DigiClips

General/Public

sdmay25-05

Jan 16, 2024 - Dec 20, 2024

https://sdmay25-05.sd.ece.iastate.edu/

Faculty Advisor: Dr. Ashfaq Khokhar

Team Members/Role:

Eshanth Chinthireddy - Testing

ec3@iastate.edu

Nguyen Do - Team organization

ndo@iastate.edu

Edmund Lim - Client Interaction

elim655@iastate.edu

Niharika Pathuri - Documentation

npathuri@iastate.edu

Varun Yeduru - Individual component Design

yvarun79@iastate.edu

Executive Summary

The DigiClips Media Search Engine project aims to provide a comprehensive media monitoring solution, focusing on accessibility for both professionals and the general public. In today's media landscape, where content is generated rapidly across platforms like television, radio, social media, and blogs, there is a critical need for efficient tools to help users stay informed. Existing solutions tend to be expensive, complex, or limited in functionality, which restricts access for many users. DigiClips addresses this gap by offering an ad-supported General Public Page, providing basic search capabilities for free while encouraging users to upgrade for premium features.

Problem and Importance

The problem lies in the overwhelming volume of media content and the lack of affordable, user-friendly tools for tracking relevant information. This is especially crucial for professionals such as reporters, lawyers, and media analysts who rely on timely information for decision-making. Without a suitable solution, users risk missing critical updates, leading to lost opportunities and inefficient workflows.

Key Design Requirements

1. Search Functionality: Basic keyword search across multiple media platforms (TV, radio, web, social media).
2. Ad-Supported Model: Display ads for non-subscribers to sustain free access.
3. Search Limits: Non-subscribers are limited to 5 searches per day.
4. User Differentiation: Distinguish between subscribers and non-subscribers, providing premium features to paid users.
5. Call-to-Action (CTA): Encourage subscription upgrades through strategically placed prompts.

Design Overview and Technologies

The system comprises a frontend interface developed with Angular 18, a backend server using Node.js, and a MySQL database for data storage. The backend handles search requests, user authentication, and role management. Non-subscribers experience limited functionality, including a daily search cap and ad displays, while subscribers enjoy unlimited searches, advanced analytics, and notifications. The platform is hosted on Amazon Lightsail for scalability and reliability.

Progress Thus Far

- Frontend and Backend Development: Initial setup of the frontend interface and backend server.
- User Role System: Implementation of subscriber and non-subscriber differentiation.
- Prototype: Developed a working prototype of the General Public Page with search limits, ad integration, and CTA prompts.
- Testing: Initial rounds of unit and integration testing have validated core functionalities.

Meeting Requirements and Addressing User Needs

The design successfully provides free access to basic search features while maintaining financial sustainability through ads. The intuitive interface and limited search functionality encourage non-subscribers to upgrade. Initial testing and client feedback indicate that the system meets core requirements, though improvements in scalability and user experience are ongoing.

Next Steps

- Scalability Optimization: Implement load balancing to handle increasing user traffic.
- Refine Ad Strategy: Optimize ad placement based on user feedback to balance revenue and usability.
- Enhanced Search Accuracy: Improve search algorithms for better relevance and efficiency.
- User Feedback: Conduct extensive user testing to refine features and address any remaining issues.

Learning Summary

Development Standards & Practices Used

Our team has adopted Agile project management to handle the project's progress and evolving requirements. By focusing on iterative development through manageable sprints, the team can implement, test, and refine features incrementally with continuous client feedback, mitigating risks and ensuring alignment with goals. Collaboration and adaptability are emphasized, supported by tools like GitHub for version control, Gantt chart for task organization, Google Meet for real-time communication, and Google Drive for document sharing. Weekly reviews and ongoing feedback ensure steady progress and the flexibility to adjust priorities as needed.

This project adheres to some engineering standards to ensure reliability, scalability, and security. It follows IEEE 23026:2023 for best practices in website engineering, focusing on usability, accessibility, and secure data handling; IEEE 29119 for comprehensive software testing to ensure system quality and performance through unit and end-to-end tests; and IEEE 27001 for information security management, implementing robust measures like encryption, access control, and risk management to protect sensitive user data.

Summary of Requirements

Functional Requirements

- 24/7 Availability: 99.9% uptime to ensure continuous access.
- Limit 5 searches per day for users.
- Display advertisements to generate revenue.
- Include a call-to-action feature to encourage subscriptions.
- Implement secure access with role-based authentication, two-factor verification, and data encryption.

UI Requirements

- Mobile-Friendly Design: Responsive layout for seamless use on various screen sizes.
- Prominent Search Interface with Filters: Easily accessible search bar with advanced filtering options.

User Experience (UX) Requirements

- Consistent, visually appealing interface across devices.
- Ease of Use: Intuitive navigation requiring minimal guidance.
- Responsiveness: Deliver search results within 5 seconds.

Integration and Compatibility Requirements

- Cross-Browser Compatibility: Support for Chrome, Firefox, Safari, and Edge.
- Integration with Third-Party Tools: Compatibility with ad providers and analytics tools.
- Front-End Framework: Built on Angular 18, requiring smooth integration with its modular, component-based design.

- Back-End Infrastructure: Align with MySQL database schema and Amazon LightSail hosting configurations for efficient data handling and deployment.

Applicable Courses from Iowa State University Curriculum

- COM S 227 Object Oriented Programming
- COM S 228 Introduction to Data Structures
- COM S 309 Software Development Practices
- COM S 319 Construction of User Interfaces
- COM S 363 Introduction to Database Management
- S E 317 Introduction to Software Testing
- S E 329 Software Project Management
- S E 339 Software Architecture and Design.

New Skills/Knowledge acquired that was not taught in courses

- Angular 18
- Amazon LightSail

Table of Contents

List of figures and Tables

Figures

Tables

1 Introduction

1.1 Problem Statement

The DigiClips Media Search Engine addresses a critical need for professionals and the general public who rely on real-time media monitoring and comprehensive analytics to stay informed and make timely decisions. In today's fast-paced world, where news, social media, podcasts, and other forms of digital content are generated at an unprecedented rate, it's becoming increasingly difficult for users to track relevant information across multiple platforms. Reporters, media analysts, legal professionals, and even everyday users need efficient tools to stay up-to-date on current events, media trends, and specific topics relevant to their interests.

Currently, existing media monitoring tools often fall short. They may be expensive, overly complex, or lack comprehensive coverage across traditional and digital media platforms. Users face the problem of missing crucial updates, spending too much time sifting through irrelevant content, or paying high fees for limited services. This lack of an integrated, user-friendly, and affordable solution results in missed opportunities and ineffective media tracking.

The rapidly changing digital world makes media tracking even harder right now. Professionals and individuals alike face the dual challenges of information overload and inadequate tools that are either unaffordable or fail to deliver necessary functionality. For example, a reporter might miss important breaking news, or a lawyer might overlook significant public opinion that could influence a case. Likewise, members of the general public may struggle to stay informed about specific topics of interest due to high costs or limited access to advanced tools.

Because of these problems, it's important to have a system that can monitor multiple media channels and make basic features available to everyone at no cost. DigiClips seeks to address this need by not only providing a professional-grade media search engine but also introducing a General Public Page. This feature allows users to register with their email and search for free, offering a limited set of functionalities supported by ad displays.

1.2 Intended Users

The DigiClip General Public Page offers a powerful yet accessible media analysis platform designed for everyone. By registering with just an email, users can access media insights from a wide range of sources, including digital media, newspapers, magazines, TV, and radio, completely free of charge. With easy-to-use keyword search, users can explore trending topics and gain valuable insights into media coverage without incurring any cost. Supported by non-intrusive ads, the platform ensures affordability while maintaining functionality. To enhance the user experience, DigiClip includes a compelling call-to-action feature, encouraging users to subscribe for premium benefits like unlimited searches, advanced analytics, and real-time updates.

A Lawyer Using the General Public Page

Use Case: A lawyer wants to stay informed about mentions of their clients in the media. They use the free version of the media search engine to perform keyword-based searches related to their clients. While this version doesn't include real-time updates, it allows the lawyer to browse through relevant content at their convenience. A call-to-action feature on the platform encourages the lawyer to subscribe for advanced features like real-time notifications and broader access to search tools.

Needs: The lawyer can search for client-related mentions across various media sources. The free version provides limited results, allowing them to get a basic overview. The system is easy to navigate, ensuring the lawyer can quickly find relevant information without extra effort.

Benefits: The general public page gives the lawyer a cost-effective way to monitor media content. While it doesn't provide real-time alerts, it serves as an entry point for exploring the platform's capabilities. The integrated call-to-action effectively communicates the value of a subscription, showing the lawyer how upgrading can enhance their ability to respond to media developments more proactively. By offering a smooth transition from basic to advanced features, the platform ensures user satisfaction and encourages continued engagement.

A Reporter Using the General Public Page

Use Case: A reporter uses the free version of the media search engine to track keywords related to ongoing stories and emerging issues. The general public page provides a straightforward way to search for relevant content across multiple platforms.

Needs: The reporter can use the platform to search for terms related to breaking news or ongoing stories. The free version provides limited results, helping them gather preliminary information. The intuitive interface allows the reporter to quickly search for and review information without delays, even in high-pressure situations.

Benefits: The general public page provides a cost-effective starting point for reporters who need basic media tracking capabilities. While it doesn't offer real-time alerts or deep customization, it introduces them to the platform's potential. The call-to-action effectively communicates how subscribing can enhance their reporting efficiency, offering tools like real-time updates and access to a broader range of sources. This encourages the reporter to transition to a paid subscription, ensuring they can stay ahead in the fast-paced media landscape while meeting their professional needs.

A Media Analyst Using the General Public Page

Use Case: A media analyst uses the free version of the media search engine to explore media trends and gather insights into content performance. The general public page provides a basic tool for discovering patterns and trends through keyword searches. The platform includes a call-to-action feature that highlights the benefits of subscribing for advanced features like customizable dashboards, detailed reports, and data visualization tools.

Needs: The analyst can search for topics and view limited results related to media interaction trends. While limited, the free version offers insights into general trends, helping the analyst identify broad patterns without deep filtering or customization.

Benefits: The general public page offers an affordable starting point for media analysts who need basic tools for exploring trends. Although it lacks advanced features like real-time tracking or detailed reporting, it introduces the analyst to the system's potential. The call-to-action effectively communicates the value of subscribing, showing how the premium version can enhance their ability to monitor trends, generate detailed insights, and make informed decisions. By providing a smooth transition from basic to advanced functionality, the platform ensures accessibility while encouraging long-term engagement and professional use.

2 Requirements, Constraints, And Standards

2.1 Requirements & Constraints

Requirements

These requirements ensure that the system meets user needs by delivering reliable, secure, and efficient media management services. The requirements are organized into functional, UI, UX, economy/market, and integration and compatibility categories.

Functional Requirements

- 24/7 Availability: The system must operate continuously with 99.9% uptime, ensuring users can access media content anytime.
- Limit 5 searches per day
- Display ad
- Call-to-action feature to encourage user to subscribe
- Secure Access and Data Management: Role-based authentication, two-factor verification, and data encryption must be implemented to secure user access and data integrity.

UI Requirements

- Mobile-Friendly Design: The platform must adjust to different screen sizes, allowing seamless access on mobile devices.
- Prominent Search Interface with Filters: A prominently placed search bar with advanced filters should enable users to refine results easily.

User Experience (UX) Requirements

- The interface should be consistent and visually appealing across all devices, using a modern design for clear navigation and usability.
- Ease of Use: Users should navigate the platform intuitively, with minimal guidance needed to access search and report functions.
- Responsiveness: The system should deliver search results within 5 seconds, ensuring a smooth experience across devices.

Integration and Compatibility Requirements

- Cross-Browser Compatibility: The platform must work seamlessly on all major browsers, including Chrome, Firefox, Safari, and Edge.
- Integration with Third-Party Tools: Enable integration with ad providers and analytics tools to extend the platform's functionality.
- Front-End Framework: The project is built on an existing Angular 18 front-end, requiring adherence to Angular's modular design principles. Any new features must integrate smoothly with the existing component-based architecture and maintain compatibility with the Angular ecosystem.
- Back-End Infrastructure: The backend leverages MySQL for the database and Amazon LightSail for hosting. Integration efforts should align with the current database schema and

hosting configurations, ensuring efficient data handling and minimal disruptions during deployments.

Constraints

The following constraints have been identified and categorized under Performance, Security/Network and Integrating based on the project's requirements. These constraints are essential to ensuring the DigiFlip system functions as expected, delivering real-time updates, secure access, and reliable operation in dynamic media environments.

Performance:

- System latency for real-time updates and interactions should not exceed 5 seconds.
- Must maintain an uptime of 99.9%.
- Analytics and reporting must be generated in under 10 seconds.

Security/Network

- Compliance with ISO/IEC 27001 for secure data management.
- Ensure fast wireless access and data transmission.

Integrating with an existing system

- Update and implement front-end using Angular 18
- Deploy back-end using MySQL and Amazon LightSail

2.2 Engineering Standards

The DigiClip Media Search Engine project is built on a foundation of robust engineering standards to ensure reliability, scalability, and user satisfaction. By adhering to recognized industry standards, the project guarantees high-quality software development practices, secure data handling, and user-centric design.

