user information, and feedback from the AI Learn backend database. Verify that the retrieved data matches the expected results.		
Expected Results: AI Learn application can fetch data related to activities, users, and feedback from the AI Learn backend database as required for functionality.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: AI Learn application can fetch data related to activities, users, and feedback from the AI Learn backend database as required for functionality.		

Number: 56	Title: AI Learn Data Integrity Check	Priority: High
ID: DB_AILEARN_DATA_INTEGRITY_CHECK		
Description: Ensure that data integrity is maintained within the AI Learn backend database.		

Pre-condition: AI Learn backend database contains existing data sets related to activities, users, and feedback.
Action(s)/Steps: Perform integrity checks on AI Learn database tables and relationships. Verify that data constraints (e.g., foreign key relationships, unique constraints) are enforced correctly.
Expected Results: Data integrity is maintained in the AI Learn backend database, preventing inconsistencies or invalid entries related to activities, users, and feedback.
Created by: Raj Ray Chaudhury
Test Status: [Pass/Fail] Actual Results: Data integrity is maintained in the AI Learn backend database, preventing inconsistencies or invalid entries related to activities, users, and feedback.

Number: 57	Title: AI Learn Database Performance Test	Priority: Low
ID: DB_AILEARN_PERFORMANCE_TEST		
Description: Evaluate the performance of database operations related to AI Learn under varying loads.		
Pre-condition: AI Learn application is under normal operating conditions.		
Action(s)/Steps: Execute a series of database operations (e.g., data insertion, retrieval, updates related to activities, users, and feedback) under increasing loads. Measure response times and resource utilization of the AI Learn backend database. Analyze the scalability and performance metrics of the AI Learn backend database.		
Expected Results: Determine if the AI Learn database can handle expected loads related to activities, users, and feedback and maintain performance standards.		
Created by: Raj Ray Chaudhury		
Test Status: [Pass/Fail] Actual Results: Determine if the AI Learn database can handle expected loads related to activities, users, and feedback and maintain performance standards.		

Number: 58	Title: Error Handling for Network Issues During API Calls	Priority: Medium
ID: ERROR_HANDLING_NETWORK_API		
Description: Simulate network connectivity issues during communication with the OpenAI API and verify the application's behavior.		
Pre-condition: Introduce simulated network delays or disruptions.		
Action(s)/Steps: Trigger a functionality that utilizes the OpenAI API while network issues are simulated. Verify that the application displays appropriate error messages indicating network connectivity problems.		

Verify that the application attempts to re-establish connection and retry the API call after a reasonable timeout.
Expected Results: The application gracefully handles network issues, displays clear error messages, and attempts to recover by reconnecting and retrying the API call.
Created by: Monish Barot
Test Status: [Pass/Fail] Actual Results: The application gracefully handles network issues, displays clear error messages, and attempts to recover by reconnecting and retrying the API call.

Number: 59	Title: Learning Activity Data Validation	Priority: Medium
ID: ACTIVITY_DATA_VALIDATION		
Description: Verify that the backend API enforces data validation rules for learning activity creation and editing.		
Pre-condition: Define data validation rules for various learning activity fields (e.g., question format, answer options, time limits).		
Action(s)/Steps: Attempt to create or edit learning activities with invalid data (e.g., missing fields, incorrect format). Verify that the API rejects invalid data submissions and displays appropriate error messages.		
Expected Results: The API enforces data validation rules, preventing the creation or modification of learning activities with invalid data.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The API enforces data validation rules, preventing the creation or modification of learning activities with invalid data.		

Number: 60	Title: Database Consistency Checks	Priority: Low
ID: DATABASE_CONSISTENCY		
Description: Verify that data manipulation through the backend API is reflected consistently across the database.		
Pre-condition: Define a set of test data for creating, editing, and deleting learning activities and student accounts.		
Action(s)/Steps: Perform various CRUD operations (Create, Read, Update, Delete) on learning activities and student accounts through the API. Verify that the database reflects the changes accurately and consistently across all tables.		
Expected Results: Database data remains consistent with the actions performed through the API, reflecting all CRUD operations accurately.		
Created by: Monish Barot		
Test Status: [Pass/Fail]		

Actual Results: Database data remains consistent with the actions performed through the API, reflecting all CRUD operations accurately.

