Milestone 4

Beta Launch, QA and Usability Testing and Final Commitment for Product Features

By Team 2

Global Distributed Software Development,

Winter Semester 23/24,

Fulda University of Applied Sciences,

Department of Applied Computer Science.

```
{ "project_name" : "MediaMajesty" }
```

Team 2

Team Lead:	Abbas Abbas; Email: abbas.abbas@informatik.hs-fulda.de
GitHub Master:	Shinu Donney
Frontend Lead:	Achraf Boudabous
Backend Lead:	Shifali Kalra
Fronted:	Duru Yilmaz
Backend:	Anwer Al-Dhify

Milestone 4

Beta Launch, QA and Usability Testing and Final Commitment for Product Features

Submitted on xx.xx.2024

Table of Contents

Product Summary

Name of the product: Media Majesty

ALL major committed functions:

1. User Registration
2. User Login
3. Media Upload
4. Media Search and Discovery
5. Messaging System
6. User Profile Management
7. Content Moderation
8. User ability to update versions of their media items, platform keeping a version history.
9. Admin ability to assign and adjust user's roles.
10. Users prompted to reset their passwords if forgotten.

Introducing MediaMajesty – Your Exclusive Gateway to a World of Academic Excellence!

In the dynamic landscape of digital education, MediaMajesty stands as a beacon of innovation, catering specifically to the unique needs of Fulda University. This groundbreaking platform is not just a tool; it's a transformative experience, enhancing collaboration, fostering innovation, and optimizing resource utilization for students and faculty alike.

MediaMajesty, exclusively crafted for Fulda University, creates a trusted space for seamless resource sharing and access, fostering a closely-knit environment that accelerates academic and creative pursuits. Our administrators meticulously ensure compliance with guidelines, providing a secure haven for your academic assets by eliminating inappropriate content. Prioritizing your privacy, MediaMajesty allows you to control access to your media, whether for sale or free use, empowering secure engagements with potential buyers. Designed for all technical backgrounds, MediaMajesty's simple interface lets you focus on content, not on navigating complicated platforms.

Join MediaMajesty – where exclusivity meets innovation, safety intertwines with freedom, and academic brilliance finds its home. Elevate your journey at Fulda University!

- URL to your product accessible to instructors, on deployment server: http://20.51.194.250:8000/

Usability Test Plan

Functions to be tested: Media Upload and Download Functionality

Test Objectives:

The primary objective of this usability test is to evaluate the user experience of the media upload and download functionality on the website. Specifically, we aim to assess the effectiveness, efficiency, and user satisfaction related to the process of uploading media (videos, images, audios, and PDFs) to Azure Blob Storage and subsequently viewing and downloading them on the website.

Test Background and Setup:

System Setup: The test will be conducted on the staging environment that replicates the production setup. The Azure Blob Storage integration and website functionalities will be identical to those in the live system.

Starting Point: Testers will start at the homepage of the website with a pre-logged-in user account.

Intended Users: The intended users for this test are individuals with basic to intermediate technical proficiency, representing the target audience for the website.

Measurement Focus (M4): The primary focus of measurement will be on user satisfaction, assessed through Likert scale evaluations. Other aspects, such as task completion time and error rates, will also be considered to gauge the overall usability.

Usability Task Description:

Task: Upload a video file to Azure Blob Storage using the provided functionality on the website.

Instructions:

a. Locate the "Upload Media" section on the website.

b. Select a video file from your local device.

c. Complete any required metadata or description fields.

d. Initiate the upload process.

e. Navigate to the "Media page" and locate the uploaded video.

f. Download the video file to your local device.

Measurement:

Effectiveness: Measure the percentage of successfully uploaded and downloaded media files.

Efficiency: Record the time taken by the user to complete the entire task.

Likert Scale Questions:

After completing the task, users were asked to rate their satisfaction on a Likert scale (1-5), where 1 indicates "Strongly Disagree" and 5 indicates "Strongly Agree."

Q1: The process of uploading media files was straightforward.

Q2: I could easily find and access the uploaded media in the Media Page.

Q3: The download process for media files was intuitive.

User	Q1: Upload Process	Q2: Media Page Access	Q3: Download Process
User 1	4	5	4
User 2	3	4	3
User 3	5	5	5

Likert Scale:

1 (Strongly Disagree) 2 (Disagree) 3 (Neutral) 4 (Agree) 5 (Strongly Agree)

Interpretation:

User1 found the upload process and media page access satisfactory but slightly less satisfied with the download process.