1. IEEE 23026:2023 - Engineering and Management of Websites for Systems, Software, and Services Information

a. Description: This standard provides guidelines for the engineering and management of websites, specifically focusing on systems, software, and services. It ensures that website-related processes follow best practices for design, development, and management. The standard also covers the proper handling of user data, usability, accessibility, and security.

b. Relevance: For the media search engine project, this standard applies directly to the development and management of the website and its backend systems. It provides guidelines for ensuring security, performance, and user data management, all of which are critical in handling media data for the general public and subscribers.

2. IEEE 29119 - Software and Systems Engineering – Software Testing

a. Description: This standard specifies a comprehensive framework for software testing. It provides processes for test management, test documentation, test design, and dynamic testing. The goal is to ensure that all aspects of software testing are covered thoroughly to verify the system's quality, performance, and reliability.

b. Relevance: Given that the project involves both front-end and back-end development, it is crucial to follow a standard for software testing to ensure the robustness of the media search engine. This includes writing and executing unit tests for individual components and end-to-end tests to simulate user interactions, ensuring the application behaves as expected.

3. IEEE 27001 - Information Security Management Systems

a. Description: This standard focuses on establishing, implementing, and maintaining an information security management system (ISMS). It helps organizations secure sensitive data and ensure confidentiality, integrity, and availability. The standard includes guidelines for risk management, access control, incident response, and data encryption.

b. Relevance: Since the media search engine involves storing user profiles, media searches, and potentially sensitive information, adhering to information security standards is vital. This standard will guide the implementation of robust security measures such as data encryption, access control, and secure communication protocols.

3 Project Plan

3.1 Project Management/Tracking Procedures

Our team has adopted an Agile project management style for the DigiClips project. Given the dynamic nature of this project and the need for continuous feedback from both the client and end-users, Agile offers the flexibility required to adapt to evolving requirements. The project includes both frontend and backend development components, which benefit from an iterative approach where each sprint focuses on specific functionality, allowing us to refine the system incrementally based on user feedback and testing.

Justification for Agile Methodology:

- Iterative Development: Agile allows us to work on small, manageable sprints, making it easier to implement, test, and modify features as needed.
- Continuous Feedback: Frequent communication with the client ensures that any necessary adjustments can be made early in the development process.
- Risk Mitigation: Agile helps us address risks incrementally, allowing us to detect and resolve issues quickly, reducing potential project delays.
- Collaboration and Flexibility: Agile's emphasis on team collaboration and adaptability makes it easier for us to adjust tasks and priorities based on feedback and project needs.

Tracking Tools and Procedures:

To ensure efficient tracking and organization of tasks, our team will utilize the following tools:

- GitHub: Used as our main version control system for code management. We will create branches for specific features, review code changes, and maintain project history to track progress and facilitate collaboration.
- Google meet: Our primary communication platform for quick discussions and updates, Google meet will allow team members to stay connected, share progress, and address blockers in real time.
- Google Drive: We will use Google Drive for document collaboration, allowing team members to create, share, and edit project documentation and presentations in a centralized location.

Process:

- Weekly reviews to assess sprint progress and adjust tasks if needed.
- Continuous client feedback to ensure alignment with project goals.

3.2 Task Decomposition

1. **Verifying users**

Task #	Description	Dependent On	Purpose
1	Define user roles (Subscriber vs. Non-Subscriber)	N/A	Establish distinct user types for feature access differentiation
2	Implement user role assignment during registration	Task #1	Assign role upon user registration or login
3	Backend logic to check user role on login	Task #2	Validate the user's role in applying appropriate access restrictions
4	Enforce feature limitations for Non-Subscribers	Task #3	Limit features such as search counts and access to analytics
5	Display ads for Non-Subscribers	Task #3	Generate revenue by displaying ads to non-subscribers
6	Display subscription prompt for Non-Subscribers	Task #3	Encourage non-subscribers to upgrade with tailored prompts
7	Integration and System Testing	Task #3,4,5,6	Ensure features work as intended for each user type

Table 9: Verifying users

2. Advertisement Integration

Task #	Description	Dependent On
8	Research and select an ad provider (e.g., Google Ads)	N/A
9	Integrate ads into the page layout	Task #8
10	Implement ad visibility settings based on subscription	Task #9
11	Test ad performance and impact on page loading	Task #10

Table 10: Advertisement Integration

3. Search Limitations

Task #	Description	Dependent On
12	Define search count limit for non-subscribers	N/A
13	Implement backend tracking for daily search usage	Task #12
14	Create user notifications for approaching search limit	Task #13
15	Test daily reset functionality for search counts	Task #14

Table 11: Advertisement Integration

4. Basic Media Search Functionality

Task #	Description	Dependent On
16	Develop basic search interface (UI/UX design)	N/A
17	Implement media type filters (TV, Radio, etc.)	Task #16
18	Enable search functionality for media sources listed	Task #17
19	Test search feature and refine based on initial user feedback	Task #18

Table 12: Advertisement Integration

5. Subscription Encouragement

Task #	Description	Dependent On
20	Design “Upgrade to Subscribe” call-to-action (CTA)	N/A
21	Implement CTA visibility based on user’s subscription status	Task #20
22	Integrate subscription upgrade link functionality	Task #21

23	Test visibility of CTA and track click-through rates	Task #22

Table 13: Subscription Encouragement

This project will follow an Agile approach. Initial sprints focus on setting up advertisements, search limitations, and basic search functionality, with later sprints covering user feedback, accessibility, and translation.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Our project plan includes key milestones, measurable metrics, and evaluation criteria to track progress and ensure we meet our project goals. Below are the proposed milestones and metrics for the initial phases of the DigiClips project.

Milestone	Description	Metric	Evaluation Criteria
Phase 1: Project Initiation			
Milestone 1: Project Setup & Planning	Define project goals, set up environments, finalize project tools.	Completion rate of project plan tasks, tool setup time	Project plan is comprehensive and approved, tools are set up and accessible to all team members within the specified timeframe.
Phase 2: Development			
Milestone 2: Basic Media Search	Implement basic search interface and backend for media data.	Search query response time, search accuracy, search coverage	Search results are relevant, accurate, and returned within 2 seconds. Search covers a wide range of media types.
Milestone 3: User Role & Access Control	Define/implement user roles (Subscriber vs. Non-Subscriber).	Successful login attempts, error rate, feature access	Users can successfully log in and access features based on their roles. Unauthorized access is prevented.
Milestone 4: Advertisement Integration	Integrate advertisements; control visibility based on subscription status.	Ad load time, ad click-through rate, ad revenue	Ads load quickly and are relevant to the content. Ad revenue meets target goals.

Milestone 5: Search Limit Enforcement	Implement daily search limits for non-subscribers.	Number of search attempts, number of limit notifications, user satisfaction	Non-subscribers are notified before reaching their daily limit. Limit resets correctly. Users are satisfied with the notification system.
Phase 3: Testing and Deployment			
Milestone 6: Subscription Upgrade CTA	Design and implement call-to-action for non-subscriber upgrades.	Click-through rate, conversion rate, revenue generated	The CTA is prominent and visually appealing. It drives a significant number of users to upgrade their subscription.
Milestone 7: User Testing & Feedback	Conduct user testing for interface and feature feedback.	Number of user testers, feedback survey completion rate, bug reports	A diverse group of users provides feedback on the usability, functionality, and overall user experience. Feedback is analyzed and incorporated into the product.
Milestone 8: Final Testing & Optimization	Ensure system stability, performance, and user satisfaction.	System uptime, bug reports, user satisfaction surveys	The system is stable, performs efficiently, and meets all functional and non-functional requirements. User satisfaction is high.

Table 6: Project Milestones, Metrics, and Evaluation Criteria

3.4 Project Timeline/Schedule

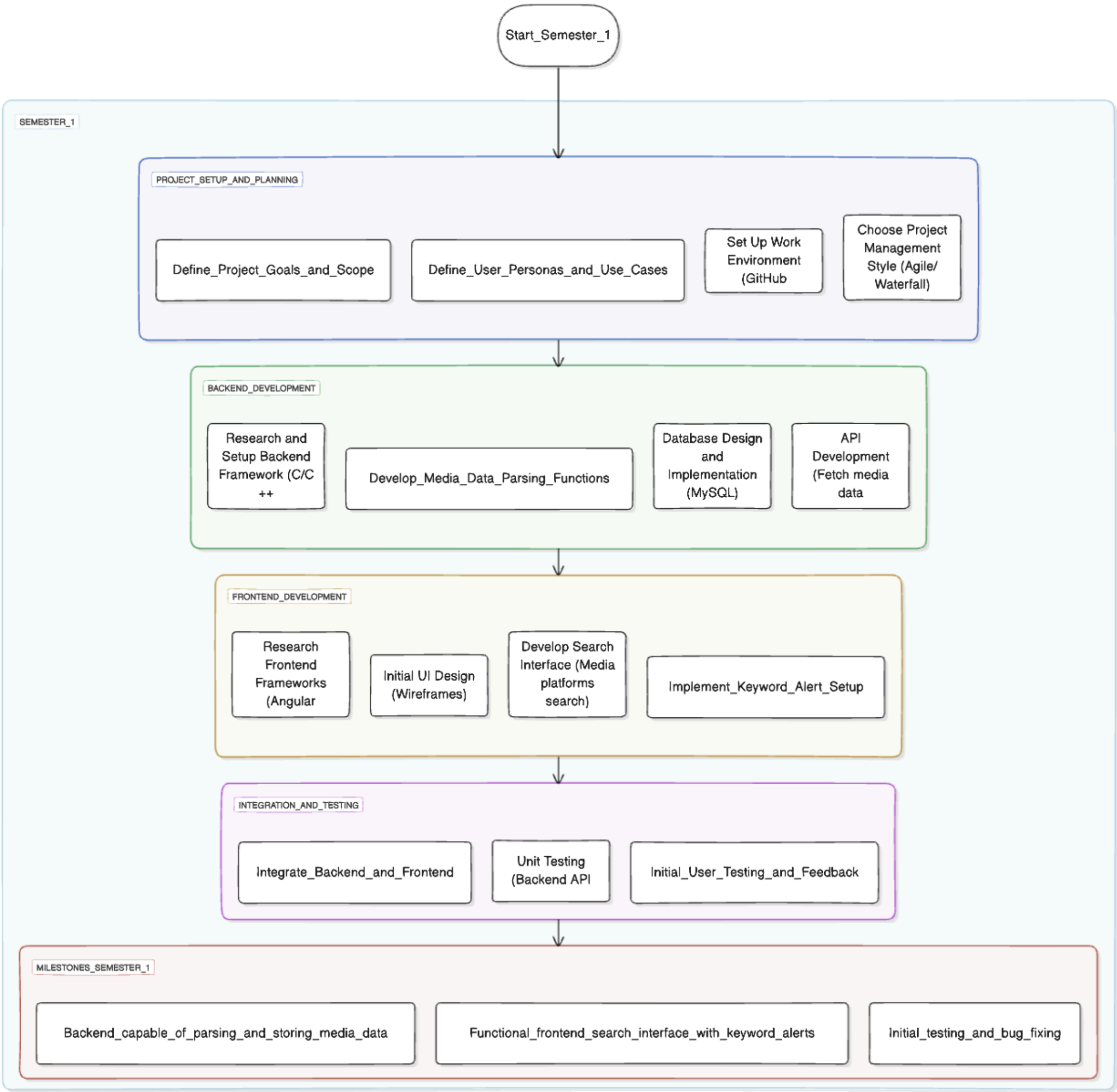
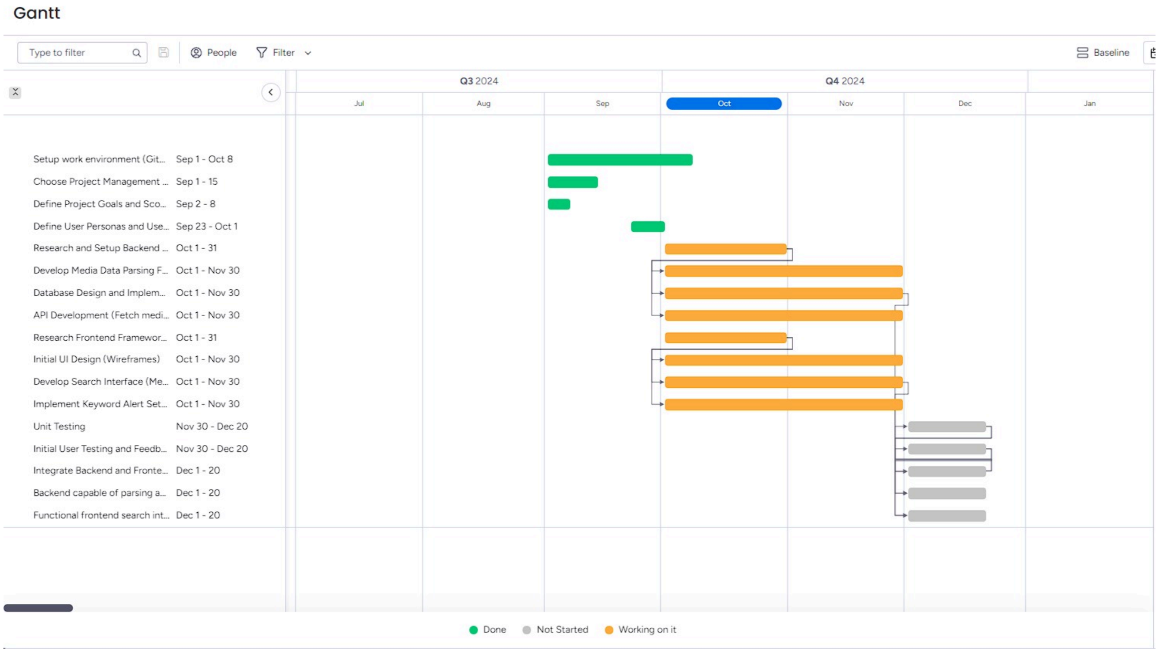

