

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: BackLogger

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

Product Owner(s): Fernando Collante

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document outlines the development of BackLogger, a web application designed to manage and track media consumption including books, movies, music, videos, games, and shows. The platform allows users to categorize media items with custom labels, track their consumption status, and maintain lists for organized access. The goal is to enhance user experience in managing media consumption efficiently, thereby reducing time spent searching and increasing engagement with content.

Table of Contents

INTRODUCTION	PAGE 5
CURRENT SYSTEM	PAGE 5
PURPOSE OF NEW SYSTEM	PAGE 5
USER STORIES	
IMPLEMENTED USER STORIES	PAGE 6
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	PAGE 7
SPRINTS PLAN	PAGE 7
SYSTEM DESIGN	
PROJECT SCOPE	PAGE 8
FRONT END DUTIES	PAGE 8
BACK END DUTIES	PAGE 9
WHOLE TEAM DUTIES	PAGE 9
APPENDIX	
APPENDIX A - UML DIAGRAMS	PAGE 10
APPENDIX B - USER INTERFACE DESIGN	PAGE 11
APPENDIX C - SPRINT REVIEW REPORTS	PAGE 16
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	PAGE 19

INTRODUCTION

BackLogger is designed as a comprehensive solution to manage and catalog various types of media, aiming to streamline the way users interact with their entertainment libraries. This system introduces a structured way to monitor and organize media consumption through an intuitive web interface.

Current System

As of the project's conception, no centralized system exists that meets all the user-specified requirements for media management as envisioned for BackLogger. Existing solutions do not fully support the integration of diverse media types or the customization features that BackLogger proposes.

Purpose of New System

The purpose of developing BackLogger is to provide a unified platform that allows users to efficiently manage and track their media items. It aims to reduce cognitive load and enhance the user experience by simplifying access to media through tailored user interactions.

USER STORIES

Implemented User Stories

- User Story 1: Research APIs
- User Story 2: Research Frameworks
- User Story 3: Language Selection (Revised and merged with User Story 2, estimated working hours for this User Story were reallocated to User Story 5: Create Initial Project Diagrams/Design Docs)
- User Story 4: Initialize Repository on GitHub
- User Story 5: Create Initial Project Diagrams/Design Docs
- User Story 6: Set-up Basic Project Structure
- User Story 7: Node.js Framework Tutorials
- User Story 8: Vue.js Framework Tutorials
- User Story 9: Set-up Backend Dev. Environment
- User Story 10: Set-up Frontend Dev. Environment
- User Story 11: Transpose Layouts to HTML/CSS/JS
- User Story 12: Complete Test Project
- User Story 13: Integrate Backend with Frontend
- User Story 14: Review Project Documentation
- User Story 15: Develop Showcase Poster
- User Story 16: Additional Front-end Functionality/Layout Modifications to Web App
- User Story 17: Develop and Integrate User Management Feature
- User Story 18: Additional Back-end Functionality/Modifications
- User Story 19: Attend Showcase
- User Story 20: Prepare Deliverables

PROJECT PLAN

Hardware and Software Resources

- Technologies used: HTML, CSS, JavaScript, Node.js/Express.js, MongoDB
- Development environments: Visual Studio Code
- Version control: GitHub

Sprints Plan

1. *January 12, 2024 – January 28, 2024 (Sprint 1) 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 (Sprints 2 and 3) 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 (Sprint 4) 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 (Sprint 5) Integrate Frontend and Backend
Ensure proper data and API integration. Initial testing of application with full functionality.*
5. *April 01, 2024 – April 14, 2024 (Sprint 6) 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 (Sprint 7) Deployment/Documentation/Presentation
Handle deployment, compile all produced documents throughout development time, presentation on 04/19/2024.*

SYSTEM DESIGN

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).

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/txt files to import/export lists of media.

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/favorite.
- 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/txt files or a database (PostgreSQL)
- Manage data structures/storage solutions.

Security

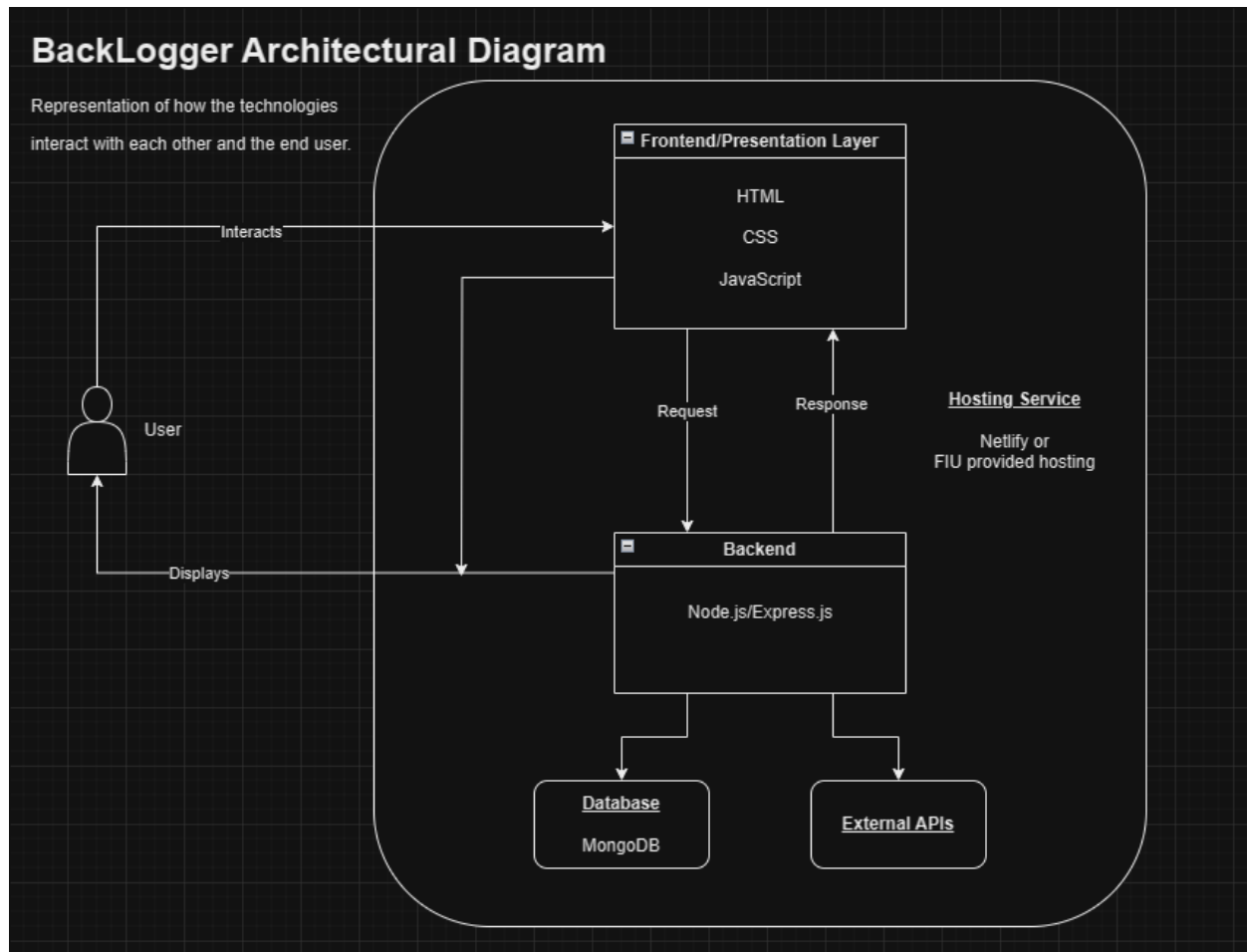
- Implement basic API security protocols like input validation (if frontend and backend are hosted on different sources then CORS, Cross Origin Resource Sharing, would need to be implemented as well)
- If time permits, implement user authorization and authorization systems.

