

Mac App Store

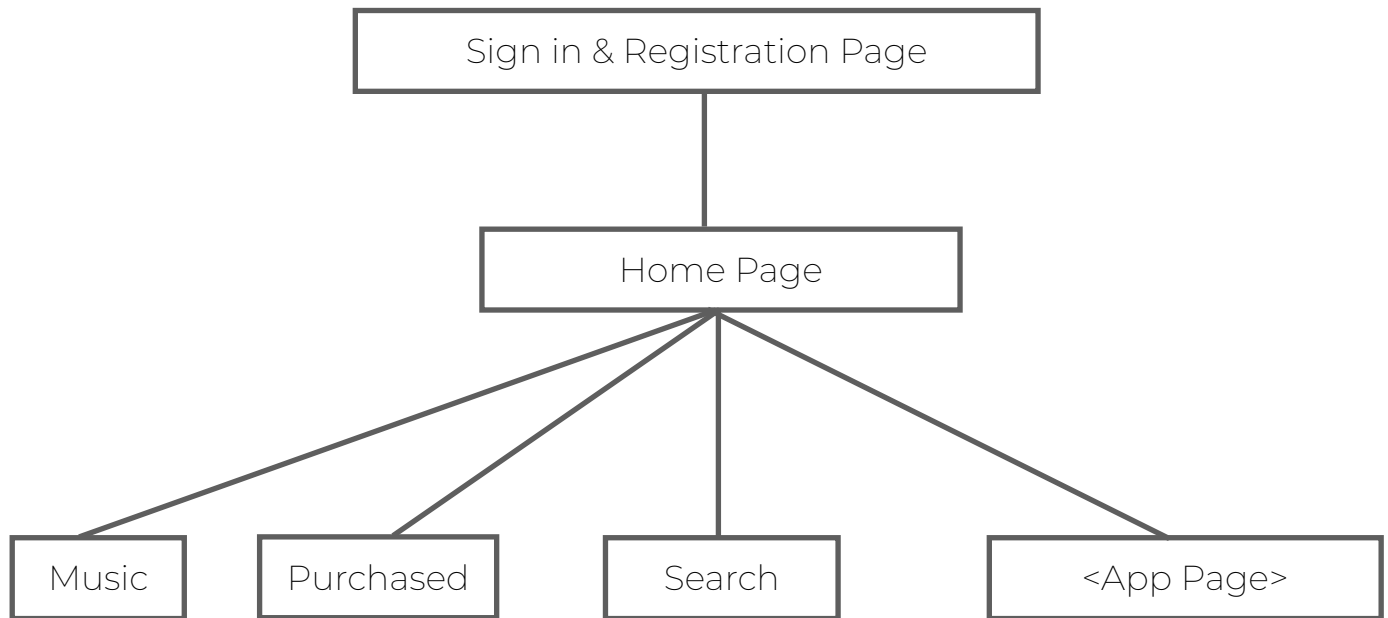
Website Developer: Bikram Chatterjee

Site Overview:

The Mac App Store website aims to be a hub for users to discover absolutely any app available on the actual Mac App Store and at the same time, serve as a place where they can find songs via a dedicated Apple Music Section. Users can also download their purchased apps from a dedicated purchased section. The Mac App Store's minimalistic design coupled with its features such featured, top charts, categories, search, and Apple Music will help users find their next app or song in no time.

Page Hierarchy and Navigation

Note: This is just the primary hierarchy. Almost all pages can be accessed from any other page in the website through the buttons in the top menu bar or app icons.