Start_Semester_1
SEMESTER_1
PROJECT_SETUP_AND_PLANNING
Define_Project_Goals_and_Scope
Define_User_Personas_and_Use_Cases
Set Up Work Environment (GitHub
Choose Project Management Style (Agile/ Waterfall)
BACKEND_DEVELOPMENT
Research and Setup Backend Framework (C/C ++
Develop_Media_Data_Parsing_Functions
Database Design and Implementation (MySQL)
API Development (Fetch media data
FRONTEND_DEVELOPMENT
Research Frontend Frameworks (Angular
Initial UI Design (Wireframes)
Develop Search Interface (Media platforms search)
Implement_Keyword_Alert_Setup
INTEGRATION_AND_TESTING
Integrate_Backend_and_Frontend
Unit Testing (Backend API
Initial_User_Testing_and_Feedback
MILESTONES_SEMESTER_1
Backend_capable_of_parsing_and_storing_media_data
Functional_frontend_search_interface_with_keyword_alerts
Initial_testing_and_bug_fixing

Figure 1. High-Level System Architecture



Setup work environment (Git...	Sep 1 - Oct 8
Choose Project Management ...	Sep 1 - 15
Define Project Goals and Sco...	Sep 2 - 8
Define User Personas and Use...	Sep 23 - Oct 1
Research and Setup Backend ...	Oct 1 - 31
Develop Media Data Parsing F...	Oct 1 - Nov 30
Database Design and Implem...	Oct 1 - Nov 30
API Development (Fetch medi...	Oct 1 - Nov 30
Research Frontend Frameworr...	Oct 1 - 31
Initial UI Design (Wireframes)	Oct 1 - Nov 30
Develop Search Interface (Me...	Oct 1 - Nov 30
Implement Keyword Alert Set...	Oct 1 - Nov 30
Unit Testing	Nov 30 - Dec 20
Initial User Testing and Feedb...	Nov 30 - Dec 20
Integrate Backend and Fronte...	Dec 1 - 20
Backend capable of parsing a...	Dec 1 - 20
Functional frontend search int...	Dec 1 - 20

Figure 2. Backend Request Processing Flowchart

Sprint 1 (Sep 1 - Oct 8):

- Setup Work Environment: Establish foundational tools and access to version control (Git) and project tracking systems.
- Choose Project Management Approach: Decide on an appropriate management style (e.g., Agile, Waterfall) based on project scope and goals.
- Define Project Goals and Scope: Identify clear project objectives and limitations for more focused research.

Sprint 2 (Sep 2 - Oct 1):

- Define User Personas and Use Cases: Conduct research to outline potential user types and their specific needs.
- Research and Setup Backend Requirements: Identify backend needs, including potential databases, frameworks, and system requirements.

Sprint 3 (Oct 1 - Nov 30):

- Develop Media Data Parsing Framework: Research methods and technologies for parsing media data effectively.
- Database Design and Implementation Plan: Outline database structures and schemas suitable for the data involved in the project.
- API Development Requirements: Investigate APIs needed to fetch media-related data and define integration strategies.

- Research Frontend Framework: Determine frontend technologies best suited for the project, focusing on UI requirements and compatibility.

Sprint 4 (Oct 1 - Nov 30):

- Initial UI Design (Wireframes): Create wireframes based on research findings for a user-centered design.
- Develop Search Interface Requirements: Research UI and UX considerations for search functionality tailored to user needs.
- Implement Keyword Alert Strategy: Define how keyword alerts could be managed based on the data availability and user preferences.

Sprint 5 (Nov 30 - Dec 20):

- Unit Testing Preparation: Develop a plan for unit tests based on the research into system requirements.
- Initial User Testing and Feedback Collection: Gather user feedback on wireframes and initial design concepts.
- Integrate Backend and Frontend Research: Begin preliminary integration planning between frontend and backend, informed by research outcomes.
- Finalize Backend and Frontend Compatibility: Ensure backend and frontend components are compatible based on gathered research insights.

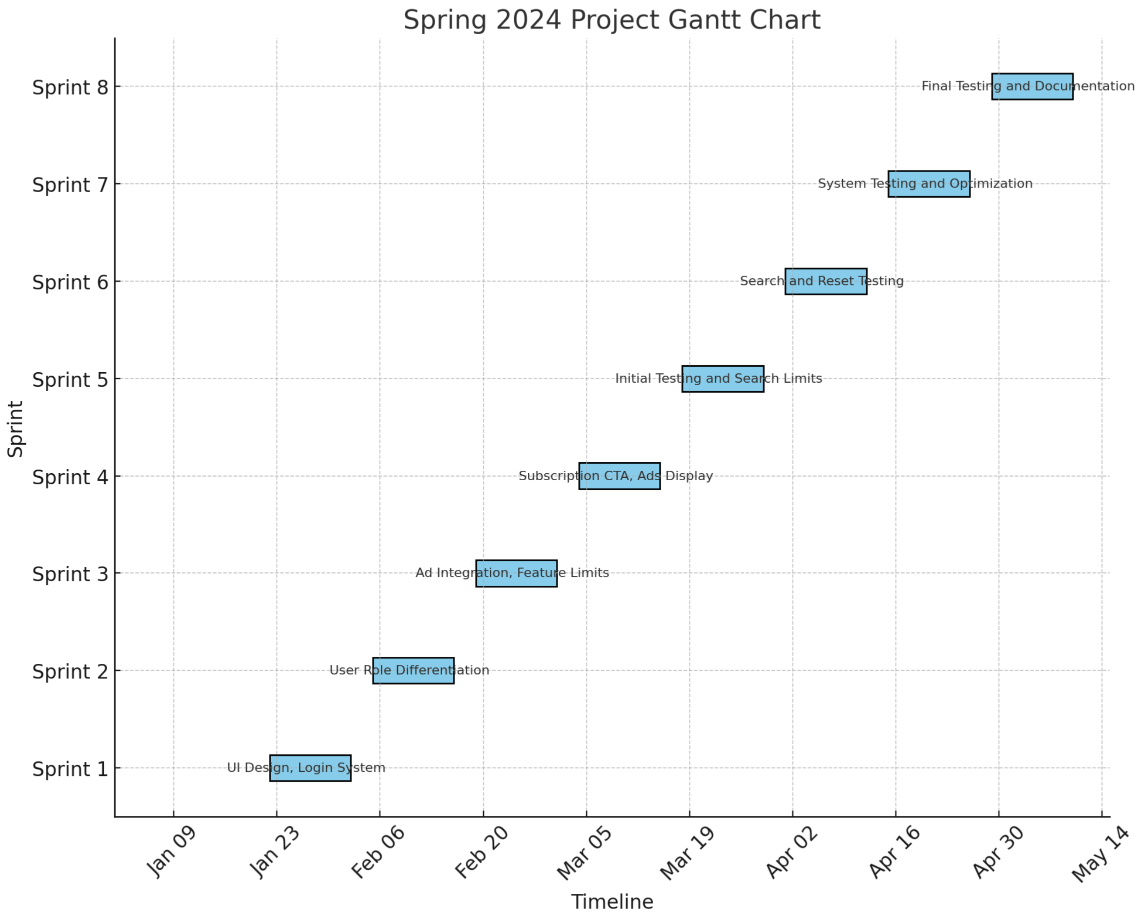


Figure 3. Timeline Visual

Sprint	Start Date	End Date	Tasks
Sprint 1	Jan 22, 2024	Feb 2, 2024	- UI Design for Different User Types - Develop the login and registration system for user types
Sprint 2	Feb 5, 2024	Feb 16, 2024	- Implement user role differentiation (Subscriber vs. Non-Subscriber) - User role assignment and role-checking backend logic

Sprint 3	Feb 19, 2024	Mar 1, 2024	- Implement advertisement integration based on user type - Enforce feature limitations for non-subscribers (e.g., search limitations, restricted menu items)
Sprint 4	Mar 4, 2024	Mar 15, 2024	- Create "Upgrade to Subscribe" CTA for non-subscribers - Integrate subscription upgrade functionality - Display ads and subscription prompts for non-subscribers
Sprint 5	Mar 18, 2024	Mar 29, 2024	- Initial testing for user type functionalities (Subscriber vs. Non-Subscriber) - Testing for search limitations and advertisement visibility
Sprint 6	Apr 1, 2024	Apr 12, 2024	- Test search feature and refine based on user feedback - Verify search limitations and reset functionality
Sprint 7	Apr 15, 2024	Apr 26, 2024	- System testing with complete feature integration - Review and optimize UI elements for user engagement
Sprint 8	Apr 29, 2024	May 10, 2024	- Final system testing

			and validation - Prepare documentation and wrap up for semester-end reviews

Table 7: Sprint Plan

3.5 Risks and Risk Management/Mitigation

This section identifies potential risks for each task in the DigiClips project, with a probability and severity assessment for each.

1. **Verifying users**

Task	Description	Risk	Probability	Severity	Mitigation Plan
1	Define user roles (Subscriber vs. Non-Subscriber)	Roles may not cover all necessary access cases	0.3	Medium	Conduct requirement analysis to ensure roles cover all use cases before implementation.
2	Implement user role assignment during registration	Incorrect role assignment may impact user access	0.4	Medium	Test registration workflows thoroughly to confirm roles are assigned correctly.
3	IBackend logic to check user role on login	Role validation may fail due to backend errors	0.5	High	Implement redundancy in role validation and use comprehensive error handling.

2. **Advertisement Integration**

Task	Description	Risk	Probability	Severity	Mitigation Plan
4	Research and select an ad provider	Ad provider may not align with system requirements or user preferences	0.3	Medium	Evaluate multiple ad providers based on system compatibility and privacy policies.

5	Integrate ads into the page layout	Ad integration could slow down page load times or affect UX	0.6	High	Use lightweight ad options, lazy loading, and test ad impact on load times. If performance remains unsatisfactory, explore alternative providers with lighter ads.
6	Implement ad visibility based on subscription	Complexity in setting visibility conditions may cause inconsistencies	0.4	Meidum	Run targeted testing with both subscribers and non-subscribers; consider standard frameworks for user-based settings to simplify.
7	Test ad performance and impact	Ads may impact user experience or system performance	0.7	High	Use a phased testing approach, measuring performance impact and adjusting ad frequency. Off-the-shelf tools like Google AdSense Performance Planner can simulate impact.

3. Search Limitations

Task	Description	Risk	Probability	Severity	Mitigation Plan
8	Define search count limit for non-subscribers	Limit may be insufficient or too restrictive for user needs	0.4	Medium	Conduct testing on different search limits to determine the optimal setting before launch.
9	Implement backend tracking for daily usage	Backend tracking may encounter logging errors or delays	0.5	Medium	Implement real-time logging verification and use reliable cloud-based logging solutions for scalability.
10	Create user notifications for search limit	Notifications may become intrusive for users	0.3	Low	Test notification intervals and offer options to mute or modify frequency of notifications for user convenience.
11	Test daily reset functionality for search counts	Reset functionality may fail due to time-zone inconsistencies or backend	0.6	High	Use a centralized server time setting to control resets. Perform manual and automated testing to confirm accuracy.

		errors			

4. Basic Media Search Functionality

Task	Description	Risk	Probability	Severity	Mitigation Plan
12	Develop basic search interface	UI/UX design may not meet user expectations	0.4	Medium	Conduct user testing with mockups to validate design choices before full development.
13	Enable search functionality	Search may not return relevant results due to media source inconsistencies	0.6	High	Test search with real-world data and adjust search algorithms as needed.
14	Test and refine search functionality	Initial feedback may require significant adjustments	0.7	High	Use iterative testing with multiple user feedback cycles to refine search. If performance issues persist, consider adding advanced caching mechanisms or switching to a third-party search API for improved efficiency.

5. Subscription Encouragement

Task	Description	Risk	Probability	Severity	Mitigation Plan
15	Design "Upgrade to Subscribe" call-to-action (CTA)	CTA may be ignored, reducing conversion rate	0.4	Medium	Use testing to optimize CTA design and positioning, focusing on user engagement insights.
16	Implement CTA visibility based on subscription	Subscription detection may be unreliable	0.3	Medium	Ensure accurate subscription status detection by integrating reliable user authentication.

17	Integrate subscription upgrade link	Links may not function due to backend or UI integration issues	0.5	Medium	Conduct end-to-end testing to ensure all links and subscription processes function correctly..
18	Test visibility of CTA and track click-through rates	Tracking may misreport engagement data due to browser limitations	0.6	High	Use multiple tracking tools for redundancy

These strategies ensure the project remains flexible and capable of adapting to unforeseen technical challenges while delivering a high-quality product to end users. Each sprint will evaluate and address risk factors, with continuous improvements based on user feedback and testing results.