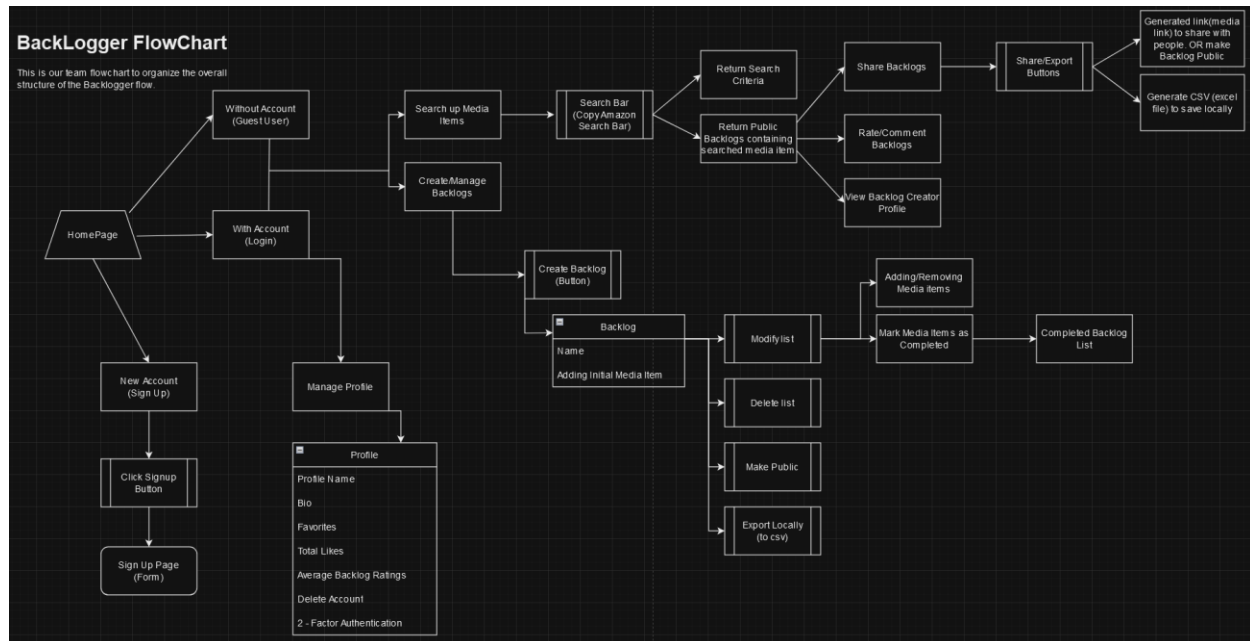
Duties of Both Teams:

- 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.