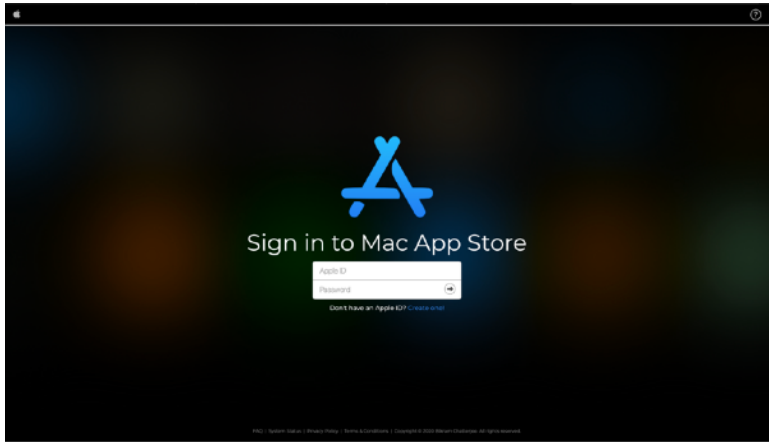


Implementation Specifications

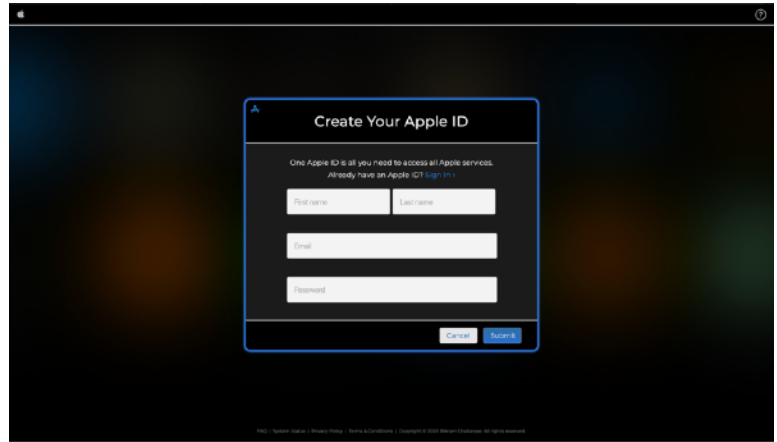
Technical Architecture:

- HTML (All Pages)
- CSS (All Pages)
- Javascript (All Pages)
- jQuery (All Pages)
- PHP. (Login/Registration Page & Logout Function)
- Local Storage (Home Page, Search Page, App Page, and Purchased Page)
- iTunes API (All Pages Except Login & Registration)

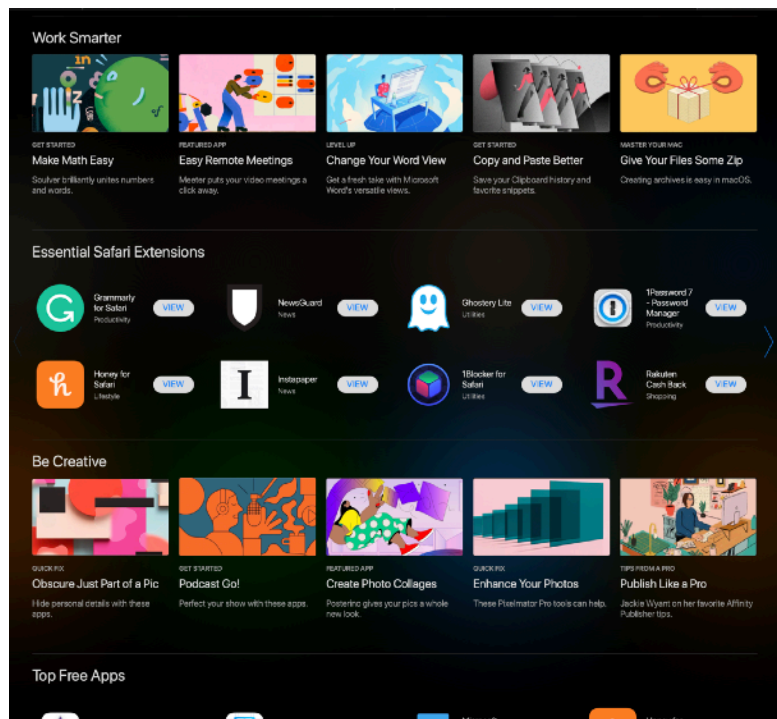
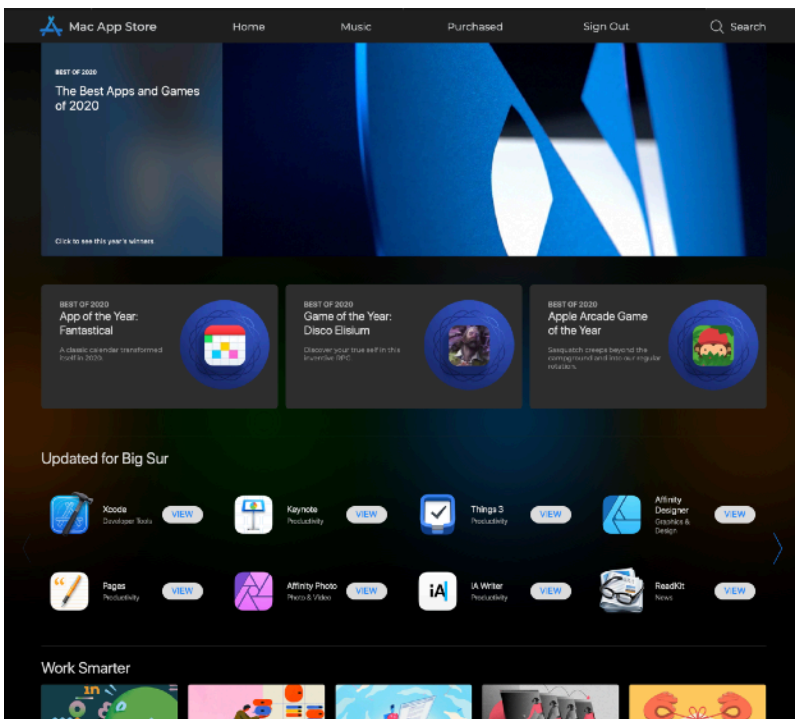
Page Design and Layout