3.6 Personnel Effort Requirements

Task	Estimated personal-hours	Explanation
Define user roles (Subscriber vs. Non-Subscriber)	8	Includes analyzing requirements and defining user roles in the database and codebase.
Implement role assignment during registration	12	Setting up backend processes for assigning roles upon user registration or login.
Backend logic to verify role on login	16	Coding and testing backend verification to confirm user roles on each login attempt.
Role-based feature enforcement	20	Enforcing role-based restrictions on features, ensuring access control.
Research and select ad provider	10	Research potential ad providers (e.g., Google Ads) and select the best fit for the system.
Integrate ads into page layout	15	Integrate chosen ads within UI while maintaining page design and responsiveness.

Implement ad visibility settings	12	Apply ad visibility settings based on user subscription level.
Test ad performance and impact	18	Monitor and test ad impact on performance, adjusting integration for optimal UX.
Define search limit for non-subscribers	8	Determine daily search limits for non-subscribers and integrate into backend logic.
Implement backend tracking for search usage	15	Set up backend counters to track search counts by user and reset daily.
User notifications for approaching search limit	10	Develop notification alerts for users when nearing the search limit.
Test daily reset functionality	12	Verify backend reset logic and ensure search count resets daily without issues.
Develop basic search interface (UI/UX)	20	Design and build the user interface for search, focusing on usability.
Implement media type filters	18	Add filtering options (e.g., TV, Radio) in the search function, enabling category selection.
Enable search for media sources	25	Connect search functionality to media sources, allowing full database query capability.
Test search functionality	15	Run extensive testing to ensure accuracy and reliability in search results and UI display.
Design “Upgrade to Subscribe” CTA	8	Design a clear call-to-action for non-subscribers to upgrade to a subscription.
Implement CTA visibility based on role	10	Make CTA visible to non-subscribers, incorporating subscription status into display logic.
Integrate subscription upgrade link	8	Link CTA to the subscription page, ensuring a seamless upgrade path for users.

Test CTA visibility and track click-through rates	12	Test CTA across various scenarios and add tracking to measure conversion rates.

Table 4: Personnel Effort Requirements

3.7 Other Resource Requirements

In addition to financial resources, the following are essential to complete the DigiClips project effectively, specifically incorporating GitHub repository, Amazon Lightsail Cloud Server, JavaScript and Angular 18, and an operating system to deploy the server:

- GitHub Repository: A centralized GitHub repository for collaborative development, version control, code reviews, and maintaining documentation. This repository will be crucial for implementing Continuous Integration (CI) and managing project versions effectively.
- Amazon Lightsail Cloud Server: A scalable and cost-effective cloud server environment to host the application, databases, and media storage. Amazon Lightsail will enable deployment and scaling of the application while offering secure and reliable server infrastructure.
- Operating System (OS) for Server Deployment: A stable and compatible operating system, such as Ubuntu Server or Amazon Linux, to host the server-side application and support JavaScript and Angular 18. This OS will provide a robust environment for deploying, maintaining, and scaling server-side processes.
- Database Management System: A database solution compatible with Amazon Lightsail, such as MySQL or PostgreSQL, to manage data for user roles, analytics, search logs, and media content efficiently.
- JavaScript and Angular 18: Primary development languages for building the front-end application, ensuring a modern, responsive, and user-friendly interface. Angular 18 will enable efficient handling of data binding, routing, and API integration.
- IDE and Development Tools: Integrated Development Environments (IDEs) like Visual Studio Code and IntelliJ for writing and debugging JavaScript and Angular 18 code, enhancing productivity and reducing errors.

4 Design

4.1 Design Context

4.1.1 Broader Context

By letting people watch and analyze media in real time, the DigiClips Media Search Engine project meets important wants in society. Professionals like lawyers, reporters, and media analysts can use it to get up-to-date, correct information from a variety of media sources that helps them make smart choices. The platform also looks at a number of social issues, such as the well-being of the public, the wider effect, the long-term viability of the economy, and the protection of the environment. The following gives a thorough breakdown of how DigiClips fits into this bigger picture:

Area	**Description**	**Examples**
Public health, safety, and welfare	Access to information and timely updates are affected, which is important for workers like lawyers and reporters who need to make decisions.	Reducing misinformation by providing accurate, real-time updates. Protecting people by making sure that legal and media workers get alerts at the right time.
Global, cultural, and social	Real-time media updates help professionals from all over the world and countries with their needs.	Helping reporters cover international stories efficiently. Allowing media analysts to study cultural trends through global media coverage.
Environmental	Minimal environmental impact since the project operates digitally without significant physical resource consumption.	Implementing digital reports to mitigate paper pollution. Enabling remote media monitoring to reduce the carbon footprint.
Economic	Provides a quick and flexible media search tool that helps businesses cut costs and increase output.	Giving people affordable access to important media information. Opening up new possibilities in data analysis and monitoring the media.

Table 8: Broader Context Analysis of DigiClips Media Search Engine

4.1.2 Prior Work/Solutions

Revised Literature Review and Background

Existing Media Search Engines

Several media search engines and monitoring tools offer a variety of capabilities that range from basic search functionalities to complex analytics, including:

1. **Hootsuite Insights**:
   - **Capabilities**: Offers real-time social media monitoring and analytics.
   - **Reference**: [1] Hootsuite, "Advanced Social Media Analytics," Hootsuite Media Inc., Vancouver, BC, Canada, 2022.
2. **Brandwatch**:
   - **Capabilities**: Provides consumer research, image recognition, and trend tracking over social platforms.
   - **Reference**: [2] Brandwatch, "Brandwatch Consumer Research," Brandwatch GmbH, Brighton, UK, 2022.
3. **Social Searcher**:
   - **Capabilities**: Allows users to search for content across multiple social media platforms without logging in, offering real-time search results and analytics.
   - **Reference**: [4] Social Searcher, "Social Media Search Engine," Social Searcher, accessed December 7, 2024, https://www.social-searcher.com/.

Conclusion and Future Directions

The DigiClips Media Search Engine seeks to democratize media monitoring by providing a tool that is both accessible and powerful, filling the niche for users needing affordable and straightforward media tracking solutions. Future developments can focus on enhancing AI-driven analytics and expanding integration capabilities to include predictive analytics based on media trends.

Advantages and Shortcomings of Existing Systems

These existing tools, while robust, come with certain limitations which DigiClips aims to address:

- **Cost**: Tools like Brandwatch, Meltwater, and even Social Searcher offer premium features at a cost, which can be prohibitive for small organizations and individuals.
- **Complexity**: Systems may offer complex features that exceed the needs of casual users or small enterprises.
- **Integration**: Limited integration capabilities with proprietary or niche platforms, an area where DigiClips seeks to excel.

DigiClips Media Search Engine Differentiation

Unique Selling Points

The DigiClips Media Search Engine is uniquely positioned to address the gaps identified in existing products:

- **Affordability**: Unlike high-cost alternatives, DigiClips offers a free tier making it accessible to individuals and small businesses.
- **Simplicity and Usability**: Prioritizes user-friendly interfaces and simplifies common tasks to accommodate users with limited technical expertise.
- **Extensive Integration**: Supports a broader range of media platforms, including emerging and niche channels not covered by most mainstream tools.

References

[1] Hootsuite, "Advanced Social Media Analytics," Hootsuite Media Inc., Vancouver, BC, Canada, 2022.

[2] Brandwatch, "Brandwatch Consumer Research," Brandwatch GmbH, Brighton, UK, 2022.

[3] Social Searcher, "Social Media Search Engine," Social Searcher, accessed December 7, 2024, https://www.social-searcher.com/.

4.1.3 Technical Complexity

The DigiClips Media Search Engine project is technically complex due to the integration of multiple components and subsystems, each utilizing distinct scientific, mathematical, or engineering principles. This complexity is further amplified by the need to meet challenging requirements that align with or exceed current industry standards. Below is a breakdown of the technical complexity:

1. Multiple Components and Subsystems
    - Frontend Development (Angular Framework):
        - The frontend involves the development of an intuitive user interface using Angular 18.
        - Scientific and engineering principles such as state management, responsive design, and dynamic data binding are applied to ensure usability and scalability.
    - Backend Processing (Node.js with API Integration):
        - The backend system handles search requests, processes data, and manages communications with the media database.
        - This subsystem utilizes algorithms for query parsing, JSON Web Token (JWT) authentication for secure data exchange, and real-time data handling principles.
    - Media Database:
        - The database stores large-scale multimedia data, applying database normalization, indexing, and query optimization to ensure efficient storage and retrieval.

    - It implements relational database principles (MySQL) to manage structured data and support advanced query features.
- Notification System:
    - This subsystem sends alerts based on user-defined keywords and preferences (e.g., real-time, daily, or weekly).
    - Scientific principles include event-driven architecture and scheduling algorithms for timely delivery.
- Authentication and Security:
    - Authentication involves JWT and session management to protect user data.
    - Security principles such as encryption, access control, and vulnerability audits are integrated to ensure a robust and secure system.

2. Challenging Requirements

- Search Limitation for General Public Users:
    - The system implements a daily search cap with real-time tracking of search usage, ensuring compliance with subscription incentives while maintaining performance.
- Ad Integration:
    - Ads are strategically placed using computational models to balance user experience and revenue generation.
- Real-Time Analytics and Reporting:
    - Subscribers gain access to real-time analytics, requiring high computational efficiency to generate reports dynamically.
- Cross-Platform Compatibility:
    - The design ensures the platform works seamlessly across devices (desktop and mobile), leveraging responsive design principles.

3. Justification

The DigiClips Media Search Engine project addresses a problem scope that matches or exceeds current solutions in the industry:

- The integration of secure backend processing with a scalable media database reflects industry standards.
- Implementing a hybrid subscription model with ad integration and search limitations demonstrates innovation in balancing user experience and revenue.
- Real-time notifications and analytics ensure competitive parity with existing media search tools.

4.2 Design Exploration

4.2.1 Design Decisions

Design Decisions #1: Implementing an Ad-Supported Interface for General Public Users

- Description: Since non-subscribers don't have access to premium features, the interface will display ads to financially support the service. This decision aligns with the need for revenue generation without requiring general users to pay.
- Importance: This ensures the project remains financially sustainable and accessible to non-subscribers. Integrating ads allows non-subscribers to continue accessing basic media search functionalities, even without a paid subscription.

Design Decisions #2: Establishing a User Role System in the MySQL Database

- Description: The MySQL database will incorporate a user role identification system to differentiate between general public users and subscribers. This design is crucial to enforce access restrictions and customized experiences.
- Importance: This database layer enforces unique access levels, such as limiting search frequency for non-subscribers and providing premium features for subscribers. Differentiating user types ensures security, streamlined data handling, and user-specific interactions, enhancing the overall functionality and relevance of the search engine.

Design Decisions #3: Implementing a Limited Search Feature for Non-Subscribers

- Description: The backend will impose a maximum of five searches per day for general public users, ensuring system resources are reserved for subscribers while still offering a limited service to non-subscribers.
- Importance: This search limitation manages the backend load effectively while encouraging general users to subscribe for unrestricted access. Limiting daily searches also maintains a fair use policy, reducing the chance of system overload and maintaining performance for all users.

4.2.2 Ideation

Exploring Options for Design Decision #3 (Implementing a Limited Search Feature for Non-Subscribers)

To determine the best way to implement a search limitation for non-subscribers, we utilized a brainstorming technique inspired by the Lotus Blossom method. Below are five potential options we considered:

Daily Search Cap (Chosen Option)

Description: Limit non-subscribers to a maximum of five searches per day.

- **Pros:**
  - Simple to implement.
  - Encourages subscriptions.
  - Manageable backend load.
- **Cons:**
  - Users may feel restricted.

    - Risk of users seeking alternative platforms.

Search Token System

Description: Provide non-subscribers with a set number of tokens per week, with each search consuming one token.

- **Pros:**
    - Offers flexibility in search usage.
    - Encourages thoughtful search behavior.
- **Cons:**
    - Complex token management system.
    - Users might find the token system confusing.

Time-Based Search Limit

Description: Allow non-subscribers to search freely but impose a cooldown period (e.g., 1 hour) between searches.

- **Pros:**
    - Reduces immediate frustration by spreading searches over time.
    - Provides a perception of continued usage.
- **Cons:**
    - Potentially increased backend load during off-peak hours.
    - Encourages sporadic and unpredictable usage patterns.

Feature-Limited Search

Description: Allow unlimited searches but restrict the depth of results (e.g., only top 3 results per query).

- **Pros:**
    - Users can explore without hitting hard caps.
    - Less disruptive user experience.
- **Cons:**
    - May reduce incentive to subscribe.
    - Frustration from incomplete search results.

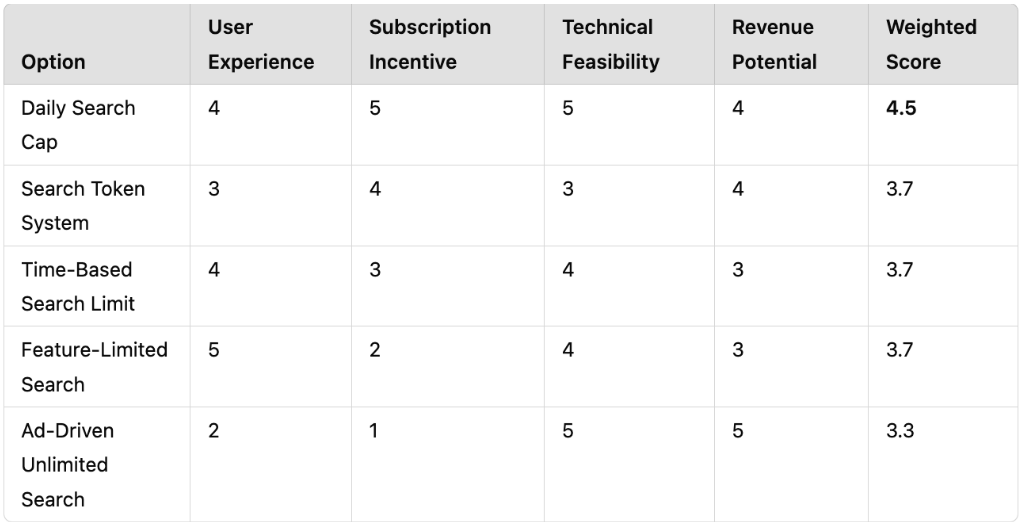
Ad-Driven Unlimited Search

Description: Offer unlimited searches, but with heavy ad placement on each search result page.

