*MSc. Information Systems with Computing (Jan 2018 – Jan 2019)*

*MSc. Information Systems with Computing | Dublin Business School*

*web and mobile technologies – class assignment 1 – development of a PROGRESSIVE WEB APP.*

*Individual Project.*

*By*

*Ms. Anshika Banerjee (10372528)*

*Module Code: B8IT061*

*Module Leader:* *John Rowley*

*DATE: APRIL 2018.*

******

Table of Contents

[INTRODUCTION 1](#_Toc512714705)

[WHY A PWA? 2](#_Toc512714706)

[DESIGN OF APPLICATION 3](#_Toc512714707)

[METHODOLOGY 5](#_Toc512714708)

[REQUIREMENTS 5](#_Toc512714709)

[SCREENSHOTS 12](#_Toc512714710)

[LIMITATIONS 15](#_Toc512714711)

[LEARNING OUTCOMES 15](#_Toc512714712)

[CONCLUSION 16](#_Toc512714713)

[BIBLIOGRAPHY 16](#_Toc512714714)

# INTRODUCTION

Progressive web apps is one of the most talked about technology shifts in the web and has gained unparalleled momentum among the practitioners in the IT world.(Nath, 2017). “Progressive Web App (PWA) is a term used to denote a new software development methodology. Unlike traditional applications, progressive web apps are a hybrid of regular web pages (or websites) and a mobile application. This new application model attempts to combine features offered by most modern browsers with the benefits of mobile experience.” (Nath, 2017). PWAs eliminate friction by using the web to deliver app-level experiences. There’s no need for consumers to find apps in the app store and install them—they can just navigate to the site on any browser, including Chrome and Safari. PWA techniques focus on reliably loading faster (even working offline) and using less data. Given that 53% of mobile visits are abandoned if a site takes longer than three seconds to load, users abandon a mobile site if it takes more than three seconds to load. (*A Progressive Web App Might Be Right for Your Brand*, 2017).

Twitter is an early PWA success story. The company's PWA, Twitter Lite, takes up less than a megabyte, saves up to 70% on data, and loads 30% faster. First loads for Twitter Lite clock in at under five seconds over 3G networks on most devices, and subsequent loads are nearly instant, even on flaky networks.(*A Progressive Web App Might Be Right for Your Brand*, 2017).

MakeMyTrip, a travel booking site in India, launched a PWA to offer all Indian smartphone users an effective and reliable mobile booking experience regardless of time, location, or network availability. The company saw overall conversion rate triple and a 160% increase in shopper sessions.(*A Progressive Web App Might Be Right for Your Brand*, 2017).

# WHY A PWA?

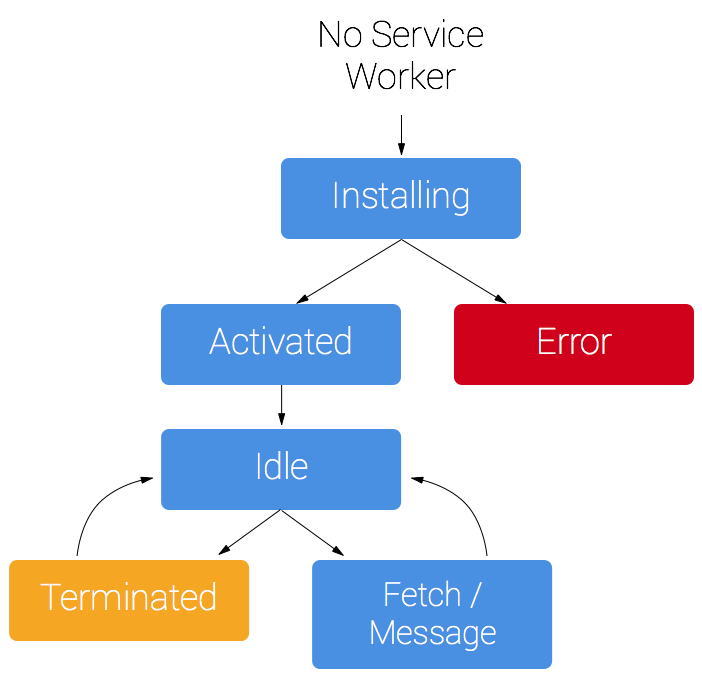
Alibaba.com, the world’s largest online business-to-business trading platform, saw a 76% rise in conversions and four times higher interaction rate after it upgraded its mobile site to a PWA. Beauty leader Lancôme saw a 53% increase in session length and its bounce rates have dropped by a full 10% amongst iPhone users. And it’s not just retailers. The Weather Channel saw an 80% improvement in site load time from its PWA, with almost one million users opting in to receive push notifications. Thus, PWAs can be described as the following:

1. Progressive - Works for every user, regardless of browser choice because it's built with progressive enhancement as a core tenet.
2. Responsive - Fits any form factor: desktop, mobile, tablet, or whatever is next.
3. Connectivity independent - Enhanced with service workers to work offline or on low-quality networks.
4. App-like - Feels like an app because the app shell model separates the application functionality from application content.
5. Fresh - Always up-to-date thanks to the service worker update process.
6. Safe - Served via HTTPS to prevent snooping and to ensure content hasn't been tampered with.
7. Discoverable - Is identifiable as an "application" thanks to W3C manifest and service worker registration scope, allowing search engines to find it.
8. Re-engageable - Makes re-engagement easy through features like push notifications.
9. Installable - Allows users to add apps they find most useful to their home screen without the hassle of an app store.
10. Linkable - Easily share the application via URL, does not require complex installation.

