



BackLogger

Spring 2024 Capstone II

Senior Project

Student: Fernando Collante (Team Lead)

Team Members: Lorena Miranda (Team Assistant), Luis Venegas,
Andres Vazquez, Brandon Torrens

Product Owner: Fernando Collante

Professor: Dr. Masoud Sidjadi



Role as Product Owner and Team Lead



Fig. 1 - Scrum Process from (2024, March 24). PM-Partners.
<https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/>

- **Role Clarity:** Product Owner & Team Lead
- **Communication:** Liaison between instructors and team members.
- **Meetings Led:** Planning, Daily Stand-ups, Backlog Grooming, Sprint Reviews, and Retrospectives.
- **Documentation:** Compiled and revised crucial meeting outcomes and scrum records.

Project Planning

- **Project Scope & Tech Stack:** Defined the fundamental technology and project boundaries.
- **Timeline:** Detailed timelines with sprint-specific milestones.
- **Role Delineations:** Assigned clear individual responsibilities for front-end and back-end development.
- **Adjustments:** Updated project plans in Sprint 7 to include learned shortcomings and future aspirations.

Project Overview created by Fernando Collante

Project Name: BackLogger
Members: Fernando Collante (Team Lead), Lorena Miranda (Team Assistant), Brandon Torrens, Luis Vemgas, Andres Vazquez
Timeframe: January 12, 2024 to April 28, 2024 (3 months, 2 weeks)

Objective: Develop a web application that allows users to create, manage and track lists of media items (books, movies, music, video games, and shows). Users will be able to mark items as, started, completed and add them to a favorites list as well as creating lists with custom names. This web application is aimed to reduce time spent looking for what media to consume, keep track of progress through one's backlog, and ultimately reduce brain rot by trying to reduce the mindlessness that comes with consuming media.

Scope:
UI - Creation and management of media lists; viewing details of media items; user interactions for adding and deleting list items; creating lists with custom names (apart from the default to be completed/started/completed/favorites).
Front Page Possible Feature - Displaying currently top-viewed and/or rated items for each media type (Alternate idea: Summary of recent activity).
Media Item Details - Displaying name, image (album cover, movie poster, box art, etc.) and sources for streaming/purchasing.
Completion/Favorites - Features to mark items as started, completed and to favorites (from a pending list).
Data Management - Backend service for handling data storage and retrieval. Initially, users should be able to use csv/text files to import/export lists of media.

Project Timeline (Estimated):

1. January 12, 2024 - January 28, 2024 (1 Sprint) Initial Research and Product Set-up
Set-up development environments, finalize which technologies will be used, further define project scope and architecture.
2. January 29, 2024 - February 25, 2024 (2 Sprints) Development of Core Features
Designing and implementing the core functionality to create and manage lists, add new items, basic UI components.
3. March 02, 2024 - March 17, 2024 (1 Sprint) Continued Development
Implementation of started/completed/favorites list, enhance user interactivity, consider possible additional features at this time.



4. March 18, 2024 - March 31, 2024 (1 Sprint) Integrate Frontend and Backend
Ensure proper data and API integration. Initial testing of application with full functionality.
5. April 01, 2024 - April 14, 2024 (1 Sprint) Developer Testing/User Testing/Feedback
General bug fixing, present project to a variety of users and record feedback, adjust UI/UX according to feedback.
6. April 15, 2024 - April 28, 2024 (1 Sprint) Deployment/Documentation/Presentation
Handle deployment, compile all produced documents throughout development time, unknown presentation date.

Frontend Duties:

Designing the User Interface

- Create mock-ups/wireframe diagrams for the applications layouts/interfaces
- Develop UI components like list views, item cards, buttons for adding/editing lists, navigation menus, etc.

Implementing Interactivity

- Develop the logic for user interaction such as actually adding/removing list items, marking items as started/completed/favorites.
- Client-side form verification (list names and if users will be allowed to add their own entries)

Responsive Design

- Ensure application is responsive and works across a range of devices (ideally cover usual resolutions of computers and modern mobile devices).
- Implement consistent, user-friendly layout and CSS.

Backend Duties:

Develop the API

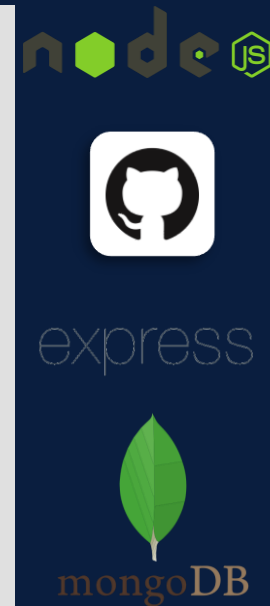
- Design and implement RESTful API endpoints for handling frontend requests.
- Ensure proper request validation and error handling within the API.
- Start and maintain documentation of the API being developed.