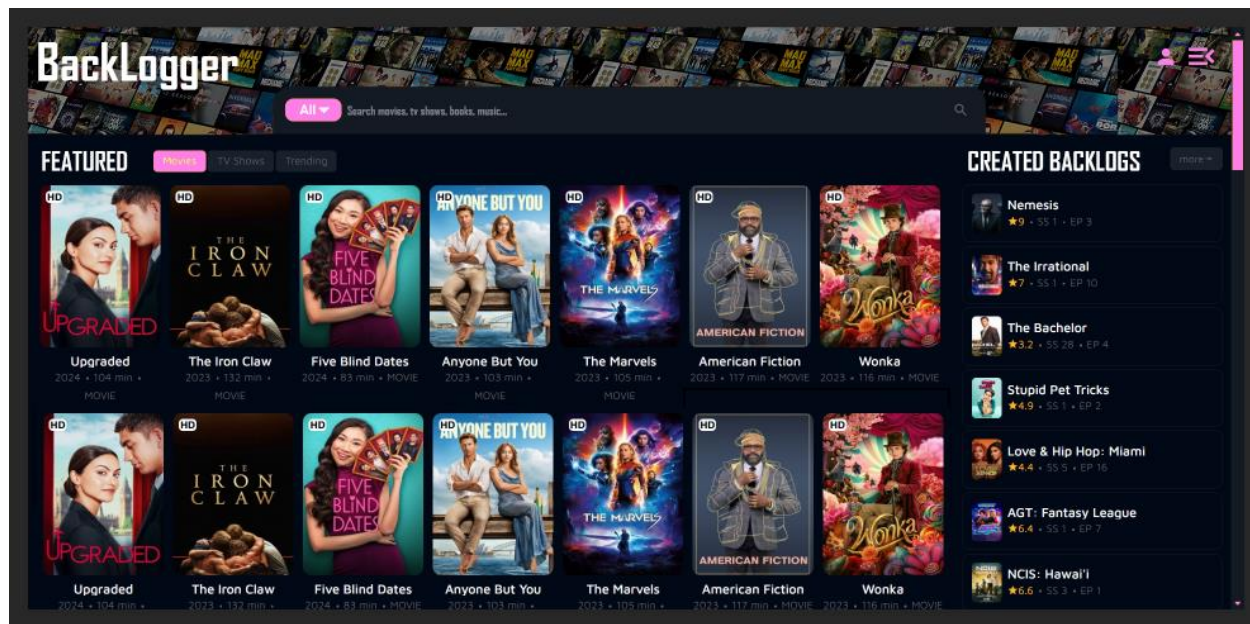
APPENDIX

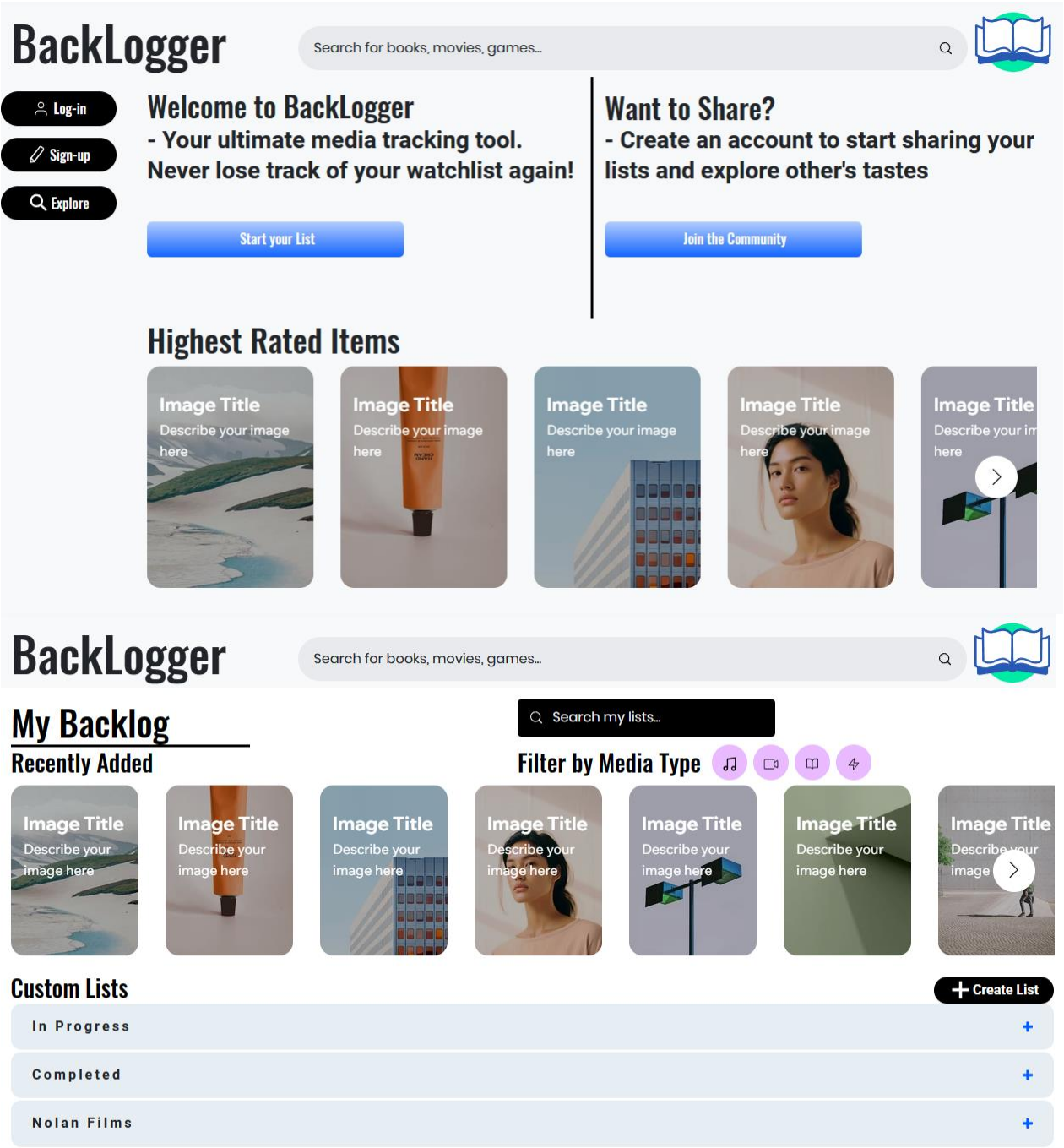
Appendix A - UML Diagrams

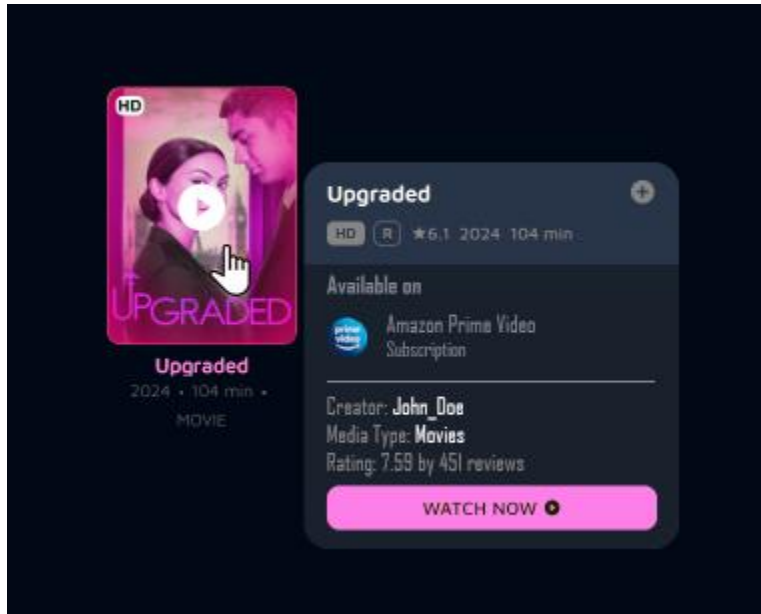




Appendix B - User Interface Design

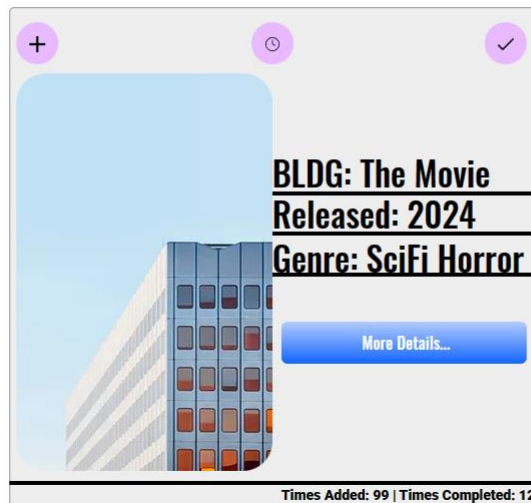


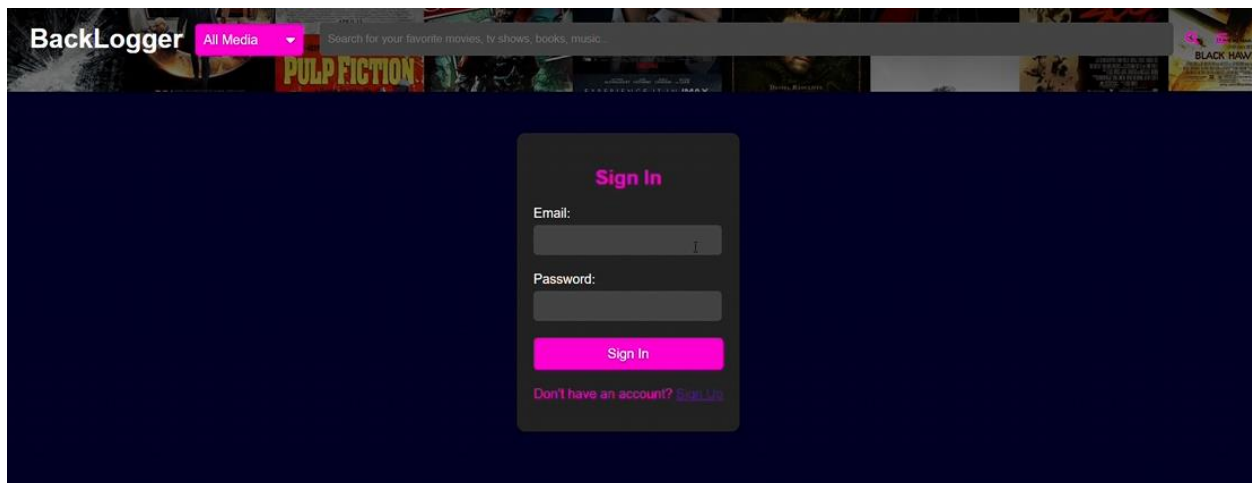
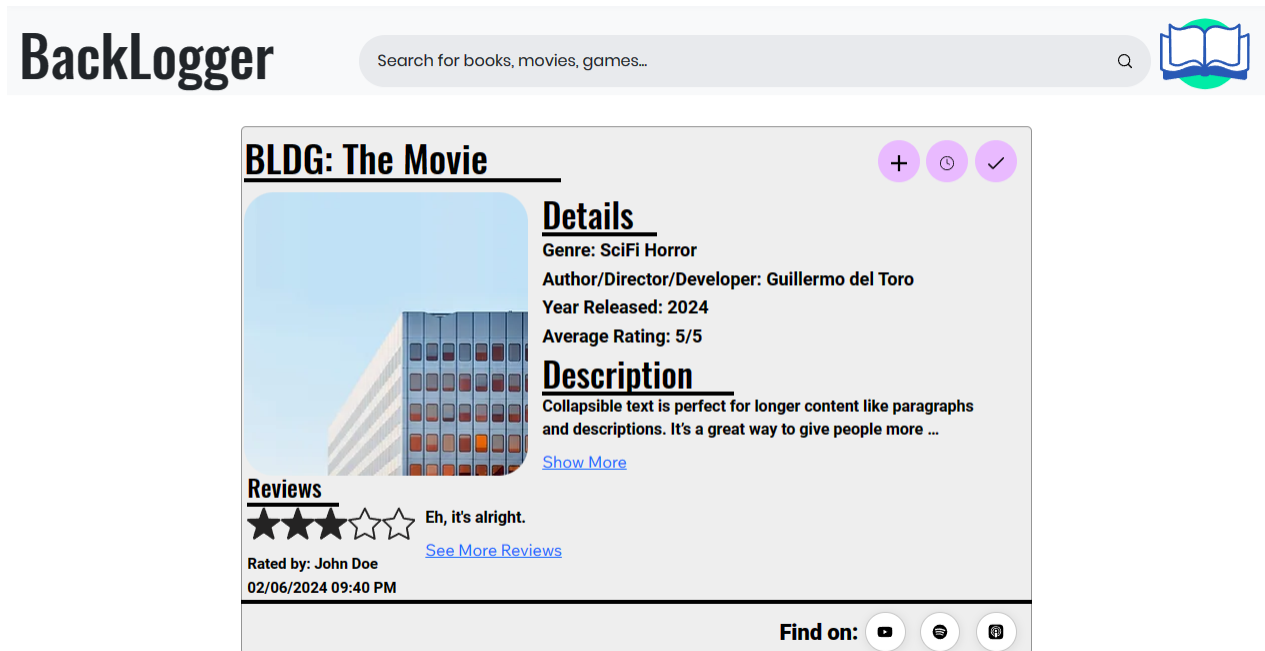