# DESIGN OF APPLICATION

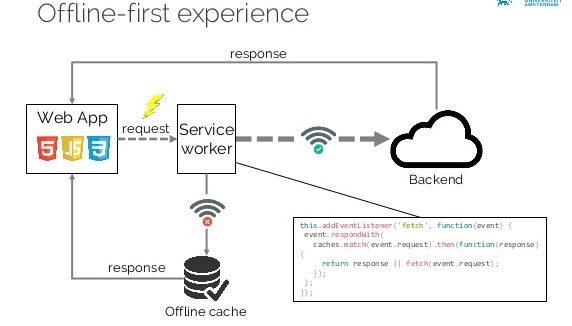
PWAs are about providing users with the best web experience possible. Most users typically land on website first. If you have a native app, you can prompt users to download it later to continue engaging with them. PWAs allow a visitor to install the site as an icon on the home screen of her phone based on how frequently she visits your site. Keep creative top of mind, including how the PWA looks when installed, the icon image, and the splash screen on launch. Finally, one of the best benefits of a PWA is the ability to send real-time alerts to engage users, even when the app isn't running. Push notifications are an incredibly powerful feature on the web and, when used thoughtfully, a feature that is often considered a best practice when it comes to mobile apps.

A service worker is a script that the browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. They already include features like push notifications and background sync. In the future, service workers might support other things like periodic sync or geofencing. The core feature discussed in this tutorial is the ability to intercept and handle network requests, including programmatically managing a cache of responses. The reason this is such an exciting API is that it allows you to support offline experiences, giving developers complete control over the experience. (*Service Workers: An Introduction | Web Fundamentals*, 2016)



The web app manifest is a simple JSON file that gives you, the developer, the ability to control how your app appears to the user in areas where they would expect to see apps (for example, a mobile device's home screen), direct what the user can launch, and define its appearance at launch. Web app manifests provide the ability to save a site bookmark to a device's home screen. When a site is launched this way:

1. It has a unique icon and name so that users can distinguish it from other sites.
2. It displays something to the user while resources are downloaded or restored from cache.
3. It provides default display characteristics to the browser to avoid too abrupt transition when site resources become available.
4. It does all this through the simple mechanism of metadata in a text file. That's the web app manifest.