- **Pros:**
    - Maximizes ad revenue while maintaining unlimited functionality.
    - Provides an alternative revenue stream.
- **Cons:**
    - High risk of user frustration due to intrusive ads.
    - Performance issues from ad-heavy pages.

4.2.3 Decision-Making and Trade-Off

To evaluate the trade-offs between the five options, we developed a weighted decision matrix with the following criteria: **User Experience (30%)**, **Subscription Incentive (30%)**, **Technical Feasibility (20%)**, and **Revenue Potential (20%)**. Each option was scored on a scale of 1 to 5.



Option	User Experience	Subscription Incentive	Technical Feasibility	Revenue Potential	Weighted Score
Daily Search Cap	4	5	5	4	**4.5**
Search Token System	3	4	3	4	3.7
Time-Based Search Limit	4	3	4	3	3.7
Feature-Limited Search	5	2	4	3	3.7
Ad-Driven Unlimited Search	2	1	5	5	3.3

Table 1: Decision Matrix for Design Options

Chosen Option: Daily Search Cap

The **Daily Search Cap** scored the highest overall, as it provides a good balance between encouraging users to subscribe and maintaining a manageable backend load. It offers a clear, predictable user experience without overwhelming users with complexity. Additionally, it ensures consistent ad revenue from general users while maintaining strong performance for subscribers.

Justification for Choice

The simplicity of the Daily Search Cap aligns well with user expectations and supports the system’s financial sustainability. It strikes a balance between providing value to non-subscribers and incentivizing them to upgrade, ensuring long-term platform viability. Furthermore, the clear limitation on searches minimizes strain on backend resources, allowing for stable performance as the user base scales.

4.3 Proposed Design

4.3.1 Overview

The DigiClips Media Search Engine is a platform built to help users keep track of real-time updates from various media sources, including TV, radio, social media, blogs, and podcasts. It gathers and organizes this information, allowing users to search for specific topics, set up notifications, and view relevant content in one place.

Key Parts of the System

User Interface: This is the main screen users see and interact with. It lets them search for media content, apply filters, view reports, and receive alerts on the topics they care about, all in a clear, easy-to-use layout.

Backend Processing: This is the "brain" of the platform, managing data requests from the User Interface. It processes searches and scans media content for relevant information, handling all the technical work in the background.

Media Database: This is the storage area for all collected media content, user information, and search history. It keeps data organized so that the system can quickly search and retrieve content when users make a request or check previous alerts.

Notification System: This part sends alerts to users when something important or relevant appears in the media. Users can choose to receive these notifications instantly or on a daily or weekly schedule.

Authentication & Security: This part ensures that only authorized users can access the system and protects all data transfers. It manages logins and user roles, keeping the platform secure.

How It All Works Together

When a user searches for specific content or sets up alerts, the User Interface sends the request to the Backend Processing system. The backend then retrieves the relevant data from the Media Database and checks if any alerts match the user’s preferences. If so, the Notification System sends a real-time or scheduled alert to the user. Throughout, Authentication & Security protects user data and access.

In short, these parts work together to create an easy, secure, and efficient way for users to stay updated on the media content that matters most to them.

4.3.2 Detailed Design and Visual(s)

The following visuals have been prepared to provide a structured and comprehensive understanding of the DigiClips Media Search Engine architecture. These diagrams illustrate the system's core components, data flows, and key operational processes, presenting each subsystem's functionality and its integration within the platform.

- High-Level System Block Diagram: This diagram presents a top-level view of the DigiClips Media Search Engine, showing how each subsystem—including the Frontend Interface, Backend Processing, Media Database, Notification System, and Authentication & Security—contributes to the platform's end-to-end functionality.

- Backend Request Processing Flowchart: This flowchart details the lifecycle of a user-initiated search request, covering steps from initial submission through data retrieval, processing, and response. It also demonstrates how alerts are triggered and notifications are sent based on keyword matches, illustrating the backend's role in secure, efficient request handling.

- Notification Flowchart: This flowchart outlines the workflow for detecting, formatting, and delivering alerts to users. It captures each step in the notification process, from detecting keywords in media content to delivering user-specific alerts based on preferences, ensuring timely and relevant notifications.

High-Level System Block Diagram

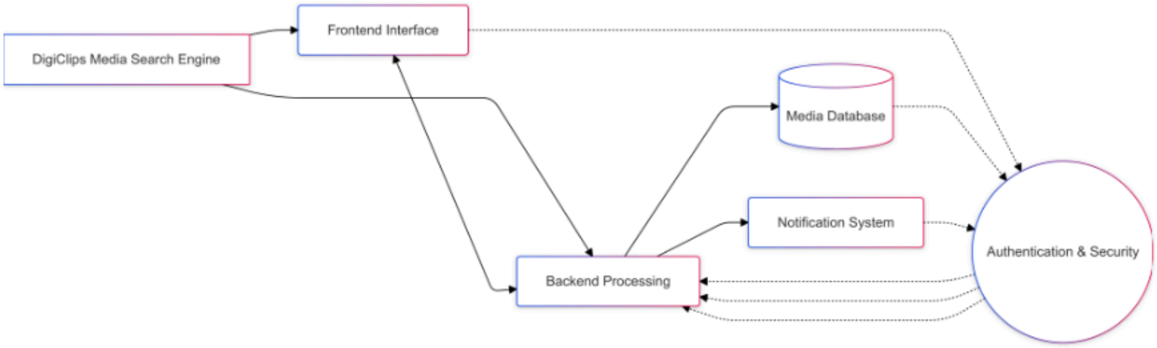


Figure 6: High-Level System Block Diagram

This diagram provides a broad overview of the key subsystems and their interconnections in the DigiClips Media Search Engine.

DigiClips Media Search Engine (Main System): The central node representing the entire media search engine platform.

Frontend Interface: The user-facing component built with Angular, responsible for search, media playback, report generation, and alert display. It interacts with the backend to retrieve and display data, as well as receive notifications from the alert system.

Backend Processing: This core processing unit is implemented in Node.js and Express, handling data requests, processing media content, and managing search and alert algorithms. It serves as the hub, connecting with the frontend, media database, and notification system.

Media Database: A MySQL database that stores media content, user profiles, search history, and metadata, managed via Docker for scalability.

Notification System: Sends alerts to users based on keyword matches or other trigger events in monitored content. Supports email and push notifications and is integrated with the backend for real-time alerts.

Authentication & Security: This layer secures data exchange across components, using JWT for authentication and HTTPS for encrypted data transfer, ensuring only authorized users can access and interact with sensitive data.

This diagram illustrates how each subsystem communicates, with API calls between frontend and backend, database queries from the backend to the Media Database, and triggers sent from the backend to the Notification System.

Backend Request Processing Flowchart

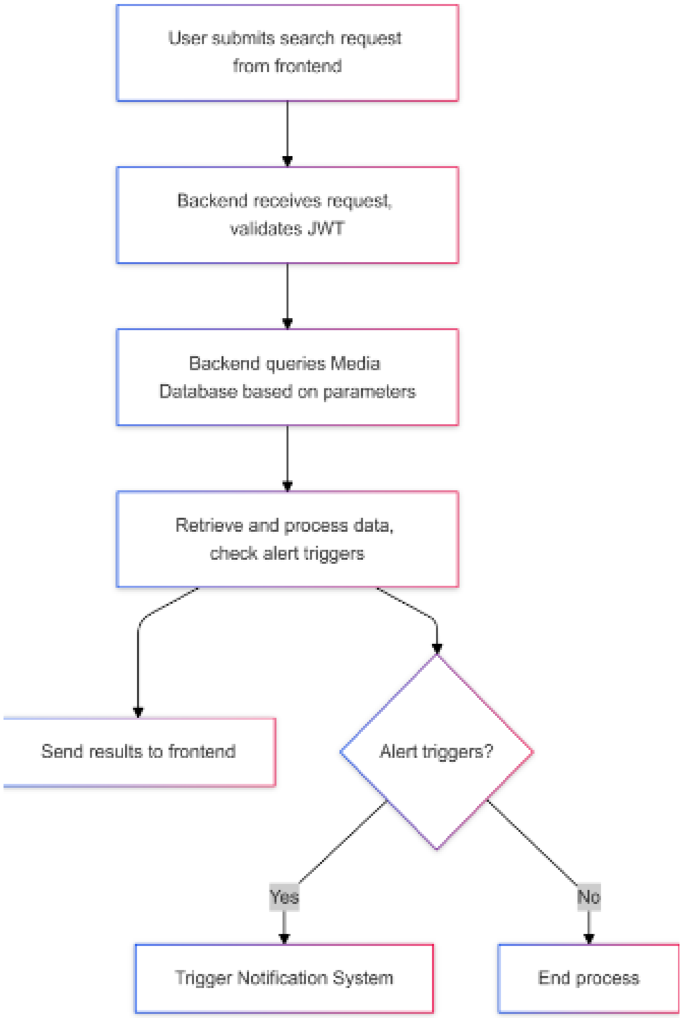

User submits search request from frontend
Backend receives request, validates JWT
Backend queries Media Database based on parameters
Retrieve and process data, check alert triggers
Send results to frontend
Alert triggers?
Yes
No
Trigger Notification System
End process

Figure 7: Backend Request Processing Flowchart

This flowchart demonstrates the backend's request handling process from the time a user submits a search request to delivering results and handling potential alerts.

Step 1: A user submits a search request through the frontend interface.

Step 2: The backend receives the request and verifies the user's JWT token to authenticate the request.

Step 3: Once authenticated, the backend queries the Media Database with the specified search parameters.

Step 4: Data retrieved from the database is processed and checked for any keyword-based alert triggers.

Step 5: The backend sends the processed results back to the frontend for display.

Step 6: If there are any alert triggers (e.g., specific keywords detected), an alert is sent to the Notification System.

End Process: Once the results are sent, the system completes the request handling process.

Notification Flowchart

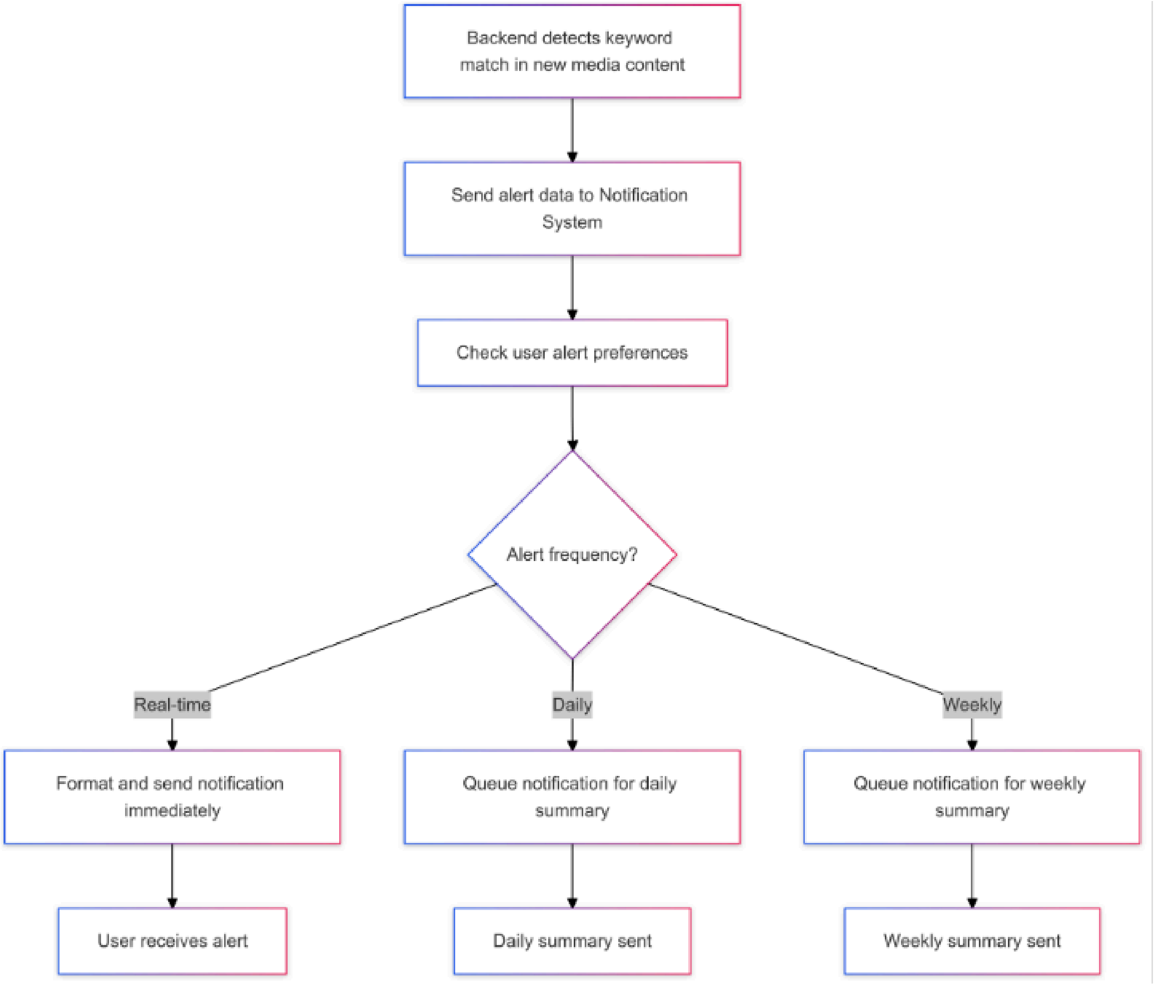

Backend detects keyword match in new media content
Send alert data to Notification System
Check user alert preferences
Alert frequency?
Real-time
Daily
Weekly
Format and send notification immediately
Queue notification for daily summary
Queue notification for weekly summary
User receives alert
Daily summary sent
Weekly summary sent

Figure 8. Notification Flowchart

This flowchart explains the process for detecting and sending alerts through the Notification System.