User2 had a moderately positive experience, finding the tasks somewhat agreeable.

User3 expressed high satisfaction with all aspects of the functionality.

QA Test Plan

Test Objectives:

The primary objective of the QA test is to ensure the reliability and functionality of the media upload and download feature on the website. This includes verifying the successful upload and download of various media types to and from Azure Blob Storage.

HW and SW Setup:

Environment: Staging environment replicating the production setup.

URL: http://20.51.194.250:8000/

Feature to be Tested: Media Upload and Download Functionality

QA Test Plan Table:

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (Browser 1 /

					Browser 2)
1	Upload Video File	Verify the successful upload of a video file.	Select a video file named "TestFile1" from the local device.	Confirm "TestFile1" is uploaded successfully.	PASS / PASS
2	Download Uploaded Video	Verify the successful download of an uploaded video.	Purchase a video named "PurchasedMedia1," and navigate to the download section.	Confirm "PurchasedMedia1" is downloaded successfully.	PASS / PASS
3	Metadata Completeness	Ensure metadata completeness during media upload.	Upload media with incomplete metadata (e.g., missing title, description) and name it "IncompleteMedia."	Verify that the system prompts for missing data.	PASS / PASS

Results of Testing:

Tested on Browser 1 (e.g., Chrome) and Browser 2 (e.g., Firefox).

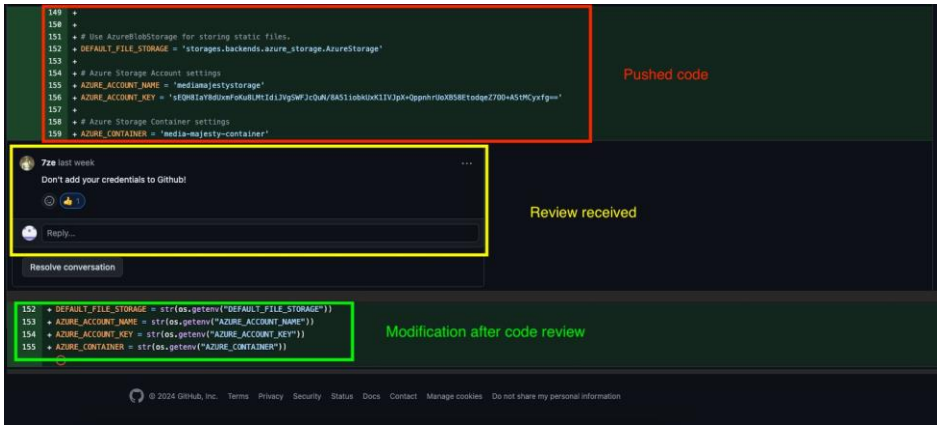
User / Test #	1 (Upload)	2 (Download)	3 (Metadata)
User 1	PASS	PASS	PASS
User 2	PASS	PASS	PASS
User 3	PASS	PASS	PASS

