

**TA:** Cory Paik

**Recitation Section:** -014

**Team Number:** 3

**Team Name:** Pretty Dandy Folks (PDF)

**Positive Test Cases:**

1. Login with the correct password
2. Upload correct file format to summarize (.pdf)

**Negative Test Cases:**

1. Login with the incorrect password
2. Upload incorrect file format to summarize (such as an image file, like .png)

#1

**Objective:**

Define what is being tested. For example, "To validate the one-click checkout process enables users to quickly and efficiently complete a purchase without navigating through multiple pages."

API routes for each of our pages:

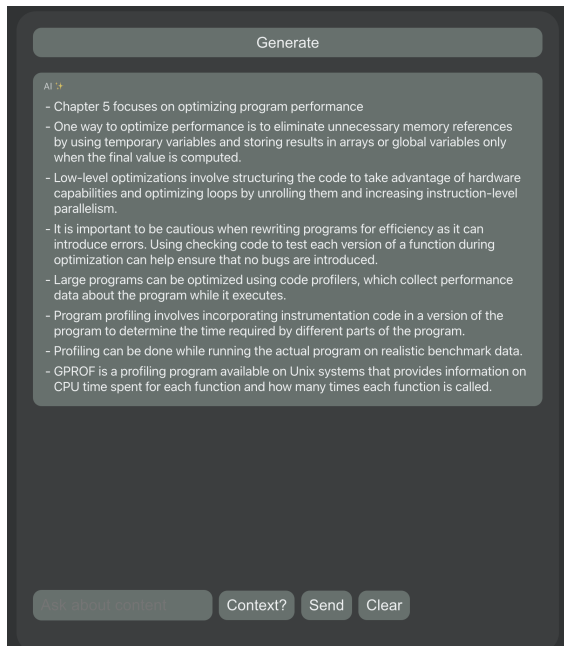
All of our pages, like the login page, will be tested by putting test cases in postman where we can authenticate the user (such as producing a message when authentication fails and prompts them to register instead).

Chat feature

We can just do test cases once we have completed the UI by using the feature...

We can also put positive and negative test cases in postman to see if the output is what we desired

Test all the buttons.

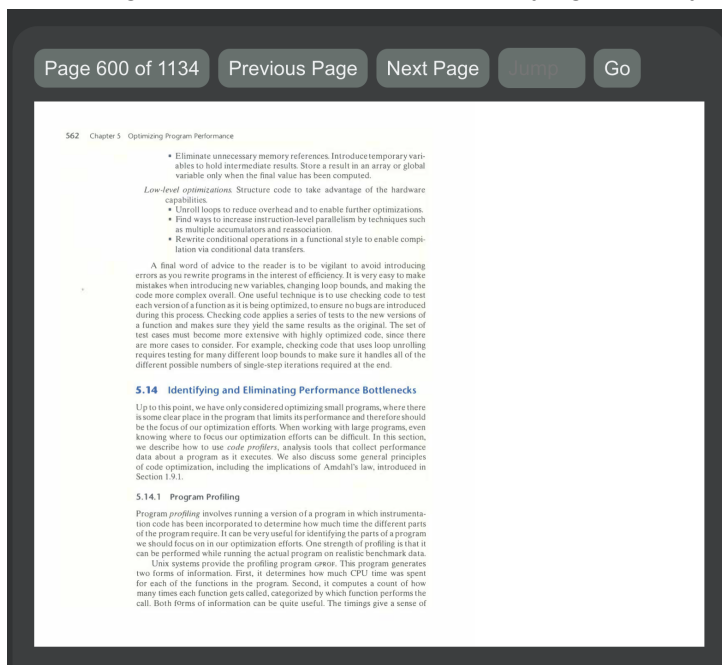


## PDF Viewer

We can test the functionality of the PDF viewer by making sure the file is in the correct format (.pdf), and producing an error message if not.

Using all the buttons associated with the PDF Viewer

Examining the output if the PDF is displaying correctly and properly



1. Create 1 document per team, within the **milestones** folder in the project directory, that describes how, at least, 3 features within your finished product will be tested.
2. The test plans should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.
3. The test plans should include a description of the test data that will be used to test the feature.
4. The test plans should include a description of the test environment that will be used to test the feature.
5. The test plans should include a description of the test results that will be used to test the feature.
6. The test plan should include information about the user acceptance testers.

| Rubric                       |   |        |
|------------------------------|---|--------|
|                              | Description   | Points |
| Part A: Automated Test Cases | Positive and negative test cases written for the <code>/login</code> endpoint on the server           | 20     |
| Part A: UAT Plan             | The test plan is well written for at least 3 features and includes all the necessary information      | 40     |
| Part B: Automated Test Cases | Positive and negative test cases written for an endpoint other than <code>/login</code> on the server | 20     |
| In class check-in            | Show your work to TA or CM.   | 20     |
|                              | Total   | 100    |

# Chat UAT

## Objective:

To validate the chat element allows users to view generated summaries from the ai as well as ask questions without any hiccups, hurdles or inconsistencies.