Login Page

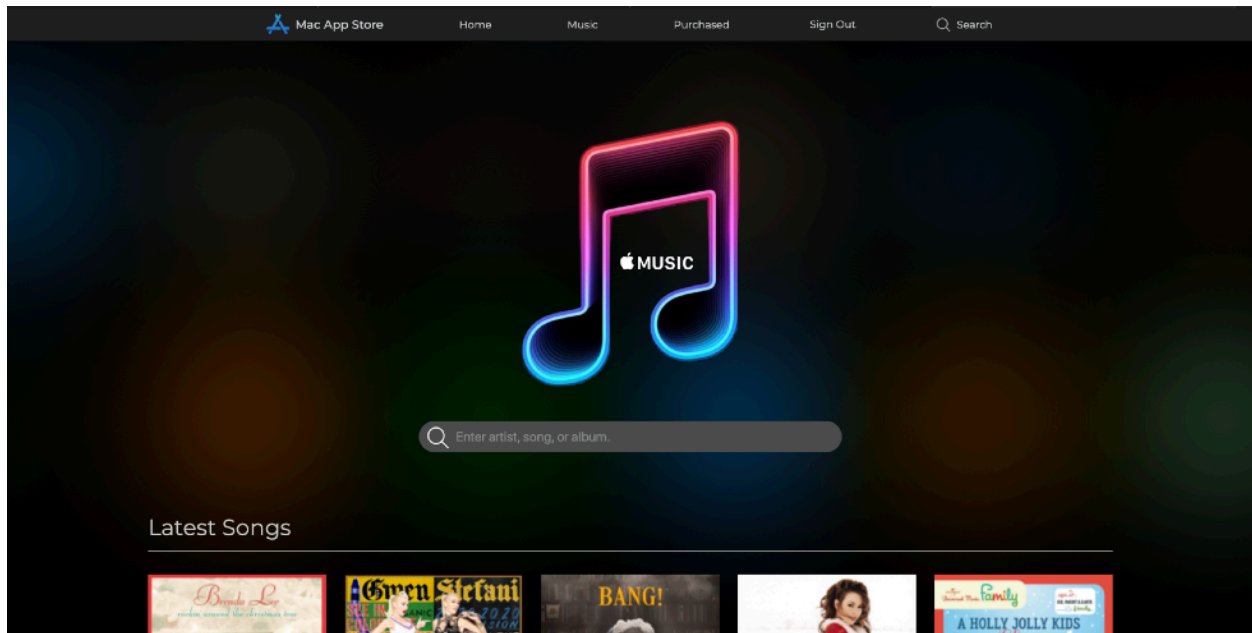


Registration Page



Home Page

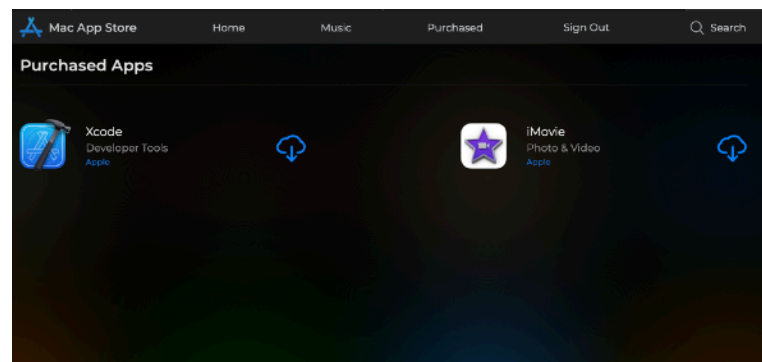
Page Design and Layout



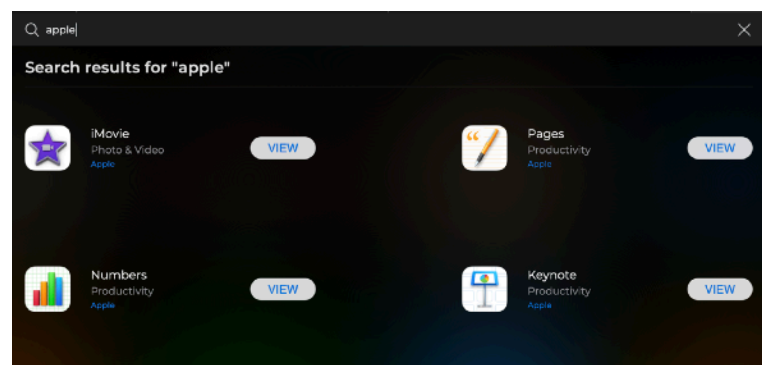
Music Page



<App Page>



Purchased Page



Search Page

Business Process/Logic Design

User Authentication (Login/Registration):