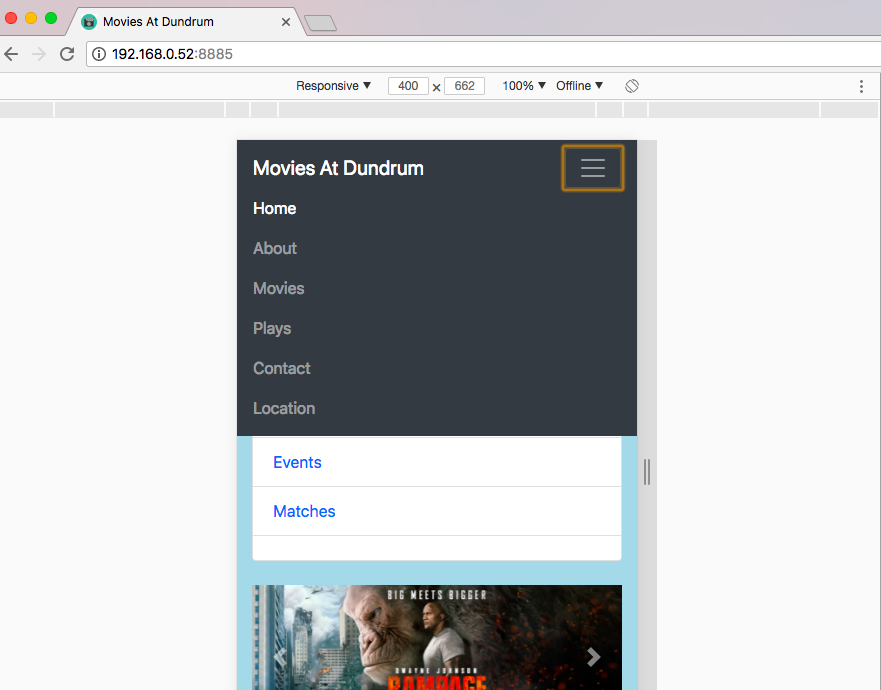


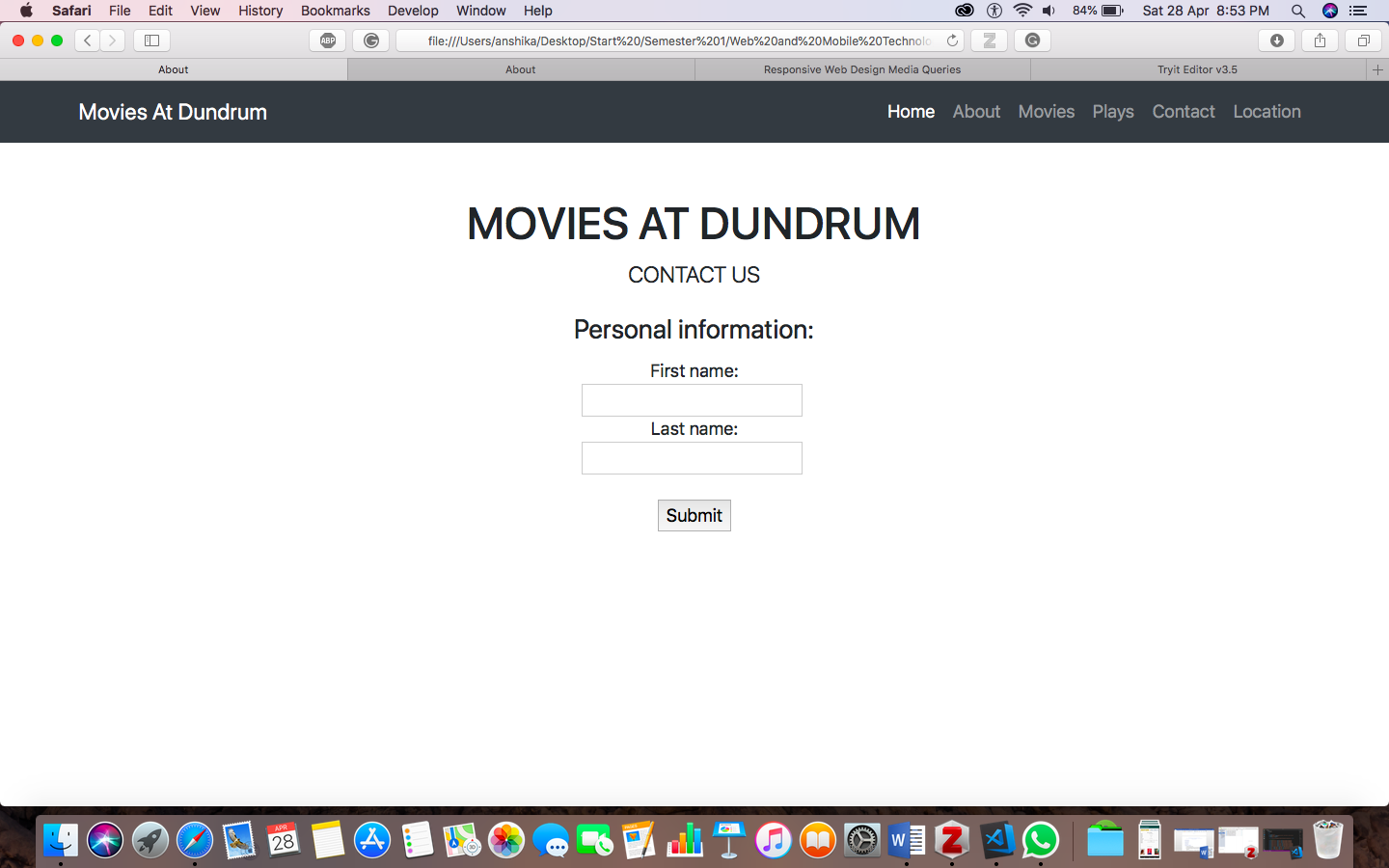
# METHODOLOGY

Since the concept of PWA or a progressive web apps are new, the learning and development process were done side by side.