## Scope:

Testing will cover generating summaries, sending messages for the AI to answer, specifying whether context should be used and if we are able to successfully clear the chat

## Test Criteria:

Acceptance criteria: The element performs its features as it is described and the output is what we expect. The experience should be seamless.

Exit criteria: All critical problems are resolved, feature functions as it is described, acceptance criteria is met.

**Features to be tested:**

Generate summary button  
Viewability, and visibility of the messages  
Chat input field  
Send button  
Context toggle  
Chat clearing button

**Test Cases:**

Create specific test cases for the functionalities. For instance:

TC1: Test generate summary and verify that output is indeed summarizing the page  
TC2: Test generate summary does not work with an empty pdf page  
TC3: Ensure chat input box has proper input validation (does not work when field is empty)  
TC4: Send button works, sending with enter key works  
TC5: Response from sent message uses sent message and previous conversation as context  
TC6: Context toggle, requests response without PDF page context  
TC7: Clear button actually clears the messages

**Environment:**

Testing features such as getting responses from AI can be completed on postman on the server.  
Remaining features would have to be tested on the client side.

**Entry Criteria:**

All required routes have been implemented  
Main is updated with these routes  
Test data is prepared

**Exit Criteria:**

All test cases are executed, and results are documented.  
Any defects found are reported, addressed, and re-tested.  
The feature complies with the acceptance criteria.

# Navbar UAT

## Objective:

To validate the navbar allows the user to switch pages easily once they are logged in.

## Scope:

Testing will cover all the different buttons in the Navbar to ensure they are functioning correctly and are clear and easy for the user to understand. The drop down function will look good and allow users to use all the options.

## Test Criteria:

Acceptance criteria: The navbar is clear and easy to use, and every button leads to the page it is supposed to. The experience should be seamless.

Exit criteria: All buttons function as expected.

## Features to be tested:

Upload leads to Upload page

History button leads to a page displaying the user's upload and question history.

Help button displays a page explaining how to use the PDF.ai

When clicking on the drop down menu:

- Displays the users username

- Has an option that allows user to upload a custom API key

- Has a logout button that logs out the user and leads to a login page.

## Test Cases:

Create specific test cases for the functionalities. For instance:

TC1: "Test that the Upload button leads to the Upload page, which is the pdf viewer."

TC2: "Test that the History button leads to the History page, which allows the user to view their past uploads and questions."

TC3: "Test that the Help button leads to the Help page."

TC4: "Check that the drop down menu functions, and displays the user's username at the top."

TC5: "Test that the Settings button leads to the Settings page."

TC6: "Test that the Logout button leads to the logout page, as well as logs the user out."

## Environment:

Testing that the buttons lead to the correct pages can be completed in the local server, through the browser.

## Entry Criteria:

All required routes have been implemented  
Main is updated with these routes  
The user is logged in and can view the site.

**Exit Criteria:**

All test cases are executed, and results are documented.  
Any defects found are reported, addressed, and re-tested.  
The feature complies with the acceptance criteria.

# PDF Viewer UAT

**Objective:**

To validate that the uploaded file is in the .pdf format and allow users to view their previous uploads.

**Scope:**

Testing will cover checking the file formatting and making sure a user can view their previous pdf uploads as well as a history of their ai summarization, and seeing if we can successfully upload another pdf while putting the previous query in the user's history.

**Test Criteria:**

Acceptance criteria: Users can review their previous pdf summarizations before uploading a new one and confirming the summarization submission.

Exit criteria: The system successfully summarizes the uploaded pdf and saves previous queries to the users search history page.

**Features to be tested:**

One-click button availability on the upload page

Error message displays is a .pdf file isn't uploaded

Help button to instruct the user what to do

A load history button that will list all their previous pdf file upload names

**Test Cases:**

TC1: Test that the upload button functions properly and submits to pdf to be summarized

TC2: Test that the correct file format is being uploaded

TC3:

**Environment:**

Testing will be done through postman, although the appearance and buttons will need to be testing in the browser.