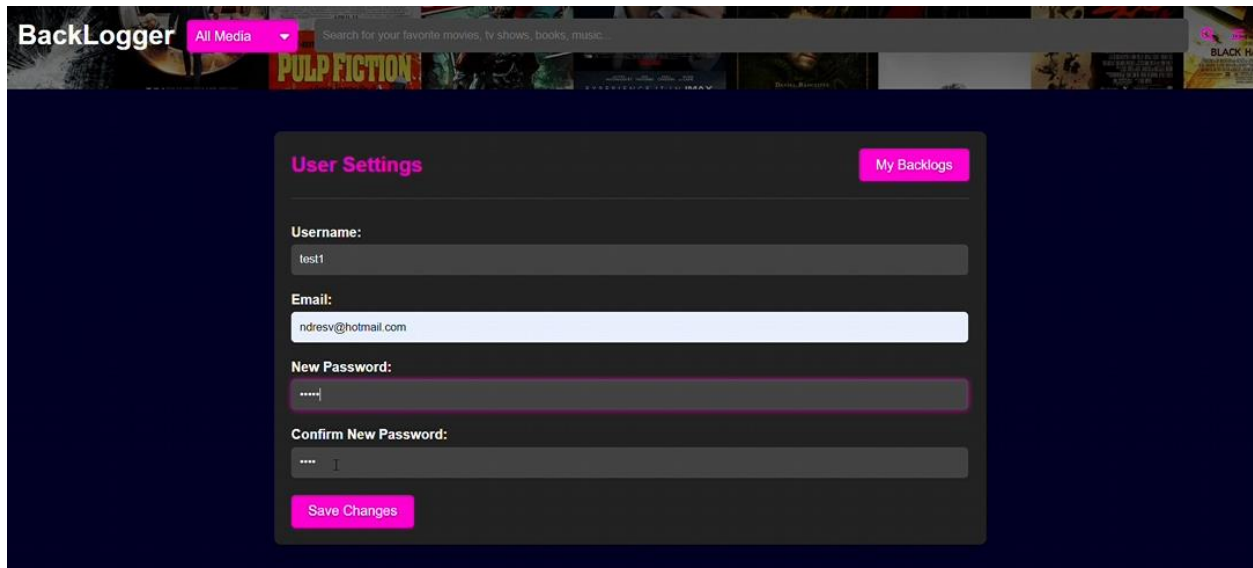


BackLogger

Search for books, movies, games...



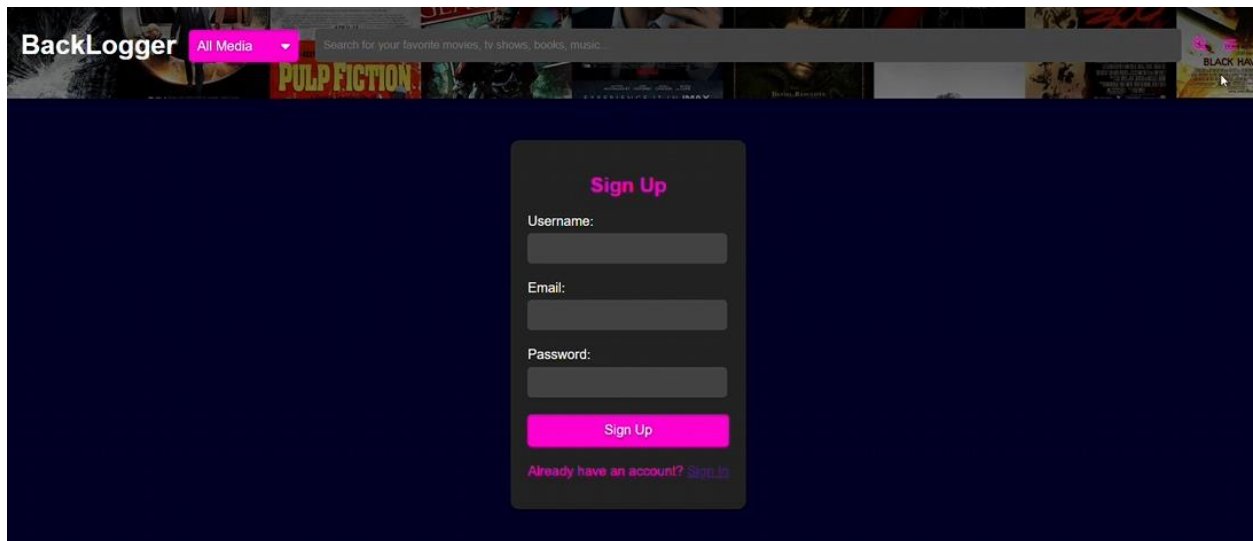




The screenshot shows the 'User Settings' page of the BackLogger application. The header includes the 'BackLogger' logo, a dropdown menu set to 'All Media', and a search bar. The main content area is a dark grey panel with the title 'User Settings' in pink. A pink button labeled 'My Backlogs' is in the top right corner. The form contains four input fields: 'Username' (with 'test1' entered), 'Email' (with 'ndresv@hotmail.com' entered), 'New Password' (with masked characters), and 'Confirm New Password' (with masked characters). A pink 'Save Changes' button is at the bottom.

BackLogger All Media Search for your favorite movies, tv shows, books, music...

User Settings

 My Backlogs

The screenshot shows the 'Sign Up' page of the BackLogger application. The header is identical to the previous screenshot. The main content area is a dark grey panel with the title 'Sign Up' in pink. The form contains three input fields: 'Username', 'Email', and 'Password'. A pink 'Sign Up' button is at the bottom. Below the button, there is a link that says 'Already have an account? Sign In'.

Sign Up

Appendix C - Sprint Review Reports

Sprint 1 Review Report (01/27/2024)

Sprint Goal: Initiate the project foundation by researching and setting up basic structures.

- **Completed User Stories:**
 - **User Story 1:** Research APIs
 - **User Story 2 & 3:** Research Frameworks and Language Selection
 - **User Story 4:** Initialize Repository on GitHub
- **Challenges and Solutions:**
 - Difficulty in aligning technology stack which delayed some initial setups.
- **Rejected User Stories:**
 - **User Story 5:** Initial Project Diagrams/Design Docs partially completed; pending detailed use case diagrams.
 - **User Story 6:** Basic Design Structure initiation delayed due to learning curve on frameworks.
- **Action Items:**
 - Complete pending diagrams and finalize the basic project structure setup.

Sprint 2 Review Report (02/11/2024)

Sprint Goal: Establish development environments and foundational project diagrams.

- **Completed User Stories:**
 - **User Story 5:** Create Initial Project Diagrams/Design Docs
 - **User Story 7 & 8:** Node.js and Vue.js Framework Tutorials
- **Challenges and Solutions:**
 - Initial environment setups were more time-consuming than anticipated.
- **Rejected User Stories:**
 - **User Story 6:** Continued challenges with the basic project structure on GitHub.
 - **User Story 9 & 10:** Backend and Frontend Development Environment setups pending completion.
- **Action Items:**
 - Prioritize completion of development environment setups and basic project structure.

Sprint 3 Review Report (02/24/2024)

Sprint Goal: Complete environment setups and begin frontend and backend integration.

- **Completed User Stories:**
 - **User Story 7 & 8:** Continued tutorials on Node.js and Vue.js frameworks.
 - **User Story 9 & 10:** Setup Backend and Frontend Development Environments.
- **Challenges and Solutions:**
 - Integration of backend and frontend environments took additional resources.
- **Rejected User Stories:**
 - **User Story 6:** Basic project structure still pending.
 - **User Story 11 & 12:** Layouts and test project completion delayed.
- **Action Items:**
 - Resolve issues with project structure and advance layout designs and testing.

Sprint 4 Review Report (03/16/2024)

Sprint Goal: Implement basic project structure and start frontend development.