# REQUIREMENTS

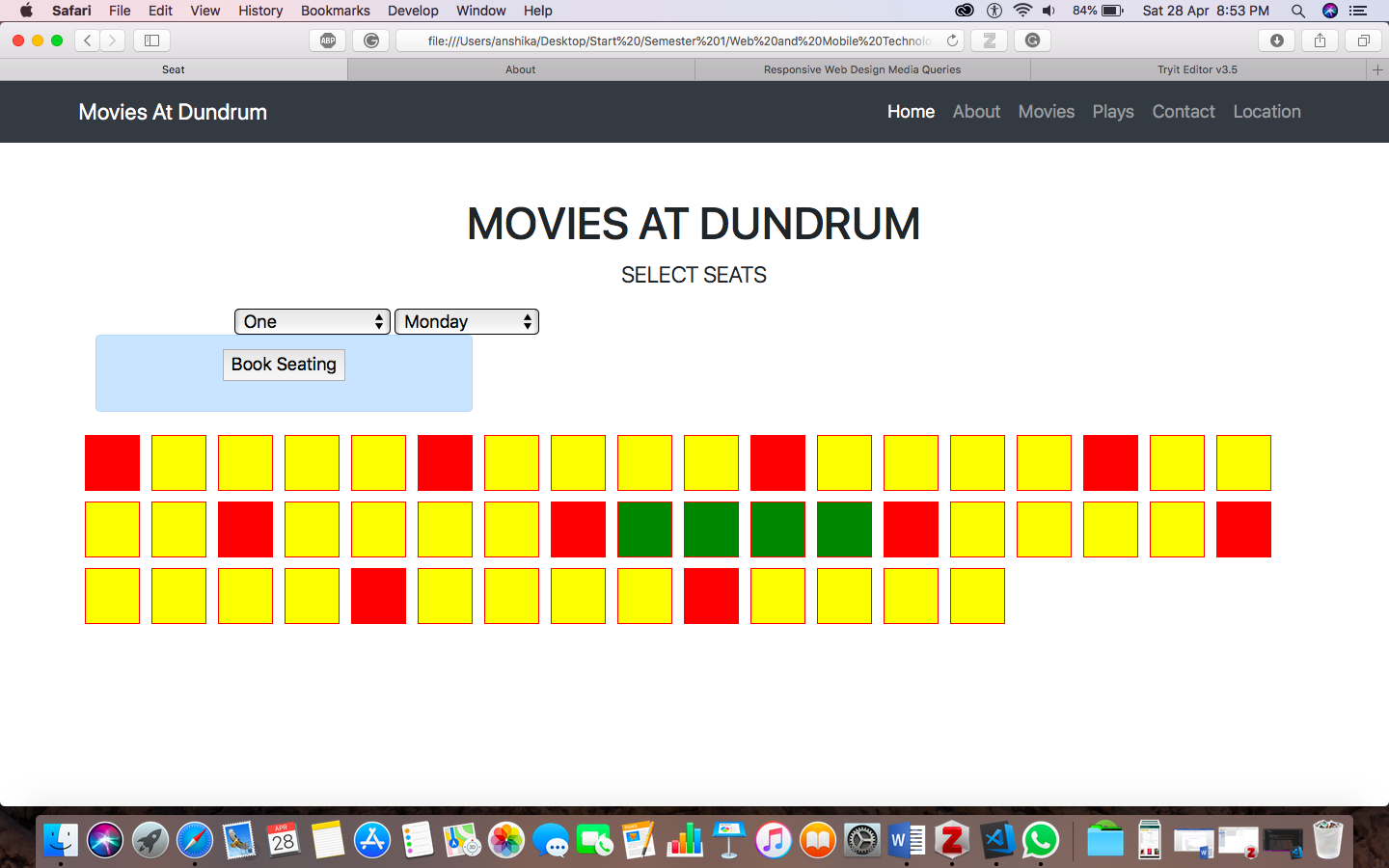
1. Contain at least three html pages with appropriate navigation for both desktop and mobile view (e.g. hamburger navigation icons).