Data Management

- Implement logic for reading/writing data to and from csv/text files or a database (PostgreSQL)
- Manage data structures/storage solutions.

Security

- Implement basic API security protocols like input validation (if frontend and backend are hosted on different sources then CORS, Cross Origin Resource Sharing, would need to be implemented as well)
- If time permits, implement user authorization and authorization systems.



Duties of Both Teams:

Both teams are required to produce documentation of their journey to substantiate the development process.

Both teams need to coordinate and understand the projects data needs and API structure (and that all API requirements are clear).

Both teams need to test the completed project and plan for deployment in tandem to ensure a smooth launch for the prototype to showcase.

Technological Stack:

Front-end Development - HTML, CSS and JavaScript
Back-end Development - Node.js/Express.js
Database Program - MongoDB
Integrated Development Environment - Visual Studio Code
Version Control - GitHub

Shortcomings:

Functionality Gaps:

- Frontend: Incomplete modal implementation (design and functionality).
- Backend: Incomplete implementation of API calls for media titles.

Learning Curve: The learning curve for new technologies impacted our development timeline.

Wishlist:

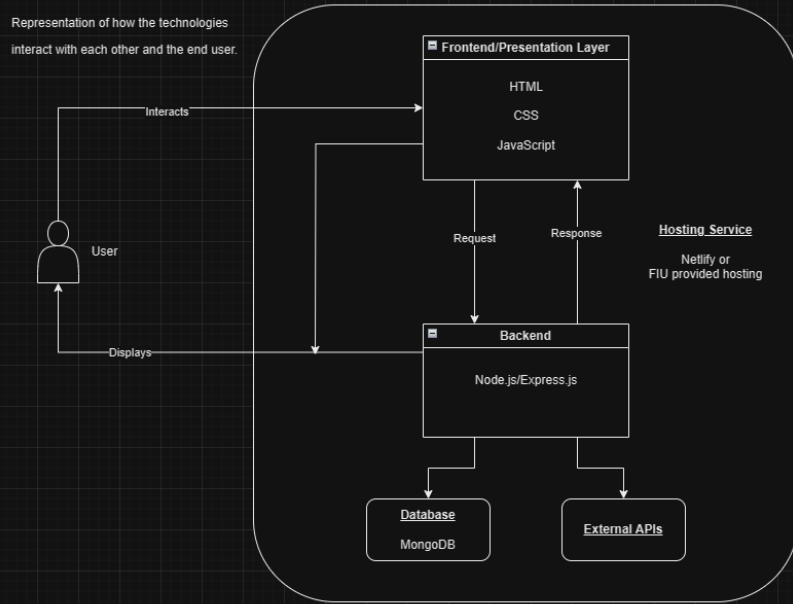
- Implement all initially planned features that didn't make it in during our development time.
- Achievements/Badges for completing media.
- List persistence across devices and also facilitate development of a sharing system for lists.
- Ability to import/export lists into CSVs or another format for offline use/sharing.