- **Completed User Stories:**
 - **User Story 6:** Set-up Basic Project Structure.
- **Challenges and Solutions:**
 - Encountered significant delays transitioning designs to functional HTML/CSS/JS.
- **Rejected User Stories:**
 - **User Story 11:** Transpose Layouts to HTML/CSS/JS not fully functional.
 - **User Story 12:** Complete Test Project; backend user management pending.
- **Action Items:**
 - Complete frontend interactivity and finalize test project setups.

Sprint 5 Review Report (03/31/2024)

Sprint Goal: Review and refine project documentation and backend integration.

- **Completed User Stories:**
 - **User Story 14:** Review Project Documentation.
- **Challenges and Solutions:**
 - Integration complexities between frontend and backend highlighted.
- **Rejected User Stories:**
 - **User Story 11, 12 & 13:** Challenges with frontend functionalities and backend integration.
- **Action Items:**
 - Focus on resolving frontend and backend integration issues and improving functionality.

Sprint 6 Review Report (04/13/2024)

Sprint Goal: Finalize integration and prepare for project showcase.

- **Completed User Stories:**
 - **User Story 13:** Integrate Backend with Frontend.
 - **User Story 15:** Develop Showcase Poster.
 - **User Story 17:** Develop and Integrate User Management Feature.
- **Challenges and Solutions:**
 - Final integration required extensive debugging and adjustments.
- **Rejected User Stories:**
 - **User Story 16:** Additional frontend functionality/layout modifications needed.
- **Action Items:**
 - Address the frontend functionality issues and prepare for the final project presentation.

Sprint 7 Review Report (04/28/2024)

Sprint Goal: Finalize all functionalities, attend the project showcase, and prepare all deliverables for project completion.

- **Completed User Stories:**
 - **User Story 16:** Additional Front-end Functionality/Layout Modifications to Web App
 - **User Story 18:** Additional Back-end Functionality/Modifications
 - **User Story 19:** Attend Showcase
 - **User Story 20:** Prepare Deliverables
- **Challenges and Solutions:**
 - None.
- **Action Items:**
 - None.

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents**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 permanence 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.
- Front Page Possible Feature – Displaying currently top-viewed and/or rated items for each media type (Alternate idea: Summary of recent activity).

Installation Manual created by Brandon Torrens

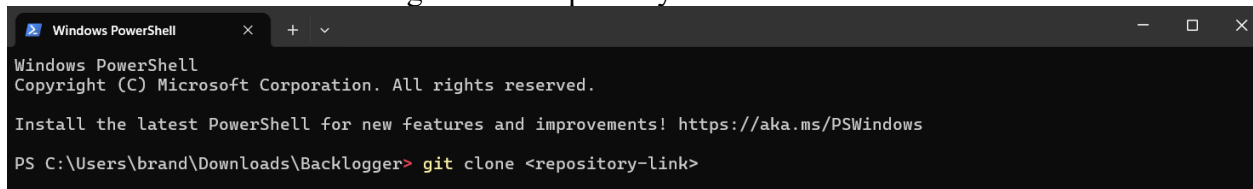
This guide is intended to assist you with setting up your development environment for the project to continue it.

Prerequisites:

- Install the latest version of Node.js: <https://nodejs.org/en/download>
- Please install your preferred advanced text editor. In this guide we will be using VS Code: <https://code.visualstudio.com/download>
- Please make sure to have installed git for version control management: <https://git-scm.com/downloads>

Setup Steps:

- Firstly, please clone the repository to a directory on your PC. You can use the following command in terminal. “git clone <repository-link>”