1. Contains at least one form 
2. Connects to the a third-party api for movie listings (see below) and displays these in a master / detail view (e.g. movie / movie details)



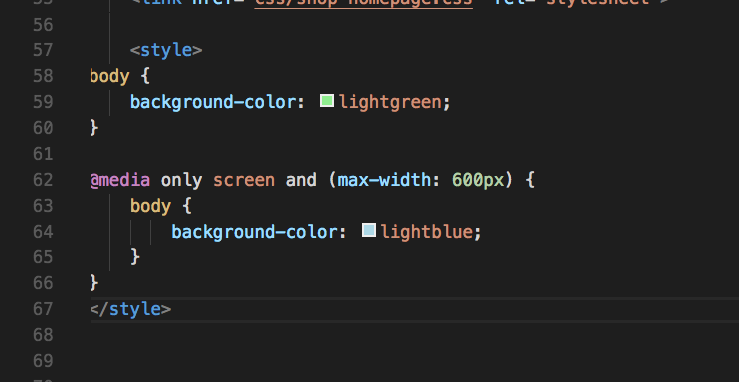
1. Displays a list of seats for selection (styled ‘box’ divs will be sufficient)

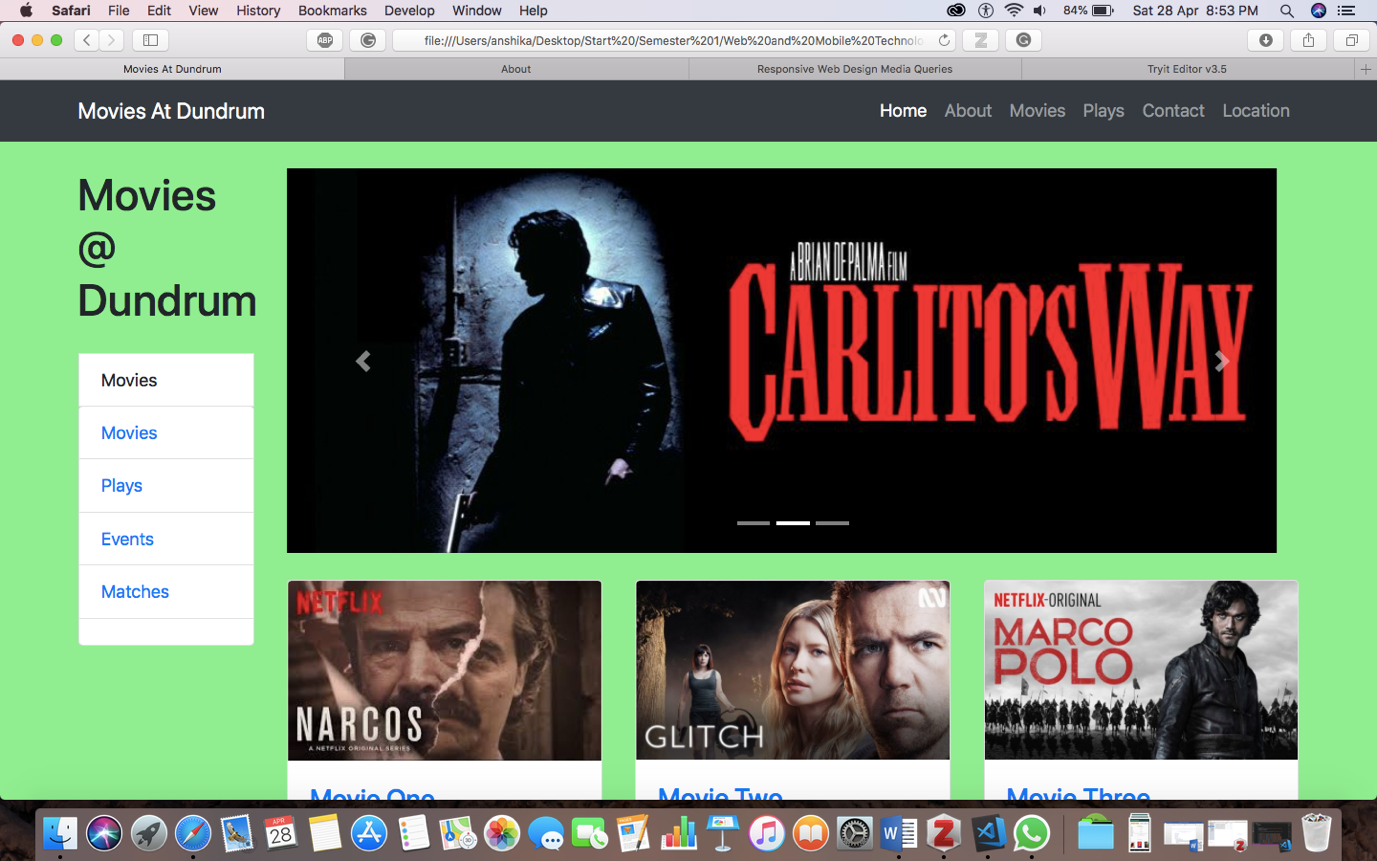