Step 1: When the backend detects a keyword match in newly ingested media content, it initiates an alert process.

Step 2: The alert data (such as user ID, media link, and timestamp) is sent to the Notification System.

Step 3: The system checks the user's alert preferences (e.g., real-time, daily, or weekly alerts).

Step 4: Based on the user's preferred alert frequency:

Real-time alerts are formatted and sent immediately.

Daily alerts are queued for a summary at the end of the day.

Weekly alerts are queued for a summary at the end of the week.

End Steps: Notifications are then delivered to the user based on the selected frequency.

4.3.3 Functionality

The General Public page is designed to give registered users access to a range of media data and reporting tools, enabling them to search and generate insightful media reports for various platforms such as Television, Radio, Newspapers, Magazines, Social Media, Blogs, Web TV, Podcasts, and SiriusXM Radio. Once a user registers with their email, they can perform up to five searches per day and receive basic reports on the selected media. However, the system will display advertisements and restrict access to premium features, such as detailed analytics and email notifications, which are only available to subscribers. To encourage users to subscribe, the interface includes a prominent call-to-action button, urging them to unlock more advanced features. As the user interacts with the platform, the system responds by limiting search functionality, showing ads, and highlighting subscription benefits to drive conversion.

Action Flows:

- Landing Page (Registration)

User Action: A user lands on the General Public page and sees a clear option to register with an email.
System Response: Upon registration, the user is granted limited access to search and generate basic reports.

- Dashboard (After Login)

User Action: After logging in, the user is presented with a dashboard that allows them to enter search queries for media platforms like TV, Radio, Social Media, etc.
System Response: The dashboard displays a search bar and a counter for remaining searches (e.g., "5 searches remaining today").
Display the search bar with category selection (TV, Radio, Social Media, etc.)
Display daily search limit counter: "5/5 Searches Remaining"
Display Ad banners in sidebar or bottom of the screen
Display a small pop-up: "Unlock detailed analytics with a subscription!"

- Search Results (Media Data)

User Action: The user enters a search query (e.g., "TV News October 2024") and views a basic report.
System Response: The system provides the search results, displaying relevant media data along with ads. Premium features are hidden.
List of search results with basic details (titles, descriptions)
Prominent banner/button: "Upgrade for detailed insights!"
Ads displayed alongside content
Limitations highlighted: "You can only view 5 results for this search. Subscribe for more."

- Call-to-Action & Subscription Prompt

User Action: The user sees a prominent button or pop-up encouraging them to subscribe for additional features (e.g., detailed analytics, email notifications).
System Response: The page responds by showing more enticing subscription options.
Large "Subscribe Now" button
Text: "Unlock unlimited searches and detailed analytics."
A small pop-up or banner: "Sign up for only $X/month to remove ads and unlock premium features!"

- Conclusion / Reminder of Limitations

User Action: If the user tries to search again, the system reminds them of the limit (if they've reached their 5 searches).
System Response: Display a message: "You've reached your limit for today. Subscribe for unlimited searches."
Greyed-out search button
Clear message stating "5/5 searches used today."

4.3.4 Areas of Concern and Development

Meeting Requirements and User Needs:

The current design satisfies user needs by providing access to basic media search functions and generating reports for various platforms (e.g., TV, radio, social media). Additionally, the role-based system meets client requirements by differentiating access levels between general users and subscribers. The integration of ad-supported interfaces for non-subscribers offers financial sustainability, while the subscription model incentivizes users to upgrade for enhanced features like detailed analytics.

Primary Concerns:

Scalability: As user numbers increase, ensuring the backend can handle a high volume of simultaneous search requests is crucial. There's a risk of performance slowdowns if the system is unable to scale effectively, especially during peak usage.

User Experience with Ads: Ad placement for non-subscribers may impact user experience, potentially leading to lower engagement. Striking a balance between ad visibility and usability will be essential to retain non-subscribers.

Data Accuracy and Relevance: Ensuring the search functionality provides accurate, up-to-date results across various media platforms is another concern, especially given the diversity of sources and real-time updates required.

Immediate Development Plans:

Optimize System Scalability: Explore load-balancing solutions and conduct stress testing to prepare for high-traffic scenarios. Implement modular upgrades to handle increased data processing demands.

Refine Ad Placement: Experiment with ad positions and formats that are less intrusive but still visible. Collect user feedback on ad experience to further refine placement.

Improve Search Relevance: Refine the backend search algorithms and explore possible integration with natural language processing (NLP) to enhance keyword matching accuracy.

Questions for Clients, TAs, and Faculty Advisors:

For Clients: Are there specific guidelines or preferences for ad placements that balance user experience and revenue generation?

For TAs and Advisors: What are some best practices for handling scalability in media search engines, especially in peak traffic conditions?

General: Are there recommended tools or frameworks for tracking user engagement with ads to measure potential impacts on user retention?

4.4 Technology Considerations

DigiClips uses a combination of various technologies to deliver real-time news monitoring services to its customers, integrating a sophisticated backend with a custom search engine and ad integration. Below is a breakdown of the key technologies used in the system, along with an evaluation of their strengths, weaknesses, and trade-offs.

MySQL Database (Data Storage)

Strength: Relational Data Management: MySQL is a widely-used, reliable relational database management system that excels at storing structured data. It’s ideal for organizing and managing large volumes of media metadata (e.g., news articles, broadcasts, etc.).

Weaknesses: While MySQL performs well for moderate data sizes, scaling horizontally (across multiple servers) can be complex. This can become a limitation when handling millions of media records and analytics data.

Trade-Offs: For structured media data (e.g., metadata about radio/TV shows, news articles), MySQL is a strong choice. However, if the data becomes too unstructured or varied, integrating a NoSQL database (like MongoDB) could provide more flexibility in managing diverse media content.

Frontend (Angular)

Strengths: Angular: Angular is a powerful JavaScript framework for building single-page applications. It enables two-way data binding, modular component-based architecture, and integrates seamlessly with RESTful APIs (e.g., for fetching news data).

Weaknesses: Angular has a steeper learning curve compared to other frontend frameworks like React or Vue.js. This could increase development time and complexity, particularly for teams with less experience in Angular. Additionally, Angular evolves quickly, with frequent major version updates that can introduce breaking changes. Older versions of Angular can become outdated and vulnerable, with dependencies that may no longer be supported or secure.

Trade-Offs: Angular provides a full-featured framework that handles routing, state management, and UI components out of the box. However, this can make the application heavier and potentially slower for smaller, simpler projects.

Backend (Node.js)

Strengths: Since Node.js uses JavaScript on both the server and client sides, it simplifies development, particularly for teams that are already familiar with JavaScript. This allows for a consistent coding environment across the entire stack. Node.js is asynchronous and event-driven, which makes it ideal for handling many simultaneous requests (such as real-time news updates from multiple sources) without blocking operations.

Weaknesses: While Node.js is fast for I/O-bound operations, it is single-threaded, meaning CPU-bound tasks can be a bottleneck. Due to its asynchronous nature, handling complex operations with nested callbacks can result in convoluted code. This can be mitigated with promises or async/await, but it remains a consideration.

Trade-Offs: Node.js is highly efficient for handling large numbers of I/O operations, but might not be the best fit for CPU-heavy operations (e.g., complex data processing or image/video analysis).

Ad Integration (Third-Party Ad Networks)

Strengths: Integrating advertisements helps monetize the platform, making it possible to offer free services while sustaining operational costs.

Weaknesses: Frequent or intrusive ads can degrade the user experience, potentially frustrating customers, especially in a platform used for real-time information like news monitoring. The balance between maximizing ad revenue and providing a seamless user experience can be tricky. Too many ads or overly disruptive formats (like pop-ups) can lead to bad user experience.

Trade-Offs: Ads can provide a necessary revenue stream, but excessive ads can hinder usability. It's a challenge to find the right balance between monetization and user satisfaction.

4.5 Design Analysis

Progress So Far:

We have implemented the foundational components of the DigiClips Media Search Engine, including the ad-supported interface for general public users, a user role system in the MySQL database to manage subscriber and non-subscriber access, and search limitations for non-subscribers. The team has successfully integrated Angular for the frontend interface, which allows users to search and view media content, with backend processing in Node.js handling requests and database interactions.

Evaluation of Proposed Design:

The initial design has largely been effective. The ad-supported model provides a consistent revenue stream, allowing non-subscribers to access basic functionality while encouraging upgrades. The user role system has proven essential for maintaining clear distinctions between access levels, aligning with client requirements for a subscription model. The search limitations feature helps manage backend loads while incentivizing subscriptions, and initial testing has shown this approach to be both effective and user-friendly.

Challenges Encountered:

Scalability: While the basic search functionality is in place, testing under higher loads highlighted the need for further scalability improvements. As the user base grows, handling simultaneous searches could lead to performance issues.

User Experience with Ads: Finding the right balance for ad placements has been challenging. The current setup risks affecting the user experience, as too many ads may reduce engagement, especially for non-subscribers who depend on ad-supported access.

Data Accuracy: Ensuring search results are accurate and relevant across various media sources has proven complex due to the diversity of data formats and sources. Further tuning of the search algorithms is required.

Future Plans:

Scalability Enhancements: Implement load balancing and explore distributed databases to support a growing number of users while maintaining performance.

Refine Ad Placement Strategy: Test less intrusive ad placements and collect user feedback to optimize ad visibility without compromising user experience.

Improve Search Algorithms: Experiment with natural language processing (NLP) and machine learning techniques to enhance keyword relevance and data accuracy, ensuring a better user experience.

Feasibility and Overall Design Implications:

The project's foundational elements are feasible and effective; however, scalability and ad integration require additional work to ensure the platform's long-term sustainability. These improvements are manageable within the project scope and will significantly enhance the product's alignment with client and user needs.

5 Testing

5.1 Overview of Testing Strategy

The testing strategy for the DigiClips Media Search Engine is designed to ensure that the system is robust, secure, user-friendly, and meets all specified requirements. The strategy includes various stages of testing from unit testing at the lowest level of code to acceptance testing with real users. Our philosophy is to test early and often, integrating testing throughout the development process to catch issues as early as possible, which is critical given the system's reliance on real-time data and user interaction. Unique challenges include ensuring the accuracy and timeliness of search results across various media types and managing large datasets securely.

5.2 Unit Testing

- Units Being Tested: Core components such as the search algorithm, user authentication, data retrieval, and advertisement management.
- Methodology: Each unit will be tested in isolation using a range of inputs, including edge cases.
- Tools: Jest for JavaScript testing to validate business logic and server-side components.

5.3 Interface Testing

- Interfaces Tested: The interaction between the front-end application and the back-end server, the API endpoints, and the database.
- Methodology: Testing will ensure that these interfaces correctly handle data exchanges and function calls under different scenarios.
- Tools: Postman for API testing and Selenium for front-end interaction testing.

5.4 Integration Testing

- Critical Paths: Integration of the user authentication system with search functionality, ensuring user queries return the correct media results.
- Justification: Critical for maintaining a seamless user experience and data accuracy.
- Methodology: Simulate complete workflows to verify components work together as expected.
- Tools: Jenkins for orchestrating a continuous integration pipeline that automatically runs integration tests.

5.5 System Testing

- Strategy: System testing will assess the complete, integrated system's performance and stability.
- Methodology: Conduct tests that mimic real-world usage to see if the system meets the technical specifications and user expectations.
- Tools: Load testing with JMeter to evaluate performance and scalability.

5.6 Regression Testing

- Purpose: Ensure new updates do not negatively affect existing functionality.
- Methodology: After updates, re-run previous test cases to check for any failures.

- Tools: Automated test scripts in Selenium to cover major functionalities.

5.7 Acceptance Testing

- Objective: Verify the system meets both functional and non-functional requirements from the stakeholders' perspective.
- Involvement: Clients and select real users will be involved in testing the system.
- Methodology: Conduct user acceptance tests with scenarios that cover typical use cases.
- Tools: Direct observation, user feedback forms, and usability testing tools.

5.8 Security Testing

- Applicability: Critical due to the handling of sensitive user data.
- Methodology: Perform security audits, vulnerability scanning, and penetration testing.
- Tools: OWASP ZAP for security vulnerabilities identification and remediation guidance.

