

Project

JSPR-Techies

Checkpoint 3

12/5/2023

Members

UI/UX Developer - Surya Subramani

Project Manager- Pallavi Dabade

Project Architect - Raksha Varahamurthy

Backend Developer - Jayani Sumanka Gerine

Table of Contents

Project Title	1
Current Iteration Updates.....	3
Functionality Completed by the current iteration.....	3
List of the User Stories Completed.....	6
Lesson Learned of Iteration.....	11
Lesson Learned.....	12
Next Final Iteration Updates.....	13
Final Iteration Coverage.....	19
GitHub Update.....	20
Jira Board Update.....	21
Meeting Updates.....	22

Current Iteration Updates - 11/20/23 to 12/4/23

Functionality completed by the end of the iteration

User Authentication:

Users will be able to login and authenticate with the system seamlessly. Developed a secure user authentication system, allowing users, including students, faculty, and alumni, to access personalized features and content upon logging in. Integration with University of Michigan credentials may be considered for added security and convenience.

Article Search Functionality:

Enable users to efficiently search for articles based on their preferences. The search functionality provides more accurate and relevant results. Consider incorporating filters, sorting options, and an intuitive user interface to streamline the article search process. This improvement aims to make content discovery more intuitive for users.

Top 5 Article Recommendations:

Users will receive a curated list of the top 5 articles based on their search criteria. Implemented an advanced recommendation system using algorithms or machine learning models. This system will analyze user behavior, preferences, and search history to generate a tailored list of top 5 articles. The goal is to enhance content discovery and engagement.

Article Viewing Enhancement:

Users should have a seamless experience when viewing articles from the list. The article viewing experience improved by optimizing the user interface, incorporating multimedia elements, and ensuring fast loading times. This enhancement aims to make the reading process engaging and user-friendly.

Screenshots of the completed functionality:

Welcome Wolverines

Click

THE M BLOGS

Type your UMID Here

Submit

10087236

123

=

Enter Keyword

- python
- data science

Top Articles

Python for Data Science: 8 Concepts You May Have Forgotten

Author: Conor Dewey

If you've ever found yourself looking up the same question, concept, or syntax over and over again when programming, you're not alone. I find myself doing this constantly. While it's not unnatural to look things up on StackOverflow or other resources, it does slow you down a good bit and raise questions as to your complete understanding of the language. We live in a world where there is a seemingly infinite amount of accessible, free resources looming just one search away at all times. However, this can be both a blessing and a curse. When not managed effectively, an over-reliance on these resources can build poor habits that will set you back long-term. Personally, I find myself pulling code from similar discussion threads several times, rather than taking the time to learn and solidify the concept so that I can...

Automated Feature Engineering in Python –

Machine Learning Exercises In Python, Part 1 –

How to build your own AlphaZero AI using Python

30 Amazing Python Projects for the Past Year

List of User Story Completed

▼ JSPRT Sprint 6 20 Nov – 5 Dec (5 issues)
📌 JSPRT-29 User Story - Display List of Article
📌 JSPRT-30 User Story - View Article from the list of articles
📌 JSPRT-34 User Story - Document Ranking - NLP task
📌 JSPRT-39 Redirect to Login Page or Article Page
📌 JSPRT-47 Frontend Interaction for User Decision after Article Reading

1. Display List of Articles

Size: 8

User Role: Student, Faculty, and Alumni of the University of Michigan

Goal: Users should be able to view a list of articles based on their search criteria.

Reason: Users can access the top 5 articles according to their preferences.

Trigger: Users click the submit button after entering words in the search box.

Preconditions: Users should enter keywords and click enter.

Postcondition: Users should be able to view a list of articles based on the search criteria.

Normal Operations:

- Users enter words in the search box and click the submit button.
- Users receive a list of articles matched to the search criteria.

Exception:

- Users cannot retrieve any article list if it does not match the search criteria.
- Users cannot retrieve any article list if they enter null.

Note: This user story focuses on providing users with the ability to access a curated list of articles based on their search criteria, enhancing the user experience by tailoring content to their preferences.

2. View Article from the List of Articles

Size: 3

User Role: Student, Faculty, and Alumni of the University of Michigan

Goal: Users should be able to search for and read relevant articles.

Reason: Users want to read articles matching their interests.

Trigger: Users click on a specific article from the list of articles.

Preconditions: Users should be able to click on a specific article to read.