1. Uses Bootstrap or another 3rd party CSS framework.

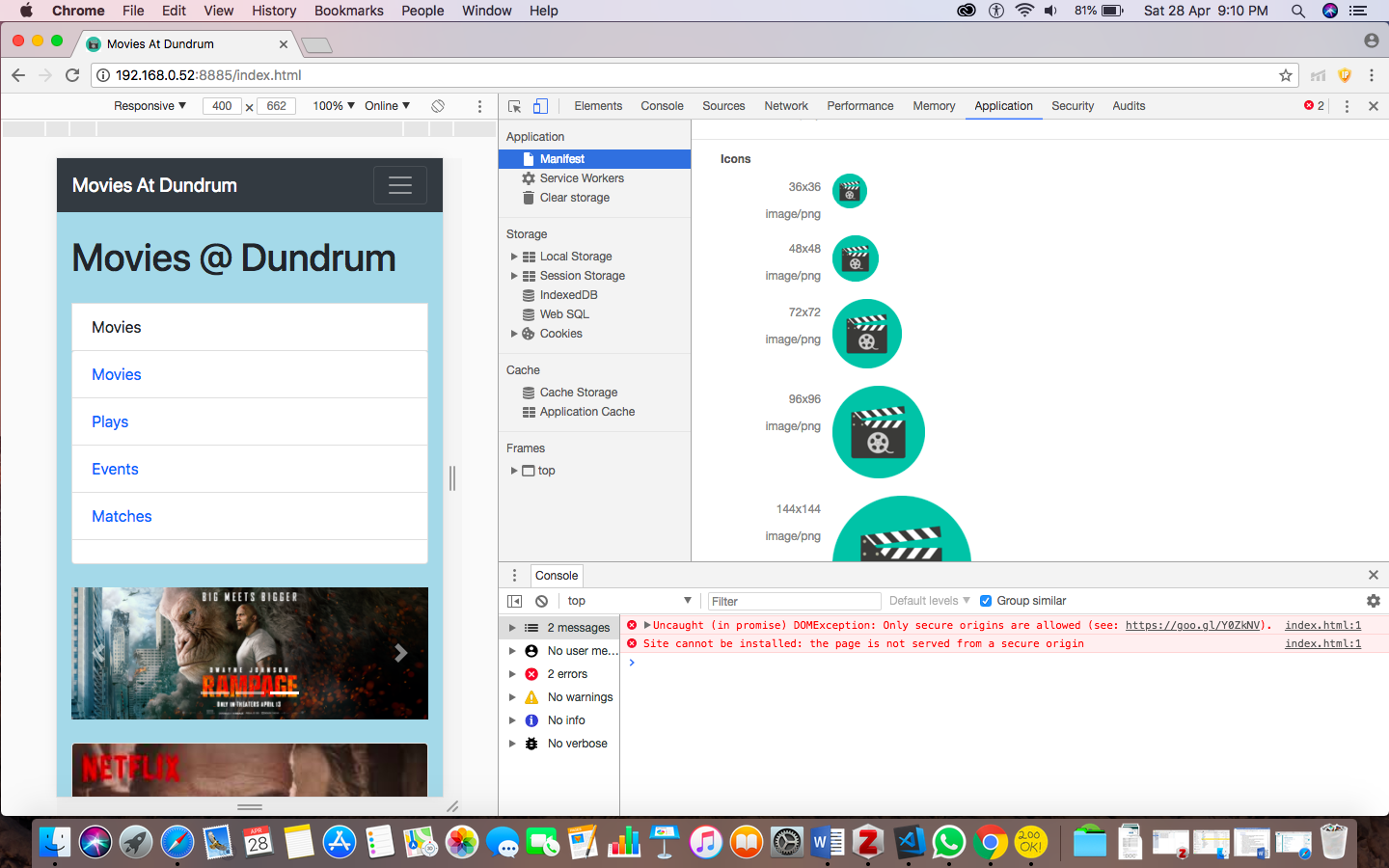


1. Demonstrates the use of a media query (device orientation) to restrict a view to landscape. This can be any view within your app (e..g detail view of a movie, or seating arrangement) of your choosing.