Results

- Findings: Initial tests have demonstrated high compliance with functional requirements, with search functionalities returning accurate results within acceptable time frames. Usability tests have highlighted some areas for improvement in navigation and user interface design.
- Data Presentation: Results include performance metrics, error rates, and user satisfaction ratings, displayed in graphical format for clear visualization.
- Compliance and User Needs: The system has so far met most of the functional requirements. However, user feedback has led to a plan for iterative improvements in UI/UX.
- Next Steps: Address the feedback from usability tests, refine the interface, and increase security measures. Additional rounds of testing will be scheduled to ensure that all issues are resolved before final deployment.

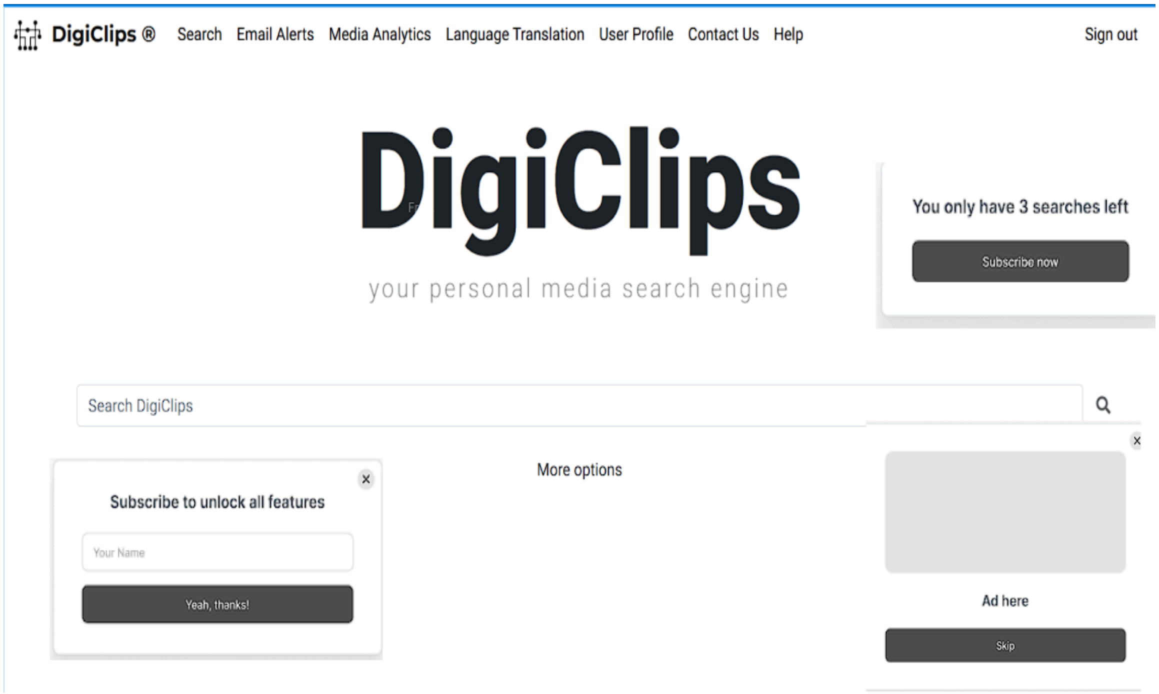
This detailed testing plan aims to thoroughly evaluate every aspect of the DigiClips Media Search Engine, ensuring that it not only meets the technical requirements but also delivers a seamless and secure user experience.

6 Implementation

We have developed a prototype for the General Public page after user login, featuring key functionalities such as:

- A search bar.
- A popup displaying the remaining search quota for the user (limited to 5 per day).
- An ad display in the bottom-right corner of the screen.
- A call-to-action button in the bottom-left corner of the screen to encourage users to subscribe.

This prototype has been discussed with DiGiClip and approved by DiGiClip.



You only have 3 searches left

Subscribe to unlock all features

Figure 4: Prototype Design

7 Ethics and Professional Responsibility

This discussion is with respect to the paper by J. McCormack and colleagues titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Professional Responsibility/Codes of Ethics

Area of Responsibility	Definition	Relevant Item from Code of Ethics	Team's Interaction with Responsibility
Work Competence	Perform work of high quality and integrity.	IEEE: Undertake technological tasks only if qualified or with full disclosure.	Regularly reviewing project requirements and ensuring the team meets deadlines with quality deliverables.
Financial Responsibility	Deliver products of value at reasonable costs.	IEEE: Seek and acknowledge honest feedback and correct errors.	Using open-source and free tools to ensure a cost-efficient project while delivering a high-quality product.
Communication Honesty	Report work truthfully and understandably.	IEEE: Be honest and realistic in claims or estimates based on available data.	Weekly meetings and transparent communication with the client and advisor to align on deliverables.
Health, Safety, and Well-Being	Minimize risks to safety, health, and welfare.	IEEE: Accept responsibility for safety and avoid injuring others.	Ensuring no sensitive data is leaked during the project implementation.

Property Ownership	Respect property and intellectual rights.	IEEE: Acknowledge and credit contributions while respecting intellectual property.	Secure handling of user data and maintaining compliance with copyright laws.
Sustainability	Protect natural resources locally and globally.	IEEE: Disclose environmental risks promptly.	Minimal environmental impact as the project focuses on software with no physical waste generation.
Social Responsibility	Benefit society and communities.	IEEE: Improve understanding of technology and its appropriate application.	Promoting accessibility through free features for the general public.

Table 2: Areas of Professional Responsibility

Well-Performing Area:

- Communication Honesty: The team conducts weekly meetings with the client and advisor to provide updates, address concerns, and ensure alignment on the project. This transparency fosters trust and ensures all stakeholders are informed.

Improvement Area:

- Work Competence: Initially, inconsistent work quality and misunderstanding of client requirements caused minor delays. To improve, the team has scheduled skill-development sessions and emphasized early validation of designs with client feedback.

7.2 Four Principles

Broader Context	Beneficence	Non-Maleficence	Respect for Autonomy	Justice
Public Health, Safety, and Welfare	Accurate search results to inform decisions.	Protect user data and privacy.	Users choose to search specific media.	Free access ensures inclusivity.
Global, Cultural, and Social	Promote media inclusivity.	Avoid cultural insensitivity in design.	Allow user-defined language preferences.	Ads support free access for diverse users.
Environmental Impact	Reduce paper-based media consumption.	Minimize server energy consumption.	Users decide on media formats.	Equal media access globally.
Economic Impact	Free access for economic inclusivity.	Prevent over-reliance on paid services.	Users manage their subscription choices.	Affordable premium features for subscribers.

Table 3: Broader Context and Four Principles

Key Area:

- Public Health, Safety, and Welfare – Beneficence: The project aims to ensure accurate and unbiased media search results to support well-informed decisions by users.

Area Lacking:

- Environmental Impact – Justice: The software depends on digital infrastructure, which has global environmental consequences. To mitigate this, the team will explore options to use energy-efficient hosting services.

7.3 Virtues

Team Virtues:

1. Integrity

   Demonstrated by adhering to project deadlines and providing honest updates during client meetings.

2. Collaboration

   Encouraged by weekly team discussions and pair-programming sessions.

3. Empathy

   Ensured by designing features that address user needs across diverse demographics.

Individual Reflection:

- Virtue Demonstrated: Collaboration

  Importance: Ensures tasks are completed efficiently and knowledge is shared.

  Demonstration: Assisted team members during debugging sessions.

- Virtue to Improve: Time Management

  Importance: Avoid last-minute rushes and stress.

  Plan: Use time-blocking techniques to allocate specific hours for project work.

8 Closing Material

8.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals. What constrained you from achieving these goals (if something did)? What could be done differently in a future design/implementation iteration to achieve these goals?

The DigiClips Media Search Engine project aims to provide an efficient and user-friendly platform for media monitoring, catering to both general public users and subscribers. Through careful design and development, our team has focused on creating a General Public Page that balances accessibility and functionality, with features such as ad-supported searches and a call-to-action for upgrades.

Our progress thus far includes:

- Designing and prototyping the General Public Page.
- Implementing core functionalities like limited searches, ad integration, and user navigation.
- Collaborating with the client to ensure alignment with their goals and expectations.

While significant progress has been made, challenges remain in ensuring seamless integration, optimizing system performance, and maintaining scalability as the project evolves. Moving forward, our team will focus on rigorous testing, user feedback incorporation, and further refining the platform to meet the outlined requirements.

This project has reinforced our ability to apply technical knowledge, embrace collaborative practices, and address real-world design challenges, setting a strong foundation for future advancements.

8.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE). See link:

[1] DigiClips Client Documents - Project Requirements and Priority Lists. Provided by Henry Bremers & Bob Shapiro, DigiClips, 2024.

[2] Angular Documentation. Available at: https://angular.io/docs

[3] Material Design Guidelines for UI Components. Available at: https://material.io/design

[4] Feedback and meeting minutes from the DigiClips advisor and client, Henry Bremers & Bob Shapiro, 2024.

8.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.

Appendix A: System Architecture Diagram

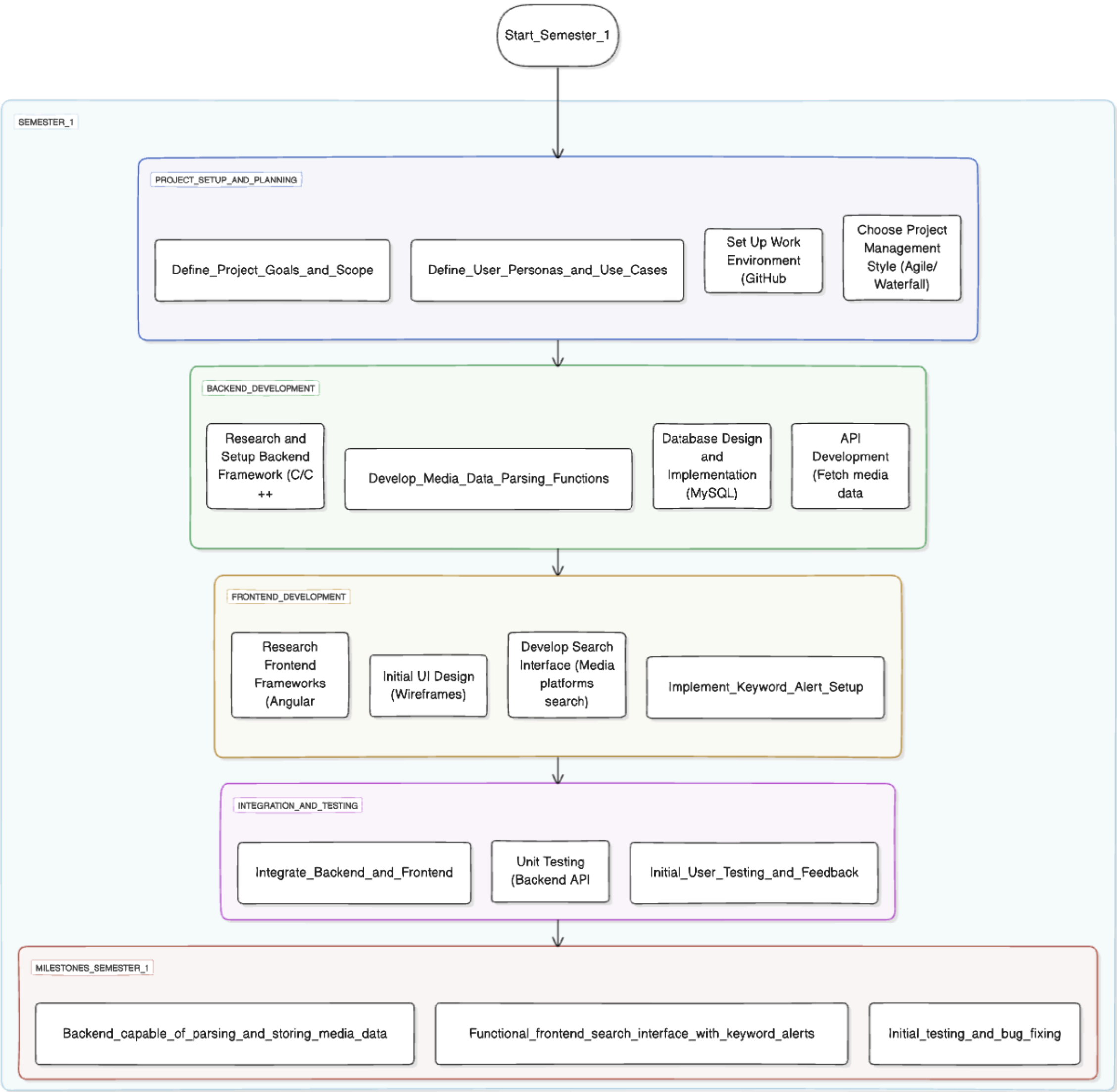

Start_Semester_1
SEMESTER_1
PROJECT_SETUP_AND_PLANNING
Define_Project_Goals_and_Scope
Define_User_Personas_and_Use_Cases
Set Up Work Environment (GitHub
Choose Project Management Style (Agile/ Waterfall)
BACKEND_DEVELOPMENT
Research and Setup Backend Framework (C/C ++
Develop_Media_Data_Parsing_Functions
Database Design and Implementation (MySQL)
API Development (Fetch media data
FRONTEND_DEVELOPMENT
Research Frontend Frameworks (Angular
Initial UI Design (Wireframes)
Develop Search Interface (Media platforms search)
Implement_Keyword_Alert_Setup
INTEGRATION_AND_TESTING
Integrate_Backend_and_Frontend
Unit Testing (Backend API
Initial_User_Testing_and_Feedback
MILESTONES_SEMESTER_1
Backend_capable_of_parsing_and_storing_media_data
Functional_frontend_search_interface_with_keyword_alerts
Initial_testing_and_bug_fixing