Normal Operations:

- Users receive a list of articles matching the search criteria they entered.
- Users click on a specific article from the list.

Postcondition: Users should be able to read articles.

Exception:

- Users cannot view the article if there is no list of articles available that matches the user's search criteria.

Note: This user story focuses on the seamless transition from searching for relevant articles to reading them, ensuring a user-friendly experience.

3. Document Ranking - NLP Task

Size: 8

User Role: Developer

Goal: Recommend the top 5 articles to the user.

Reason: Technical requirement.

Trigger: Users enter each criterion in the search box and click the submit.

Preconditions: Ranking algorithm calculated to suggest the top 5 articles to users.

Normal Operations:

- Users enter each criterion in the search box and click the submit button.
- Users receive the top 5 articles matched to the search criteria they entered.

Postcondition: Users will be able to access those articles for defined keywords.

Exception:

- Users will not get the top 5 articles if there are not more than 5 articles available that match their search criteria.
- Users will not get any result if there is no article available that matches their search criteria.

4. User Redirection After Article Reading

Size: 5

User Role: Reader (Student, Faculty, Alumni)

Goal: Automatically redirect users to either the login page or the article search page upon completing an article reading session.

Reason: To enhance user experience by guiding them towards further engagement, either by logging in for personalized content or initiating a new search for additional articles.

Preconditions:

- Users have just finished reading an article on the JSPPR-Techies blog.

Triggers: The user reaches the end of an article.

Postcondition: Users are seamlessly redirected to either the login page or the article search page based on their engagement preference.

Normal Operations:

- Upon completing an article, users are presented with an option to either log in for personalized content or initiate a new search for additional articles.
- Users selecting the login option are directed to the login page where they can enter their credentials.

- Users opting for a new search are redirected to the article search page, ready to explore more content.

Exceptions:

- If the user has already logged in, they are automatically redirected to the article search page.
- Technical issues affecting the redirection functionality may result in users staying on the current page.

User Login Page Enhancement for JSPR-Techies Blog:

- **Size:** 5
- **User Role:** Student, Faculty, and Alumni of University of Michigan
- **Goal:** Enhance the HTML page for user registration on the JSPR-Techies blog.
- **Trigger:** Existing users accessing the JSPR-Techies blog enter their Umich ID and click the submit button.
- **Preconditions:** Users enter their Umich ID and click the submit button.
- **Postconditions:** Users successfully log in.
- **Normal Operations:**
 - Existing users navigate to the improved login page.
 - The enhanced page allows users to enter their unique usernames and email addresses for login.
- **Exception:**
 - If the entered credentials do not match, users will receive feedback indicating unsuccessful login attempts.

User Login Page Enhancement for JSPR-Techies Blog:

- **Size:** 5
- **User Role:** Student, Faculty, and Alumni of University of Michigan
- **Goal:** Enhance the HTML page for user registration on the JSPR-Techies blog.
- **Trigger:** Existing users accessing the JSPR-Techies blog enter their Umich ID and click the submit button.

- **Preconditions:** Users enter their Umich ID and click the submit button.
- **Postconditions:** Users successfully log in.
- **Normal Operations:**
 - Existing users navigate to the improved login page.
 - The enhanced page allows users to enter their unique usernames and email addresses for login.
- **Exception:**
 - If the entered credentials do not match, users will receive feedback indicating unsuccessful login attempts.

Major Topic Block Navigation Enhancement

- **Size:** 3
- **User Role:** Student, Faculty, and Alumni of the University of Michigan
- **Goal:** Enable users to click on major topic blocks on the search page to access the top five articles under that specific topic.
- **Reason:** To offer users a convenient way to explore and access the most relevant articles within a specific major topic.
- **Preconditions:**
 - Users should be logged into the JSPR-Tec blog.
 - **The major topic blocks are visibly displayed on the search page.**
- **Triggers:** The user accesses the search page on the JSPR-Techies blog.
- **Postcondition:** Users should be able to click on a major topic block to view the top five articles related to that specific topic.
- **Exceptions:**
 - If the user is not logged in, they will be prompted to log in before accessing major topic blocks.
 - In the event of technical issues affecting the functionality, users may face limitations in navigating and accessing top articles under major topics.

User Story: Frontend Interaction for User Decision after Article Reading

Size: 3

User Role: Reader (Student, Faculty, Alumni)

Goal: Engage users in deciding their next action after completing an article reading session.