3.5. OpenAI Flask API

Number: 61	Title: OpenAI API Key Validation	Priority: High
ID: KEY_VALIDATION		
Description: Verify that the application successfully authenticates with the OpenAI API using a valid API key.		
Pre-condition: A valid OpenAI API key is configured in the application settings.		
Action(s)/Steps: Trigger a functionality that utilizes the OpenAI API (e.g., student submits a response). Verify successful API call and response with a 200 status code.		
Expected Results: The application successfully connects to the OpenAI API and receives a valid response.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The application successfully connects to the OpenAI API and receives a valid response.		

Number: 62	Title: OpenAI Model Selection	Priority: High
ID: MODEL_SELECTION		
Description: Verify that the application selects the appropriate OpenAI model based on the configured settings for the learning activity.		
Pre-condition: Different OpenAI models are available and configured for specific learning activity types.		
Action(s)/Steps: Set the OpenAI model for a specific learning activity type. Trigger a functionality that utilizes the OpenAI API for that learning activity type. Verify that the API call specifies the chosen OpenAI model in the request parameters.		
Expected Results: The application uses the correct OpenAI model based on the learning activity configuration.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The application uses the correct OpenAI model based on the learning activity configuration.		

Number: 63	Title: AI Response Processing	Priority: Medium
ID: RESPONSE_PROCESSING		
Description: Verify that the application correctly processes the AI response received from the OpenAI API and presents it to the user		
Pre-condition: A valid student response is submitted and processed by the OpenAI API.		
Action(s)/Steps: Trigger a student response submission. Verify the application receives an AI response from the OpenAI API. Verify that the AI response is displayed to the student in a clear and understandable format.		
Expected Results: The AI response is received, processed, and presented to the student accurately.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The AI response is received, processed, and presented to the student accurately.		

Number: 64	Title: Error Handling for OpenAI Failures	Priority: High
ID: OPENAI_ERROR_HANDLING		
Description: Verify that the application handles potential errors or failures during communication with the OpenAI API gracefully.		
Pre-condition: Simulate various scenarios that might cause OpenAI API errors (e.g., invalid API key, network issues).		
Action(s)/Steps: Simulate an error scenario during OpenAI API communication. Verify that the application displays appropriate error messages to the user. Verify that the application logs the error details for further investigation.		
Expected Results: The application displays user-friendly error messages and logs detailed information about OpenAI API failures.		
Created by: Monish Barot		
Test Status: [Pass/Fail] Actual Results: The application displays user-friendly error messages and logs detailed information about OpenAI API failures.		

Number: 65	Title: OpenAI Performance Monitoring	Priority: Low
ID: AI_PERFORMANCE_MONITORING		
Description: Monitor the performance of the chosen OpenAI model over time, analyzing its response accuracy and consistency.		

Pre-condition: Define metrics for measuring model performance (e.g., accuracy, response time, bias detection).
Action(s)/Steps: Collect data on AI responses generated for various learning activities and student submissions. Analyze the collected data to identify trends in model performance over time. Investigate potential issues or biases detected in the model's responses.
Expected Results: The OpenAI model maintains consistent performance within acceptable thresholds, with minimal bias detected in its responses.
Created by: Monish Barot
Test Status: [Pass/Fail] Actual Results: The OpenAI model maintains consistent performance within acceptable thresholds, with minimal bias detected in its responses.

4. Risks and Assumptions

4.1. Accuracy

Due to the new and constantly developing nature of AI, the AI being utilized currently might not provide the most optimal or effective method of coaching in the future. Due to this uncertain nature, changes might need to be made in the future as the technology used is improved. It will be assumed as of current, that the AI model will remain consistent throughout testing.

4.2. Time

The completion of the testing and the completed product needs to be finished by May 22nd, 2024. Since a final video needs to be completed, the testing of major systems and bugs must be finished by May 12th, 2024, to allow optimal time. With this, certain tasks that would need to be completed may take priority over some testing.