Figure 5: System Architecture Diagram

Appendix B: Project Timeline

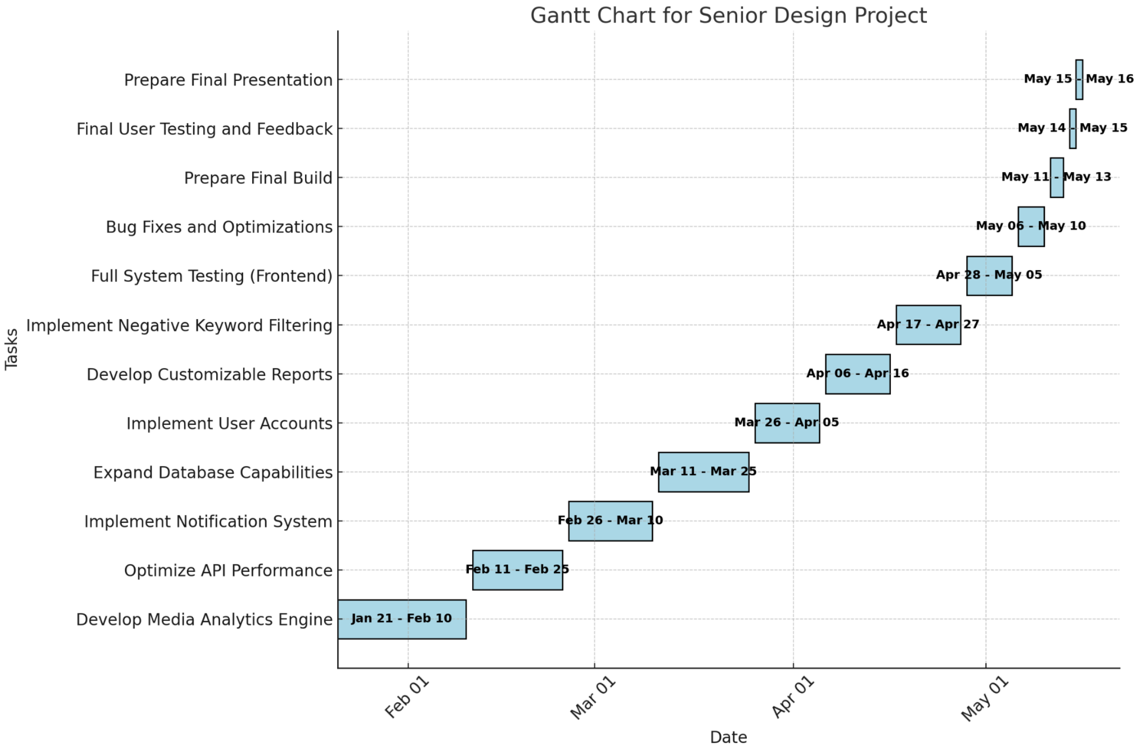


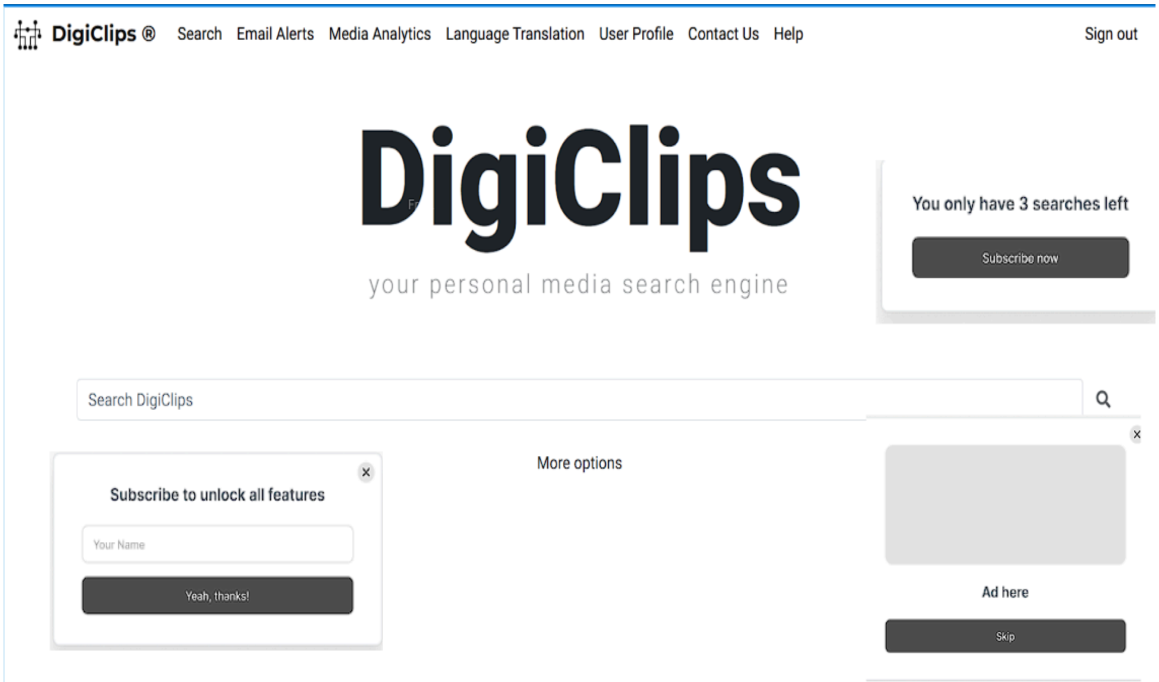
Table 5: Project Timeline

Appendix C: Team Contributions

<u>NAME</u>	**<u>Individual Contributions</u>** *(Quick list of contributions. This should be short.)*	**<u>Hours this week</u>**	**<u>HOURS cumulative</u>**
Edmund	- Delivered a Lightning Talk on ethics, and task assigned for backend with Nguyen	3	37
Nguyen	- Present the lightning talk about ethics and communication honesty, and have planned detailed tasks for the backend. Implement the partial working prototype; present prototype to clients.	3	39

Niharika	● Documented client and team meetings, ensuring all tasks, decisions, and key updates were accurately recorded. ● Presented the lightning talk about ethics and communication honesty, and have planned detailed tasks for the backend.	4	38
Varun	● I worked with my teammates to finish the lightning talk about communication and ethics.presented both in a presentation and to our clients as prototypes. have a far deeper comprehension of how to begin our activities.	4	36
Eshanth	● I collaborated with my teammates to complete the lightning talk on ethics and communication.presented prototypes to our clients and also in presentation. Gained a much more deeper understanding on how to get started on out tasks.	6	38

Appendix D: Prototype Designs



You only have 3 searches left

Subscribe to unlock all features

9 Team

9.1 Team Members

- Nguyen Do
- Edmund Lim
- Eshanth Chinthireddy
- Varun Reddy Yeduru
- Niharika Pathuri

9.2 Require skill set for the project

- Angular 18
- Database management (MySQL)
- C/C++
- Amazon Lightsail

9.3 Skill sets covered by the team

- Niharika and Varun enjoy creating websites and have experience with HTML/CSS; they have spent time familiarizing themselves with the Angular framework.
- Nguyen and Edmund have worked on projects using C/C++ and are familiar with Relational Database Management.
- Esanth is passionate about testing and has worked on projects related to testing.

9.4 Project Management style adopted by the team

Our team has adopted Agile project management to handle the project's progress and evolving requirements. By focusing on iterative development through manageable sprints, the team can implement, test, and refine features incrementally with continuous client feedback, mitigating risks and ensuring alignment with goals.

9.5 Initial Project Management Roles

- Niharika Pathuri - Documentation
- Edmund Lim - Client Interaction
- Nguyen Do - Team organization
- Varun Reddy Yeduru - Individual component Design
- Eshanth Reddy Chinthireddy - Testing

9.1 Team Contract

Team Name: SDMAY25-05

Team Members:

1. Niharika Pathuri
2. Edmund Lim
3. Nguyen Do
4. Varun Reddy Yeduru

5. Eshanth Reddy Chinthireddy

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings: We will hold an online meeting at 1:30 pm every Wednesday on Discord, and the meeting can last until 2:30 pm. It may end earlier if everything has been discussed.
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face): We will use Discord for team meetings, WhatsApp to notify each member of any updates, and Google Meet for meeting with the advisor and client.
3. Decision-making policy (e.g., consensus, majority vote): We will use majority voting to make important decisions.
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived): Niharika Pathuri will be responsible for taking minutes of the meetings, including the main points, time, and decisions made.

Participation Expectations

- Expected individual attendance, punctuality, and participation at all team meetings:
  - Everybody should be on time for meetings, also let your teams know prior to meetings if you are not going to make it.
- Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
  - Everyone is responsible for completing their assigned tasks on time and reporting to the team. If there is an issue preventing a member from completing their task, they should contact the team early to discuss ways to help them finish the work.
- Expected level of communication with other team members:
  - Everyone should respect each other and listen to each other's opinions. Everyone has the right to freely express their ideas and engage in discussions to achieve the common goal of the team's success, but they must respect each other.
- Expected level of commitment to team decisions and tasks:
  - Once team decisions are made, everyone should fully commit and support them. Each individual person should take responsibility for their assigned tasks.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
   a. Team Organization: Nguyen Dob.
   b. Client Interaction: Edmund
   c. Individual component Design: Varun Reddy Yeduru
   d. Project Documentation: Niharika Pathuri
   e. Testing: Eshanth
2. Strategies for supporting and guiding the work of all team members:
   a. We will ensure regular check-ins during our weekly meetings to track progress and address any challenges.
   b. We will ensure that everyone understands the team's goals and their individual responsibilities.

3. Strategies for recognizing the contributions of all team members:
    a. We will acknowledge and celebrate achievements during our meetings.
    b. Shoutout section at the end of the team meeting.
    c. We will maintain a shared document or log where each team member can document their contributions ensuring visibility of everyone's efforts.

Collaboration and Inclusion

1. 1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
    a. Niharika Pathuri: Niharika has experience in Java, Javascript, HTML, CSS, Node JS, MongoDB, and React. She is open to contributing to both Frontend and Backend development of the project.
    b. Edmund Lim: Edmund has full stack development skills with experience in PHP, javascript, jQuery, and MongoDB, essential for both frontend and backend development of the media search engine, Edmund will mainly focus on backend development in this project.
    c. Nguyen Do: Nguyen have experience with Java, HTML, CSS, Javascripts, Node JS, React, Express, EJS, Neo4J, MySQL and will be in charge of the front-end of the project, but Nguyen also want to learn more and help the back-end members when needed.
    d. Varun Reddy Yeduru: Varun will mainly handle the frontend development for our project. He has experience with Java, Javascript, HTML, CSS, Node JS, Angular and React. However, he is ready to help with the backend if needed.
    e. Eshanth Reddy Chinthireddy: I will handle the Angular upgrades, I have experience with Angular, Java, Javascript, Node.JS so I will be working on frontend features, backend features
2. Strategies for encouraging and supporting contributions and ideas from all team members:
    a. We will actively invite input from all team members during discussion and also brainstorming sessions.
    b. We will ensure that each member has a chance to speak by taking turns.
    c. We acknowledge that all contributions are valuable.
    d. We make it clear that diverse perspectives are welcome.
    e. We offer feedback that is positive and constructive, focusing on building upon the idea rather than criticizing the person.3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
    f. If a team member feels that the team environment is obstructing their opportunity, they should inform the team during the meetings or through private communication channels.
    g. We will be addressing these concerns which includes discussing the issue openly to understand the root cause and collaboratively finding a solution.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

   a. Our primary goal is to develop a functional media search engine that meets the requirements of the clients.
   b. We ensure regular communication and collaboration within the team to keep everyone aligned with the project's progress.

2. Strategies for planning and assigning individual and team work:
   a. We will do our best to allocate tasks in a way that best suits the abilities and preferences of the members.
   b. We also track down our tasks at the end of every team meeting.
3. Strategies for keeping on task:
   a. The team organizer checks in often to ensure everyone's on track and meeting their respective deadlines.
   b. We will support each other and share all difficulties and challenges.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
   a. We will ensure that all team members have agreed to the team contract and understand their responsibilities.
   b. If a team member does not meet obligations in the team contract, we will first try to address the issue privately to understand the cause and work towards a solution. The focus will be on open communication and solving the problem together, so the team can keep working effectively.
   c. When a team member commits a violation for the second time, we will report it to the advisor and school staff to find a solution.
2. What will your team do if the infractions continue?
   a. We will reach out to the instructor for further guidance.
   b. We will consider modifying the team contract if we have any problems that can not be resolved by the current version of the contract.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1. Niharika Devi Pathuri _______________________________ DATE 09/11/2024_________
2. Edmund Lim_____________________________________ DATE 09/11/2024_________
3. Eshanth reddy___________________________________ DATE 09/11/2024_________
4. Nguyen Do_______________________________________ DATE 09/11/2024_________
5. Varun Reddy Yeduru_________________________________ DATE 09/11/2024_________