Reason: To provide users with personalized options for a seamless post-reading experience.

Preconditions:

- Users have just finished reading an article on the JSPPR-Techies blog.

Triggers: The user reaches the end of an article.

Postcondition: Users make a selection from the presented options, guiding their post-reading experience.

Normal Operations:

- Upon completing an article, users are presented with a frontend interface displaying three major input blocks, each representing a different action.
- The first input block prompts users to "Continue Reading" the article, offering the option to delve deeper into the content.
- The second input block allows users to "Sign Out," providing a convenient option for those who have finished their reading session and wish to log out.
- The third input block encourages users to "Return to Article Search," enabling them to explore additional content.
- Users click on their chosen input block to proceed with the selected action.

Exceptions:

- If the user has already logged in, the option to "Sign Out" may be dynamically adjusted to reflect their logged-in status.
- Technical issues affecting the presentation of input blocks or user interactions may be addressed promptly through bug fixes or system improvements.

Lesson Learned of Iteration

What are the "lessons learned" at the end of this iteration? What would you do differently next time? Explain. (2 pts)

Structured HTML Files and CSS Styling:

Lesson: Initiating the development process with well-structured HTML files and the application of CSS styling is foundational. This approach ensures a clear and organized foundation for building the frontend, fostering easier collaboration among developers and facilitating future updates.

Python for HTML Parsing and Backend Connectivity:

Lesson: Leveraging Python for parsing HTML code and establishing connections with backend endpoints has proven to be a strategic choice. This integration enhances the accessibility of backend data within the HTML, promoting seamless communication between the frontend and backend components.

Browser Inspection for Functionality Verification:

Lesson: Regularly inspecting HTML files in the browser during development is a valuable practice. This step aids in verifying that the implemented functionality aligns with expectations, providing developers with real-time insights and facilitating prompt issue identification.

Use of Debuggers for Issue Resolution:

Lesson: Utilizing debuggers has emerged as a critical aspect of the development process. Debuggers are instrumental in pinpointing and resolving issues related to connectivity and functionality, streamlining the troubleshooting process and accelerating the overall development timeline.

Code Review and Continuous Enhancement Discussions:

Lesson: The completion of thorough code reviews, coupled with discussions on potential enhancements in both code and HTML visuals, is integral to the development lifecycle. This collaborative approach ensures the implementation of high-quality code, fosters knowledge sharing among team members, and sets the stage for continuous improvement.

Next Iteration - 11/5/23 - 12/8/23

List of the User Stories for the Next Iteration

▼ JSPRT Sprint 7	✎ Add dates	(7 issues)
✓	📌	JSPRT-40 Code Inspection
✓	📌	JSPRT-41 QA Testing
✓	📌	JSPRT-42 Stress Testing
✓	📌	JSPRT-43 UAT Testing
✓	📌	JSPRT-44 User Manual
✓	📌	JSPRT-45 Project Demo Preparation
✓	📌	JSPRT-46 Final Documentation

1. User Story - Code Inspection:

Objective: Review and analyze the codebase to identify and rectify potential issues, ensuring code quality and adherence to coding standards.

Description: A development team conducts a code inspection for JSPR-Tech blogs to ensure code quality, identify potential issues, and enhance overall maintainability.

Acceptance Criteria: The code inspection covers all functions and components within JSPR-Tech blogs. Each team member participates in the inspection process. Identified issues are documented, and action items are assigned for resolution. The inspection results in a list of recommendations for code improvement.

Tasks:

- Schedule a code inspection meeting.
- Conduct a collaborative code review session, discussing each part of JSPR-Techies.
- Distribute the relevant code files and documentation to team members in advance.
- Document identified issues, including bugs, code smells, and potential improvements.
- Assign tasks for issue resolution to responsible team members.
- Summarize inspection findings and recommendations.

2. User Story - QA Testing for Each Functionality

Objective: Conduct comprehensive Quality Assurance (QA) testing for each implemented functionality, including user authentication, article search, and recommendation systems. This involves functional testing, regression testing, and performance testing to verify the correctness of the features.

Description:

As a QA Tester, I want to conduct thorough testing for each implemented functionality, including user authentication, article search, and recommendation systems. This testing will involve functional, regression, and performance testing to ensure the correctness and reliability of the features.

Acceptance Criteria:

User Authentication:

- Verify that users can successfully register with valid credentials.
- Confirm that users can log in using registered UMICH credentials.
- Ensure proper error handling for invalid login attempts.
- Validate the security of user authentication processes.