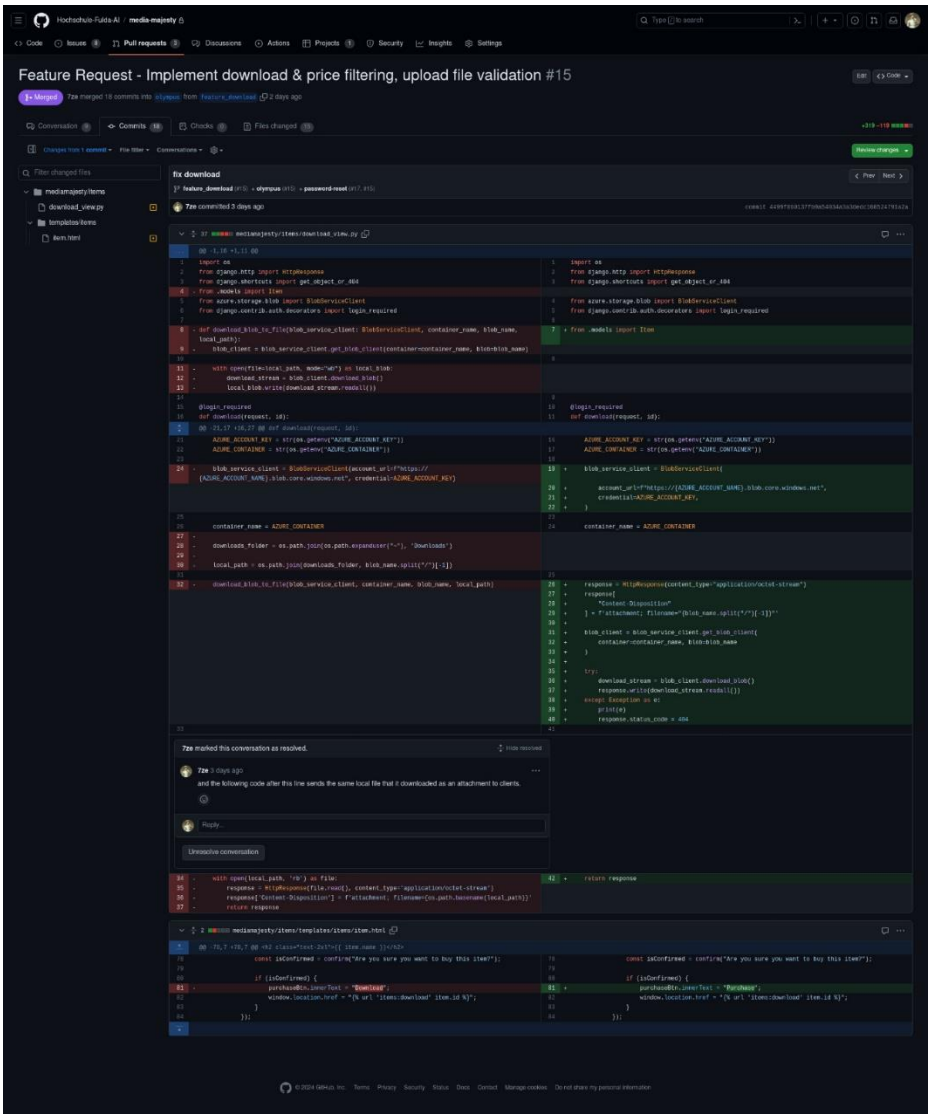
Interpretation:

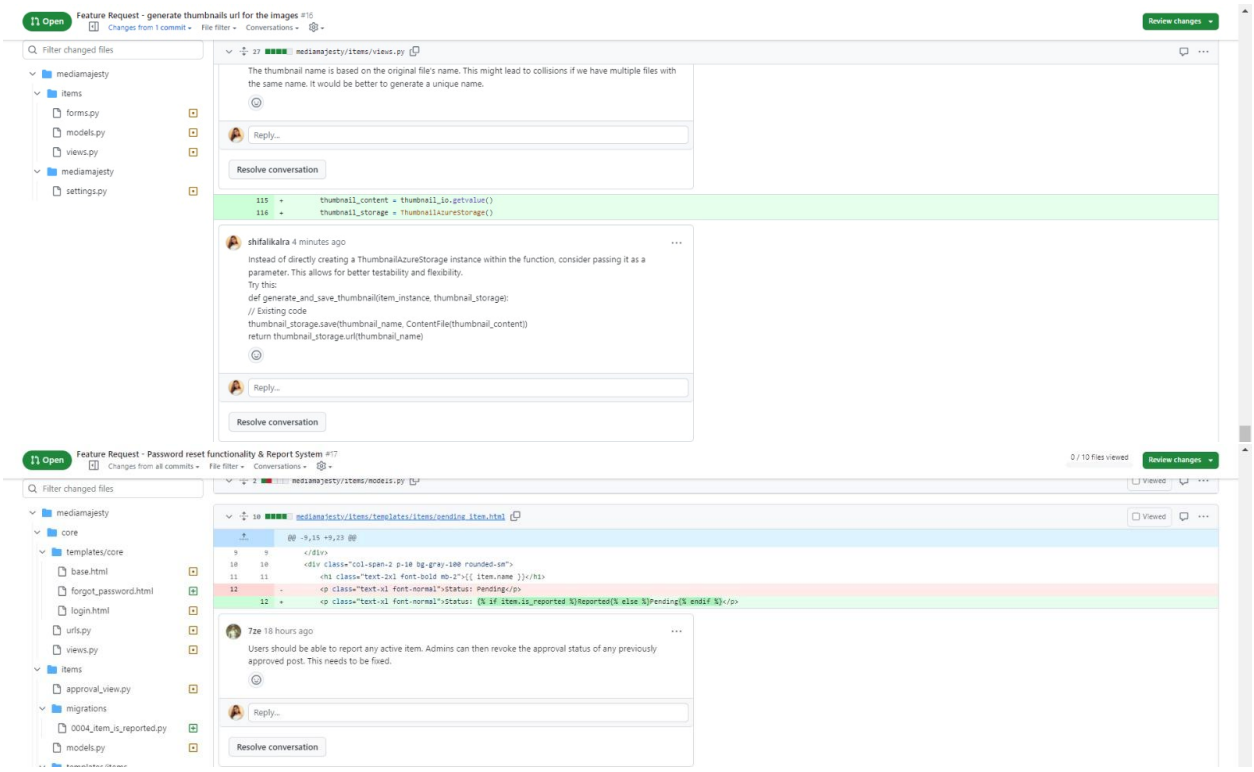
All test cases were successfully passed on both browsers, indicating the reliability and functionality of the media upload and download feature.