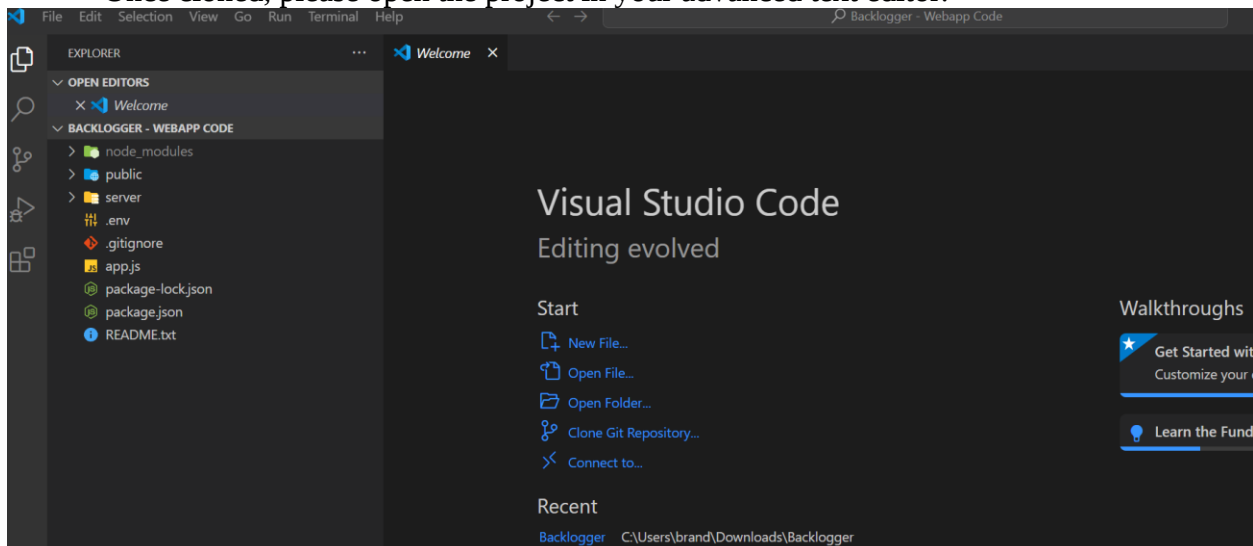


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

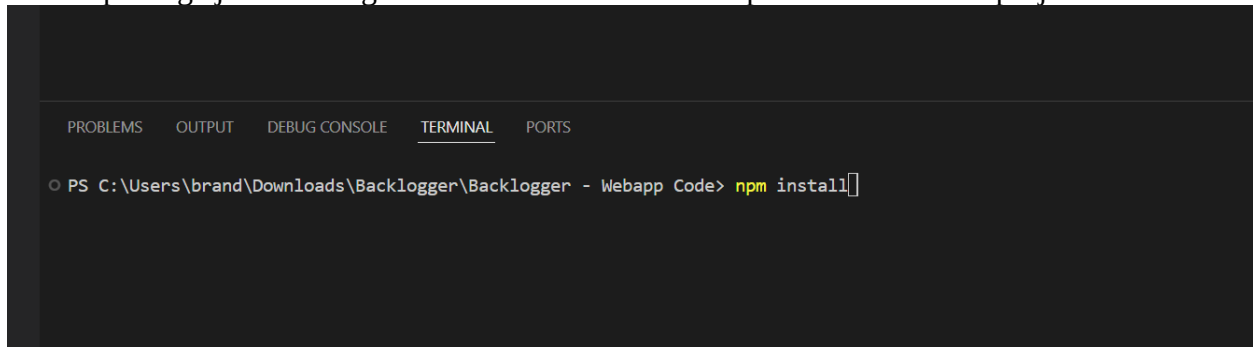
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\brand\Downloads\Backlogger> git clone <repository-link>
```

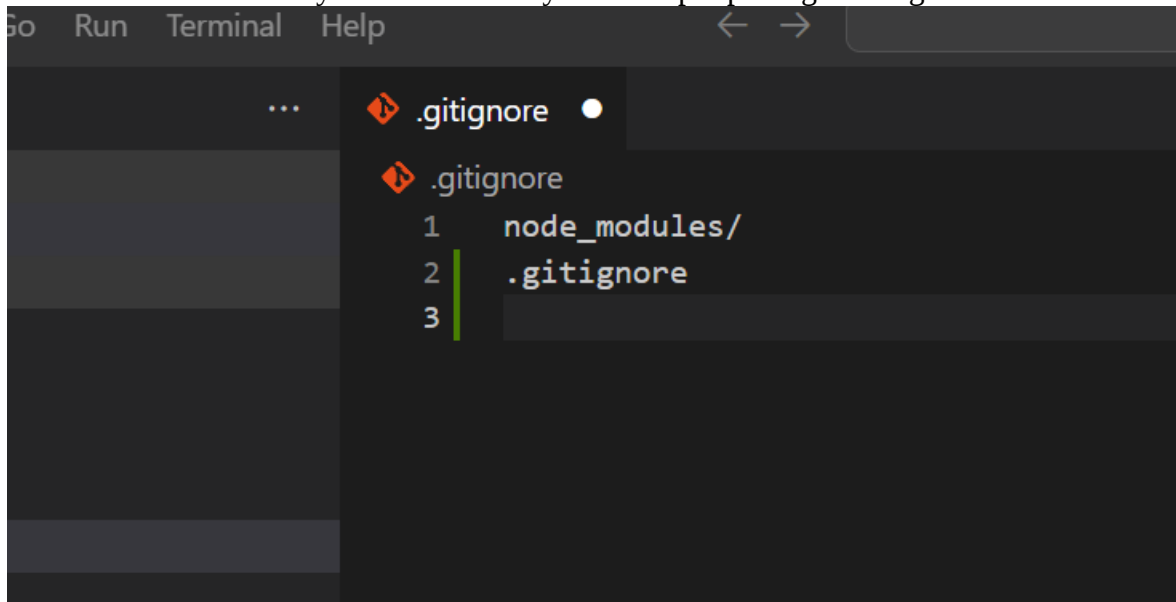
- Once cloned, please open the project in your advanced text editor.



- When in the project, please open a new terminal within the advanced text editor.
- Please type in the following into the terminal “npm install” this will open the package.json and begin to download all needed dependencies for this project.



- Then make sure to add a .gitignore file to your project structure. Please added the following lines to the file “node-modules/” and “.gitignore”. You shouldn’t upload these files to your version control system, as node-modules can weigh allot and be heavy in a version control system. This is why we use npm package manager to handle this for us.



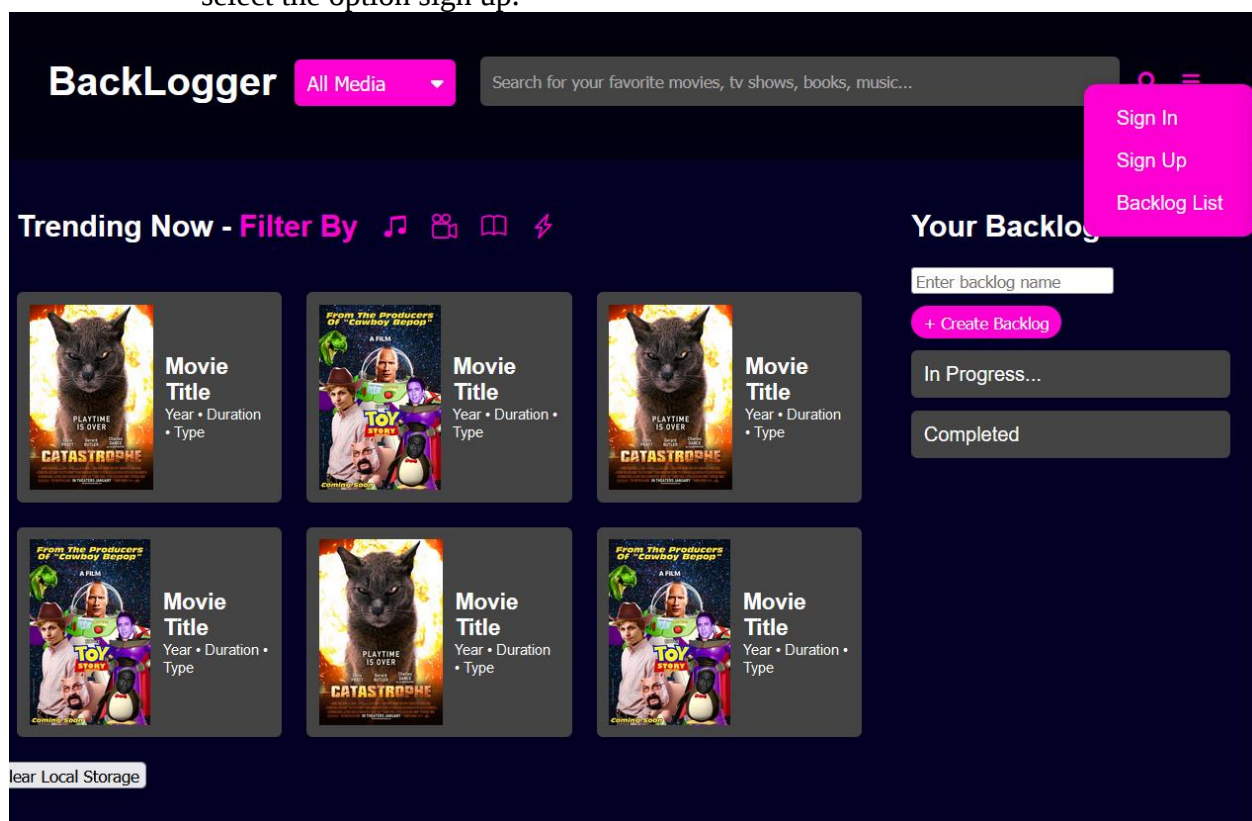
Then make sure you setup your debugger properly. To do this, open the app.js file, and then press either F5 or the button to run/debug your code. Make sure to select the node.js compiler.

User Manual created by Brandon Torrens

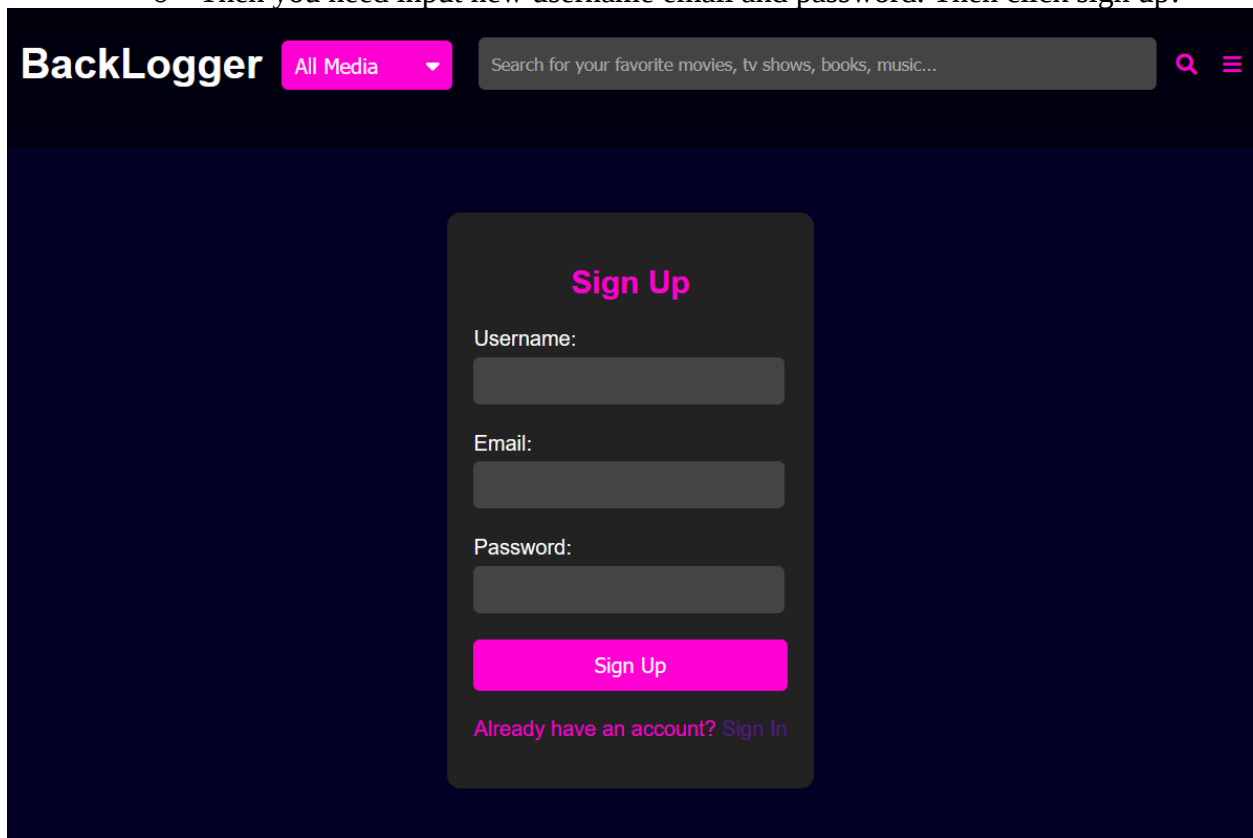
This user manual is intended to show the user how to explore and use the website. PLEASE NOTE this project is not finished and it is intended to be updated as future students improve the project and continue to add more functionality to the web application.