Article Search:

- Test the search functionality for accurate and relevant results.
- Check if the search handles different input scenarios gracefully.
- Ensure that pagination, sorting, and filtering options work as expected.
- Verify the responsiveness of the search feature across various devices.

Recommendation Systems:

- Validate that personalized recommendations are generated based on user preferences.
- Test the recommendation engine's ability to adapt to user behavior changes.
- Ensure that recommended articles align with the user's interests.
- Evaluate the performance of the recommendation system under various user scenarios.

Functional Testing:

- Conduct end-to-end testing for each feature to confirm their proper integration.
- Verify the overall functionality of the system with positive and negative test cases.
- Ensure that user interactions with the system result in the expected outcomes.

Regression Testing:

- Perform regression testing after each software update to confirm existing functionalities are unaffected.
- Check for any unintended side effects on previously tested features.
- Confirm that bug fixes do not introduce new issues.

Performance Testing:

- Assess the system's performance under normal and peak load conditions.
- Identify and address any performance bottlenecks or latency issues.
- Confirm that the application can handle a significant number of simultaneous users.

Tasks:

1. Develop detailed test cases for user authentication, article search, and recommendation systems.
2. Execute functional tests to ensure each feature works as intended.
3. Implement regression test cases to verify the stability of existing functionalities.
4. Conduct performance testing using appropriate tools to assess system scalability.
5. Document and report any defects, issues, or performance bottlenecks found during testing.
6. Collaborate with the development team to prioritize and address identified issues.
7. Provide clear and concise testing documentation for future reference.

3. User Story - Stress Testing

Objective: Evaluate system performance under stress conditions, such as high user loads or concurrent requests. Identify potential bottlenecks and optimize system responsiveness.

Description:

As a Performance Tester, I want to evaluate the system's performance under stress conditions, such as high user loads or concurrent requests. The objective is to identify potential bottlenecks and optimize system responsiveness, ensuring that the tech blog can handle increased traffic without compromising user experience.

Acceptance Criteria:

High User Load:

- Simulate a high user load to assess the system's behavior under increased traffic.
- Measure response times and resource utilization during peak usage.
- Verify that the system maintains stability and responsiveness under stress.

Concurrent Requests:

- Generate concurrent requests to various endpoints to evaluate system concurrency.
- Assess how the system handles simultaneous user interactions.
- Identify and address any issues related to data consistency or race conditions.

Resource Utilization:

- Monitor CPU, memory, and network usage during stress testing.
- Identify resource-intensive operations and optimize as necessary.
- Ensure that the system scales gracefully with increased demands.

Bottleneck Identification:

- Analyze performance metrics to identify potential bottlenecks in the system.
- Determine if database queries, server processing, or external dependencies are limiting performance.
- Prioritize and address bottlenecks based on their impact on system responsiveness.

Optimization Strategies:

- Propose and implement optimizations to enhance system performance.

- Evaluate the effectiveness of optimizations in mitigating bottlenecks.
- Conduct iterative testing to validate the impact of optimizations.

Scalability Testing:

- Evaluate the system's ability to scale horizontally or vertically.
- Test the deployment of additional resources to handle increased loads.
- Ensure that the system architecture supports seamless scalability.

Tasks:

1. Develop stress testing scenarios that mimic real-world usage patterns.
2. Use performance testing tools to simulate high user loads and concurrent requests.
3. Monitor and collect performance metrics during stress testing sessions.
4. Identify and document any performance bottlenecks observed during testing.
5. Collaborate with the development team to implement optimizations based on findings.
6. Conduct post-optimization testing to validate the effectiveness of improvements.
7. Document and share recommendations for maintaining optimal system performance.

4. User Story - User Acceptance Testing (UAT)

Objective: Engage end-users in the testing process to validate that the implemented features meet their expectations and fulfill user requirements. Gather feedback for further refinement.

Description:

As a Product Owner, I want to engage end-users in the testing process to ensure that the implemented features of the tech blog meet their expectations and fulfill user requirements. The objective is to gather feedback for further refinement and ensure that the final product aligns with user needs.

Acceptance Criteria:

Test Coverage:

- Verify that all implemented features are included in the UAT process.
- Ensure comprehensive coverage of user journeys and key functionalities.