Architectural and Data Flow Diagrams

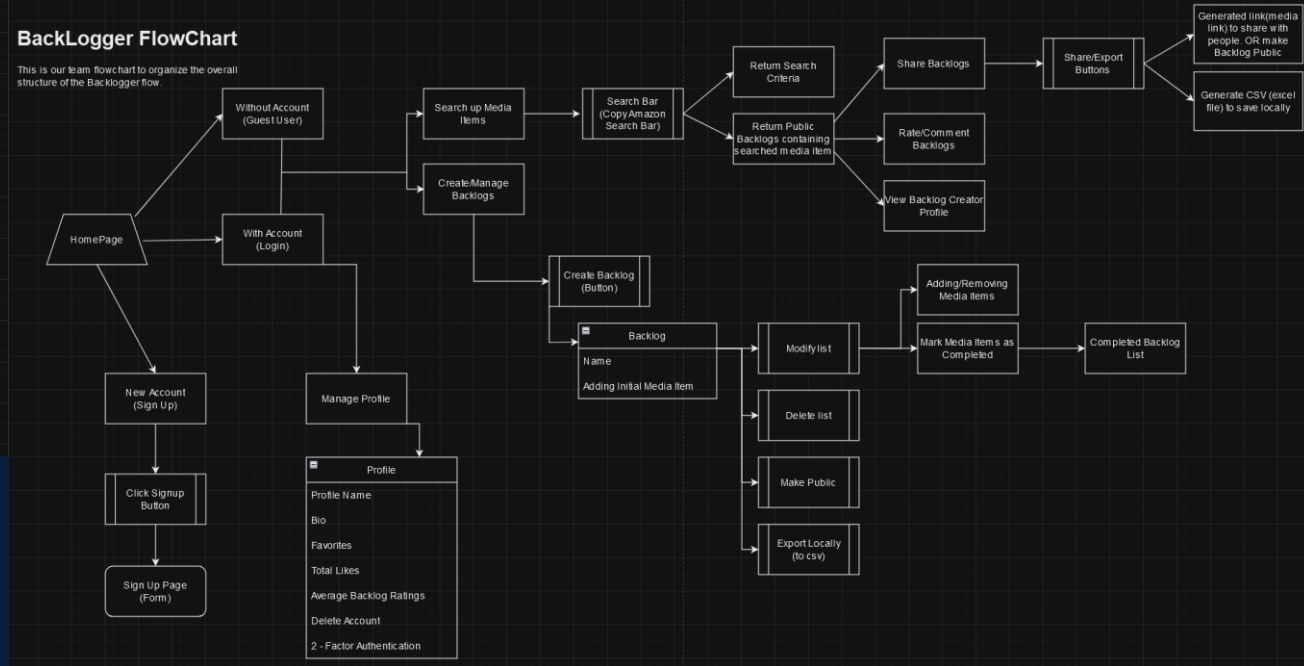
BackLogger Architectural Diagram

Representation of how the technologies
interact with each other and the end user.



BackLogger FlowChart

This is our team flowchart to organize the overall
structure of the BackLogger flow.





Final Presentations and Guides

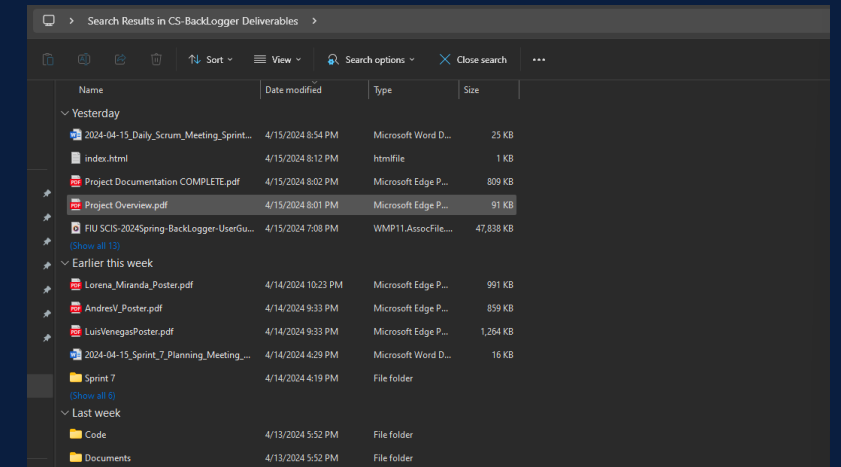
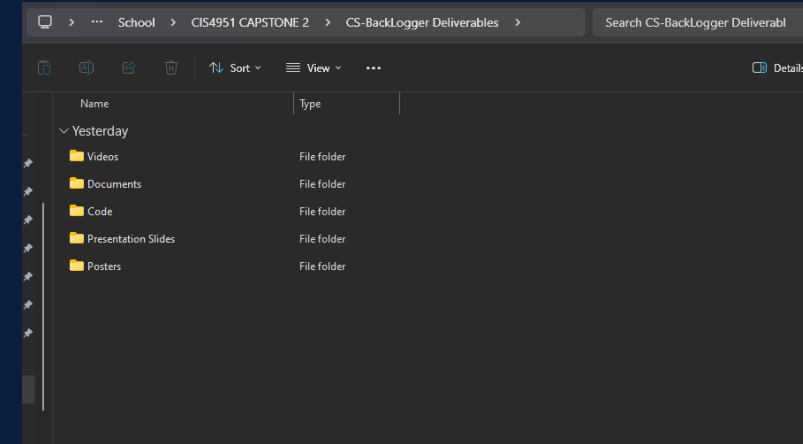
- [Introductory Video](#): Overview for the BackLogger project.
- [Shortcomings/Wishlist Video](#): This video addresses functional gaps in the project and also lists features that the original BackLogger team would like to see implemented in the future. These features range from originally planned to be in the project to lofty ideas.
- [User Guide/Demo](#): Presented by Brandon Torrens and myself, featuring both back-end and front-end functionalities.
- [Presentation Slides](#)