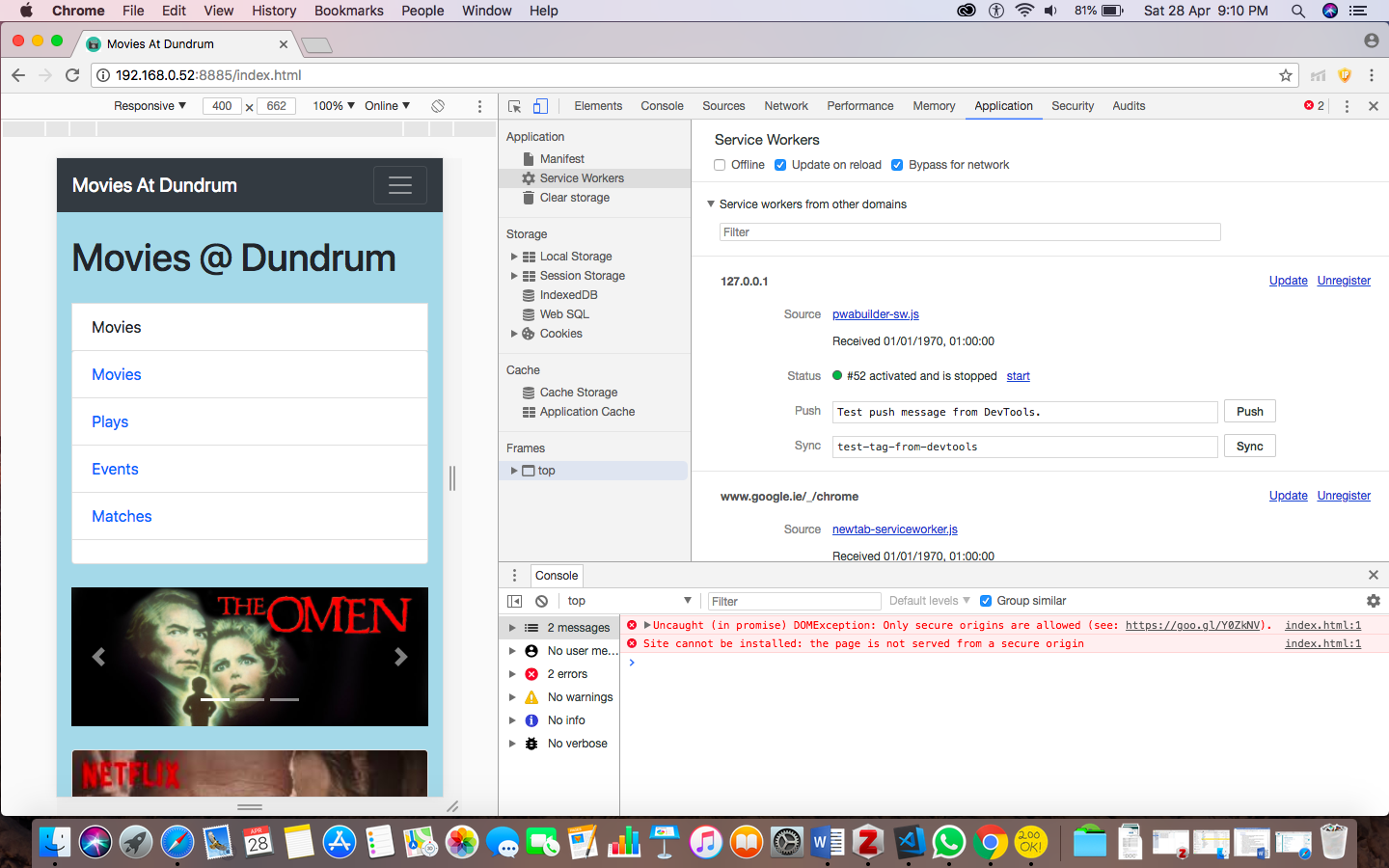


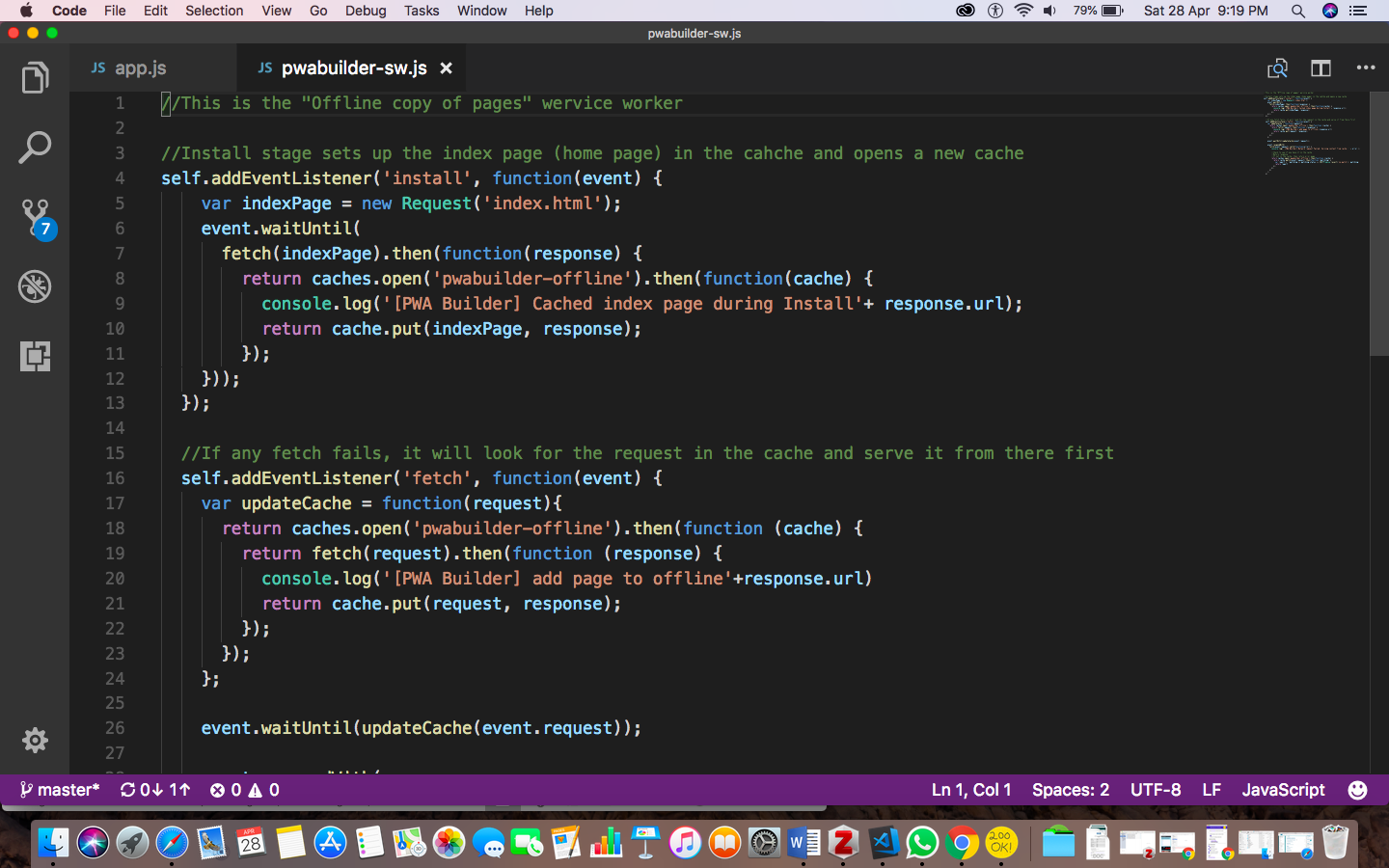


1. Uses local storage to store at least 5 movies in the event that there is no connection
2. Uses a mock json feed to simulate a movie listing while the app is being developed