User Engagement:

- Actively involve end-users in the testing process.
- Provide clear instructions and scenarios to simulate real-world usage.

Feedback Collection:

- Create channels for users to submit feedback on the tested features.
- Gather qualitative feedback regarding usability, functionality, and overall user experience.

Bug Identification:

- Encourage users to report any identified bugs or unexpected behavior.
- Document and prioritize reported issues for resolution.

User Interface (UI) Evaluation:

- Validate that the UI aligns with user expectations and is intuitive.
- Gather feedback on the clarity of navigation and accessibility.

Content Review:

- Have users review and provide feedback on the quality and relevance of content.
- Ensure that the blog articles and information are valuable and meet user expectations.

Performance Assessment:

- Verify that the system performs well under typical user loads.
- Identify any performance-related issues reported by end-users.

Refinement Iteration:

- Collaborate with the development team to address reported issues and feedback.
- Iteratively refine features based on user input.

Tasks:

1. Communicate the UAT schedule and testing instructions to end-users.
2. Provide access to the UAT environment or staging version of the tech blog.
3. Monitor feedback channels and promptly address user inquiries during testing.
4. Document and categorize user feedback for further analysis.
5. Conduct regular meetings or surveys to gather additional insights from users.
6. Prioritize and address critical issues identified during UAT.
7. Iterate on feature refinements based on user feedback.
8. Confirm that all critical issues are resolved before final deployment.

Final Iteration Coverage

Describe the functionality that your (partially implemented) system will have at the end of this iteration.

- Code Inspection will be complete. Team will Review and analyze the codebase to identify and rectify potential issues, ensuring code quality and adherence to coding standards.
- QA Testing for Each Functionality testing will be conducted. Quality Assurance (QA) testing will be conducted for each implemented functionality, including user authentication, article search, and recommendation systems. And coverage of functional testing, regression testing, and performance testing to verify the correctness of the features.
- Stress Testing will be conducted to evaluate system performance under stress conditions, such as high user loads or concurrent requests. Identify potential bottlenecks and optimize system responsiveness.
- User Acceptance Testing (UAT) will be conducted by engaging end-users in the testing process to validate that the implemented features meet their expectations and fulfill user requirements. Gather feedback for further refinement.
- By completing testing and improvement based on the testing results, the aim is to ensure the system is well tested and ready to deliver the JSPPR-Techies blog.

GitHub Update

Active branches	
front_end	Updated 4 minutes ago by surya5289
initial-backend-code	Updated yesterday by jayanigerine
documentation	Updated 2 days ago by pallavid99
test	Updated 2 days ago by Raksha varahamurthy

GitHub Repository Link: https://github.com/Rakshasv18/Tech_Blog

Jira Board - Current Iteration

Projects / JSPRTechies

JSPRT Sprint 7



TO DO

IN PROGRESS 7

DONE ✓

▼ Issues without Epic (7 issues)

Code Inspection



JSPRT-40

1

R

Final Documentation



JSPRT-46

1

P

Project Demo Preparation



JSPRT-45

1

P

QA Testing



JSPRT-41

1

S

User Manual



JSPRT-44

1

J

Stress Testing



JSPRT-42

1

P

UAT Testing



JSPRT-43

1

P

Meeting Updates

Google Meet: meet.google.com/xjg-ijjr-bmz

Meeting Date and Time: Friday- 11/24/2023 - 10:30 a.m.

Meeting Duration: 1:00

Meeting Attendance: Jayani, Pallavi, Raksha, Surya

Meeting Points:

- Review Dashboard and Tasks
- Discuss any Roadblocks
- Review Documentation
- Code Review
- Completed Functionality Overview
- Code Review
- Remaining Task discussion

Google Meet: meet.google.com/xjg-ijjr-bmz

Meeting Date and Time: Friday- 12/1/2023 - 10:30 a.m.

Meeting Duration: 2:00

Meeting Attendance: Jayani, Pallavi, Raksha, Surya

Meeting Points:

- Review Dashboard and Tasks
- Discuss any Roadblocks
- Review Documentation
- Discuss about Testing
- Discussion about Demo Preparation

Google Meet: meet.google.com/xjg-ijjr-bmz

Meeting Date and Time: Friday- 12/5/2023 - 10:30 a.m.

Meeting Duration: 1:00

Meeting Attendance: Jayani, Pallavi, Raksha, Surya

Meeting Points:

- Review Professors's queries
- Review Documentation
- Discuss progress