Final Documentation

Comprehensive Documentation Review

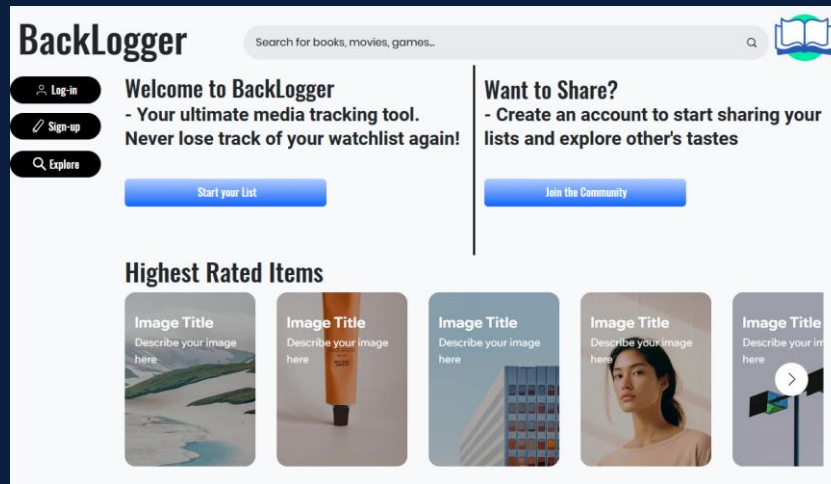
- **Documentation Integrity:** Ensured all documents accurately reflected project developments and outcomes.
- **Project Submission:** Organized and submitted deliverables for both showcase and final submission.



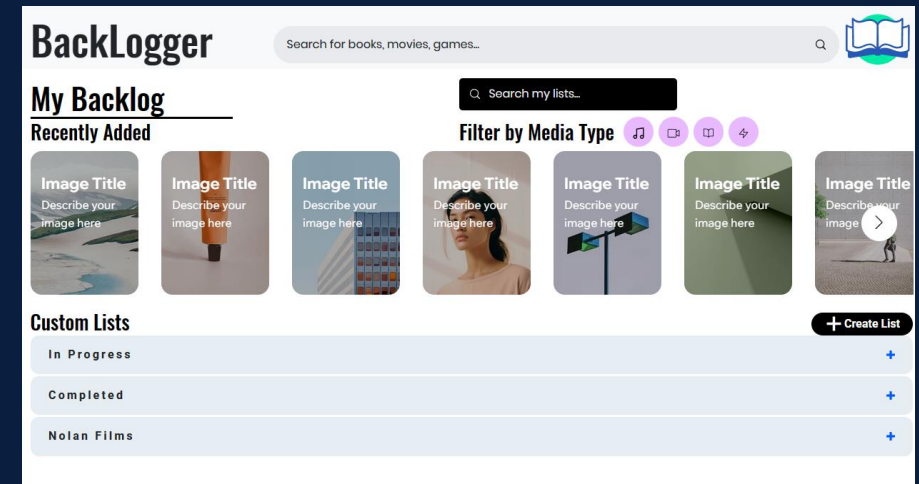


Frontend Design Contributions

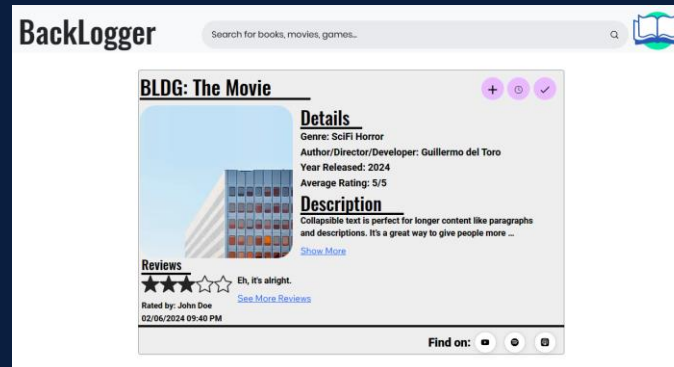
Collaboration: Co-designed
with Lorena Miranda and
Andres Vazquez.



- **Home Page Design:** Gateway to project functionality.



- **Backlog List Page:** Structured view of pending items.



- **Detail Pop-Up:** Enhanced user interaction with media items.