The following demonstrations show off what is currently working when the node server is running and connecting to it locally on the same network.

- User Creation
 - To create a new user, you need to click on the three bars on the top right and select the option sign up.

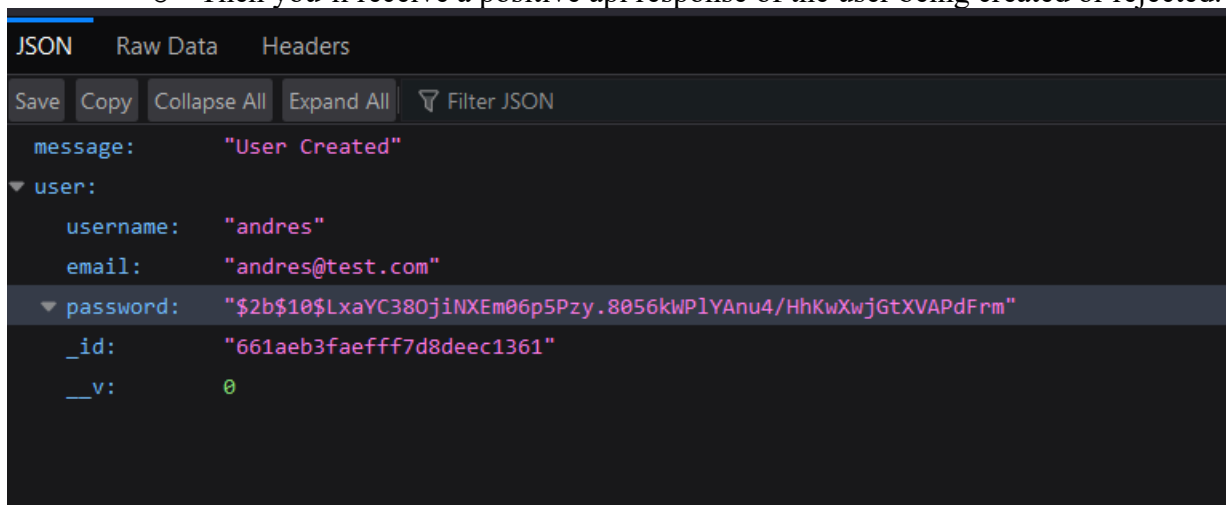


- Then you need input new username email and password. Then click sign up!



The screenshot shows the BackLogger web application interface. At the top, there is a navigation bar with the 'BackLogger' logo, a dropdown menu set to 'All Media', a search bar with the placeholder text 'Search for your favorite movies, tv shows, books, music...', and a magnifying glass icon. Below the navigation bar, a central 'Sign Up' form is displayed. The form has a title 'Sign Up' in red. It contains three input fields: 'Username:', 'Email:', and 'Password:'. Below these fields is a red 'Sign Up' button. At the bottom of the form, there is a link that says 'Already have an account? Sign In'.

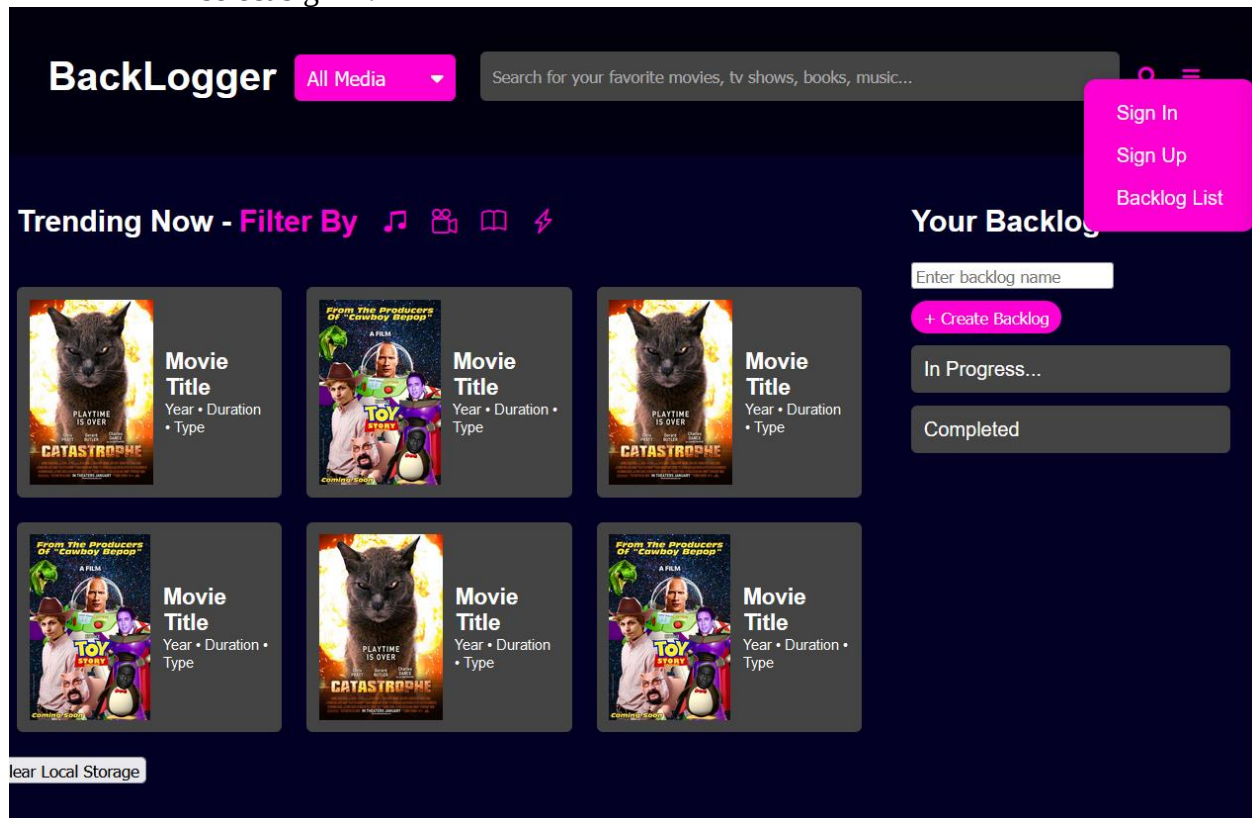
- Then you'll receive a positive api response of the user being created or rejected.



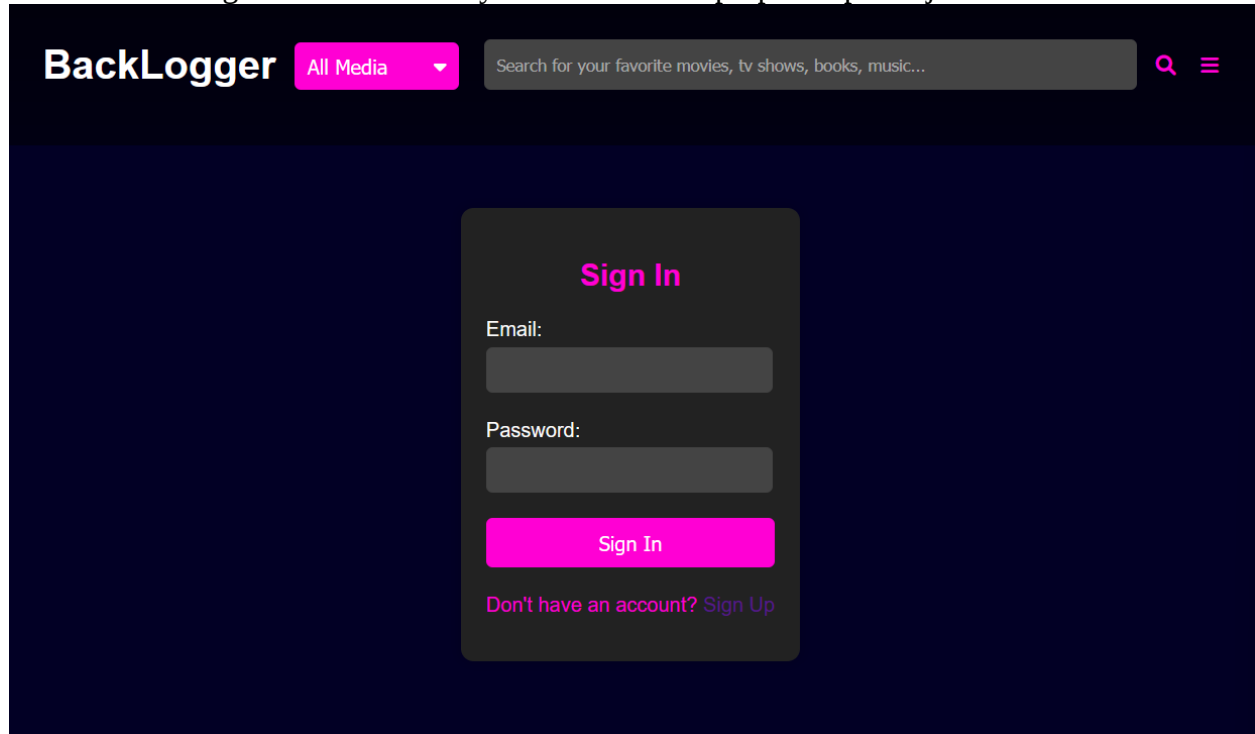
The screenshot shows a JSON API response in a developer tool. The response is a JSON object with the following structure:

```
{  "message": "User Created",  "user": {    "username": "andres",    "email": "andres@test.com",    "password": "$2b$10$LxaYC380jiNXEm06p5Pzy.8056kWPlYAnu4/HhKwXwjGtXVAPdFrm",    "_id": "661aeb3faefff7d8deec1361",    "__v": 0  }}
```

- User Sign on
 - To sign in with a user, you need to select the three bars in the top right, and then select sign in.



- Then you'll be presented with the sign in page, simply input valid credentials to sign in. If it not valid you will receive a proper response json.



The screenshot shows the BackLogger application interface. At the top, there is a dark blue header bar. On the left, the text "BackLogger" is displayed in white. Next to it is a red button labeled "All Media" with a small downward arrow. To the right of the button is a search bar with the placeholder text "Search for your favorite movies, tv shows, books, music...". On the far right of the header bar are two icons: a magnifying glass and a hamburger menu icon.

Below the header bar, the main content area has a dark blue background. In the center, there is a white rounded rectangle containing the "Sign In" form. The form has the title "Sign In" in red. Below the title are two input fields: "Email:" and "Password:". Below these fields is a red button labeled "Sign In". At the bottom of the form, there is a link that says "Don't have an account? Sign Up" in red.

- When signed in successfully you'll be presented with the user profile configuration page.

The screenshot shows the 'BackLogger' application interface. At the top, there is a dark blue header with the 'BackLogger' logo on the left, a pink 'All Media' button with a dropdown arrow, a search bar with the placeholder text 'Search for your favorite movies, tv shows, books, music...', and a magnifying glass icon and a hamburger menu icon on the right. Below the header, the main content area is dark blue. A central dark gray card contains the 'User Settings' section. The 'User Settings' title is in pink, and there is a pink 'My Backlogs' button in the top right corner of the card. The settings form includes four input fields: 'Username:', 'Email:', 'New Password:', and 'Confirm New Password:'. At the bottom left of the card is a pink 'Save Changes' button.

BackLogger All Media Search for your favorite movies, tv shows, books, music... 🔍 ☰

User Settings My Backlogs

Username:

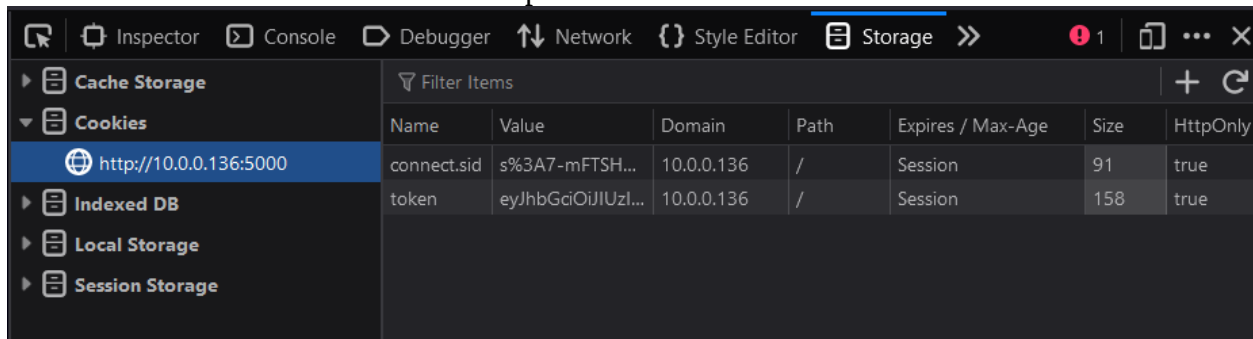
Email:

New Password:

Confirm New Password:

Save Changes

- User Cookie validation
 - When you sign in there will be a cookie that gets created and validated as you navigate pages to validate you are still signed in with the proper user. If this cookie is deleted for any reason, the user gets signed out. The item called “token” is the cookie in this example.



The screenshot shows the Chrome DevTools Storage panel. The left sidebar lists storage types: Cache Storage, Cookies, Indexed DB, Local Storage, and Session Storage. The 'Cookies' section is expanded, showing a list of cookies for the URL http://10.0.0.136:5000. The 'token' cookie is selected and highlighted in blue. The main panel displays a table of cookies with the following data:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly
connect.sid	s%3A7-mFTSH...	10.0.0.136	/	Session	91	true
token	eyJhbGciOiJIUzI...	10.0.0.136	/	Session	158	true

- When the user cookie gets deleted, the verification will detect that and properly disconnect you.

Data Structures created by Brandon Torrens and Luis Venegas

- Account
 - Name
 - First Name
 - Last Name
 - Email
 - Phone Number (Two Factor Authentication Later)
- Backlog
 - Name of Backlog
 - Specific Platforms?
 - An option would be All
 - List of options (objects of media)
- Media Item
 - Book
 - Name
 - ISBN
 - Overall Ratings
 - Year Released
 - Date/Time added to backlog list
 - List of Related Platforms (the different platforms it can be found on)
 - Current Progress (Page/Chapter/completed/not completed)
 - Movie
 - Name
 - Director
 - Overall Ratings
 - Year Released
 - Date/Time added to backlog list
 - List of Related Platforms (the different platforms it can be found on)
 - Current Progress (Completed/Not Completed)
 - TV Show
 - Name
 - Director
 - Original Producer (HBO – Game of Thrones)
 - Overall Ratings
 - Year Released

- Date/Time added to backlog list
 - List of Related Platforms (the different platforms it can be found on)
 - Current Progress (Episode/Season/completed/not completed)
- Video Game
 - Name
 - Publisher
 - Developer
 - Overall Ratings
 - Year Released
 - Date/Time added to backlog list
 - List of Related Platforms (the different platforms it can be found on)
 - Current Progress (Achievement Hunting/Completed/not completed)
- Related Platforms
 - Name of platform
 - url redirection link to specified medium (if we can get that)
 - If associated item has price display possible price/ sale of item.