- PHP is used to store credentials in session cookies after user registers for a new account.
- Bcrypt is implemented to hash the password submitted by the user during registration and then compare the hash with the stored hash when the user wishes to sign in.
- Registration page requires a valid looking email address and a strong password in order for the user to successfully create a new account.
- User is not allowed to go back to the Sign in/Registration Page until the user is logged out. User is also not allowed to go to any other page except the login/registration unless the user is signed in.

Dynamic Webpages:

- **Login/Registration:** If the user clicks on the “Create One!” link to register a new account, a new registration window will pop up and replace the existing login body. Closing the registration window will make it disappear and show the login window again.
- **Search:** Searching for an app on the home page will clear existing content in the body and show the app (if found) with a link to the app’s page. Closing the search window will bring the user back to the previous screen.
- **App Page:** Upon clicking on any app on the website, the user will be directed to a dynamic webpage showing the description, screenshots, reviews, etc. of that particular app only.
- **Music Page:** This page dynamically changes when the user searches for music. This page also dynamically changes back and forth between itself and the search page according to the user’s actions.
- **Purchases Page:** This page dynamically changes back and forth between itself and the search page according to the user’s actions.

API/AJAX:

- The iTunes API is accessed using a AJAX method to retrieve data about all apps and songs across the website which is then inserted into html code via jQuery.

Scrolling:

- Clicking the left and right buttons in the Home Page will smoothly scroll the content left and right. The buttons will also fade and reappear according to the content's position.

Local Storage:

- Information about the app page the user wants to visit or the apps that the user has purchased is stored in the local storage and is used by various pages to retrieve that specific app data.

Implementation Timeline

Date Completed	Task
08-26-2020	Finished initial project setup
09-04-2020	Finished designing the <i>Login</i> Page and started working on the <i>Registration</i> Page
09-18-2020	Finished designing the Registration page and programmed the logic of validating input
10-10-2020	Programmed the logic of switching back and forth between red input fields and error messages when the user enters invalid input in the <i>Registration Page</i>
10-25-2020	Finished implementing password hashing, user authentication via PHP, and combined both the <i>Sign in</i> & Registration Page into one by using jQuery to make the page's content dynamically switch back and forth without reloading.
11-09-2020	Programmed the menu bar to dynamically switch between search mode and regular menu mode. Initiated programming the retrieval of data from the <i>iTunes</i> API
11-20-2020	Started designing/creating the <i>Search</i> page and finished writing a jQuery/Javascript script that retrieves data from the <i>iTunes</i> API and dynamically updates the search page
12-03-2020	Designed the App page and programmed it to dynamically retrieve data from the iTunes API by storing and retrieving app ids from the local storage. Utilized jQuery to forbid users from visiting other pages while not logged in. Finished implementing the purchased page.
12-09-2020	Designed the Home Page and programmed it to retrieve data from the API. Finished implementing the Music Page. Tested the entire website to find and fix bugs.

Summary

The biggest challenge I faced while working on this project was figuring out how to retrieve the data from the iTunes API and then dynamically insert it into the HTML code. As a new website developer, designing the website itself was extremely time consuming as I needed to align and space everything evenly, down to each and every pixel. Another challenge was figuring out the horizontal scroll in the homepage such that the left and right buttons appear or fade according to the scrollable content's position.

This project helped me learn a lot about jQuery methods and how we can store data in local storage. Furthermore, I also learned more about CSS such as how to hide the browser's scroll bar or use flexbox to space things out evenly. Dynamic content was also new and fascinating to me as it was almost magical when the website's content got replaced without any reloading at all.

In future versions, I would like to improve the music page such that it uses an external plugin to play and control song previews from iTunes. Next, I would like to implement a special section dedicated to developers that will allow them to sign up, grow their business with resources (videos or documentation) designed to help them create incredible apps, and finally distribute their apps to the users after agreeing to meet the Mac App Store's design and review guidelines. Lastly, I would also like to develop a separate dynamic category page which shows apps based on the user's selected category.