Code Review

Our team has adopted GitHub as the primary platform for our code review process. Every team member has submitted a substantial portion of the code for peer review. The review has been conducted entirely within the GitHub environment, utilizing the platform's built-in features such as pull requests and inline comments. The peer review includes constructive feedback, suggestions, and any necessary comments directly within the codebase on GitHub. Screenshots of the review process, capturing the inline comments and overall feedback, will be submitted to document and ensure a transparent and collaborative code review experience for the team. This approach leverages the version control capabilities of GitHub to enhance code quality and collaboration within our development workflow.


Pushed code
7ze last week
Don't add your credentials to Github!
Reply...
Resolve conversation
Review received
Modification after code review


Feature Request - Implement download & price filtering, upload file validation #15
fix download
7ze marked this conversation as resolved.
and the following code after this line sends the same local file that it downloaded as an attachment to clients.
Unresolve conversation



Self-Check on Best Practices for Security

Various protective measures are employed across key assets to fortify the platform against potential threats. In terms of user data, the focus is on averting unauthorized access and data breaches by regularly updating security protocols. Media content is shielded from unauthorized copying or distribution, as well as offensive material, through the implementation of content moderation and manual review processes. The messaging system is safeguarded against unauthorized access and phishing attacks with a comprehensive set of protections, including Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), SQL injection, Clickjacking, and SSL/HTTPS. User authentication credentials are protected against password cracking and unauthorized access through stolen credentials by storing hashed and salted passwords. Overall security measures include encryption of user passwords, data validation to prevent injection attacks, regular security audits, user role management, password reset mechanisms, and strict access controls. Notably, the encryption of passwords in the database and input data validation for the search bar are highlighted as crucial components of the security architecture. These collective measures ensure the confidentiality, integrity, and resilience of the system across various assets.

Self-Check: Adherence To Original Non-Functional Specs

List of non-Functional requirements:

- Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in Milestone 0. Application delivery shall be from chosen cloud server. ✓
- Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers. ✓
- All our selected application functions must render well on mobile devices. ✓
- Data shall be stored in the database on the team's deployment cloud server. ✓
- Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner. ✓
- No more than 50 concurrent users shall be accessing the application at any time. ✓
- Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users. ✓
- The language used shall be English (no localization needed) ✓
- Application shall be very easy to use and intuitive. ✓
- Application should follow established architecture patterns. ✓
- Application code and its repository shall be easy to inspect and maintain. ✓
- Google analytics shall be used (optional for Fulda teams)
- No email clients shall be allowed. ✓
- Pay functionality, if any (e.g., paying for goods and services) shall not be implemented nor simulated in UI. ✓
- Site security: basic best practices shall be applied (as covered in the class) for main data items. ✓
- Application shall be media rich (images, video etc.). Media formats shall be standard as used in the market today. ✓
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. ✓
- For code development and management, as well as documentation like formal milestones required in the class, each team shall use their own GitHub to be set-up by class instructors and started by each team during Milestone 0. ✓

- The application UI (WWW and mobile) shall prominently display the following exact text on all pages "Fulda University of Applied Sciences Software Engineering Project, Fall 2023 For Demonstration Only" at the top of the WWW page. (Important to not confuse this with a real application).. ✓