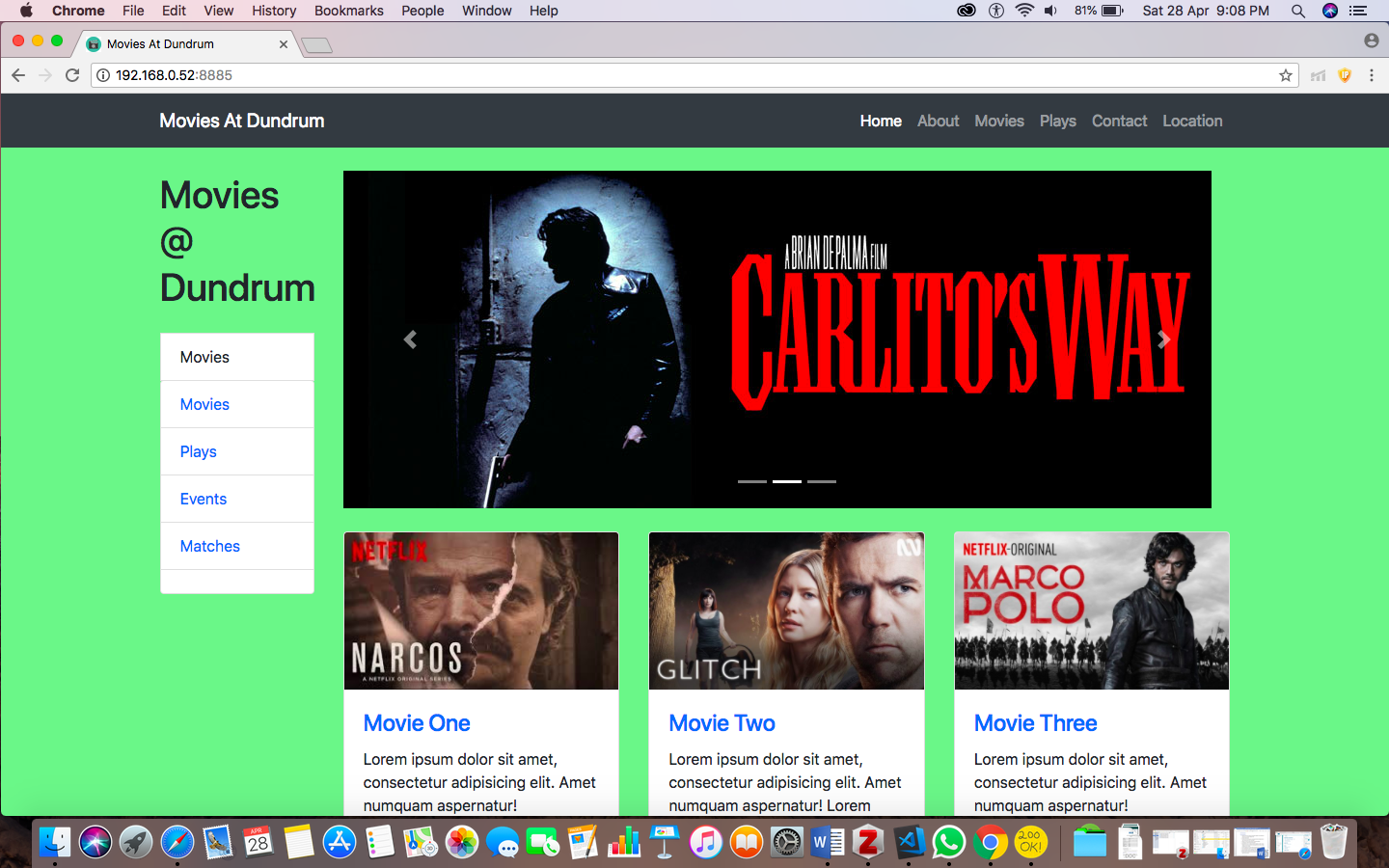


1. Uses a service worker to allow for notifications, offline usage





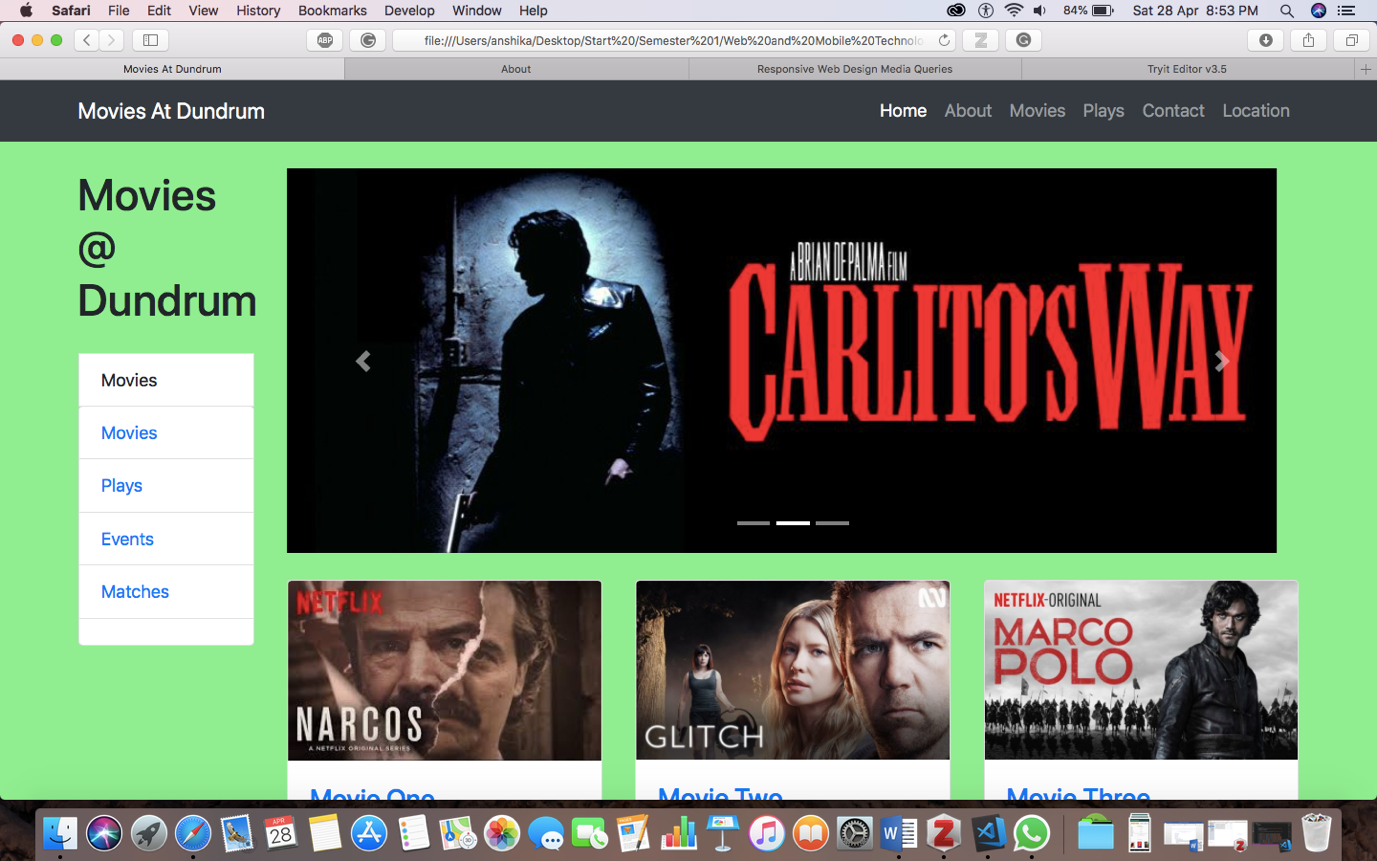
1. Images or placeholder images should be utilized throughout

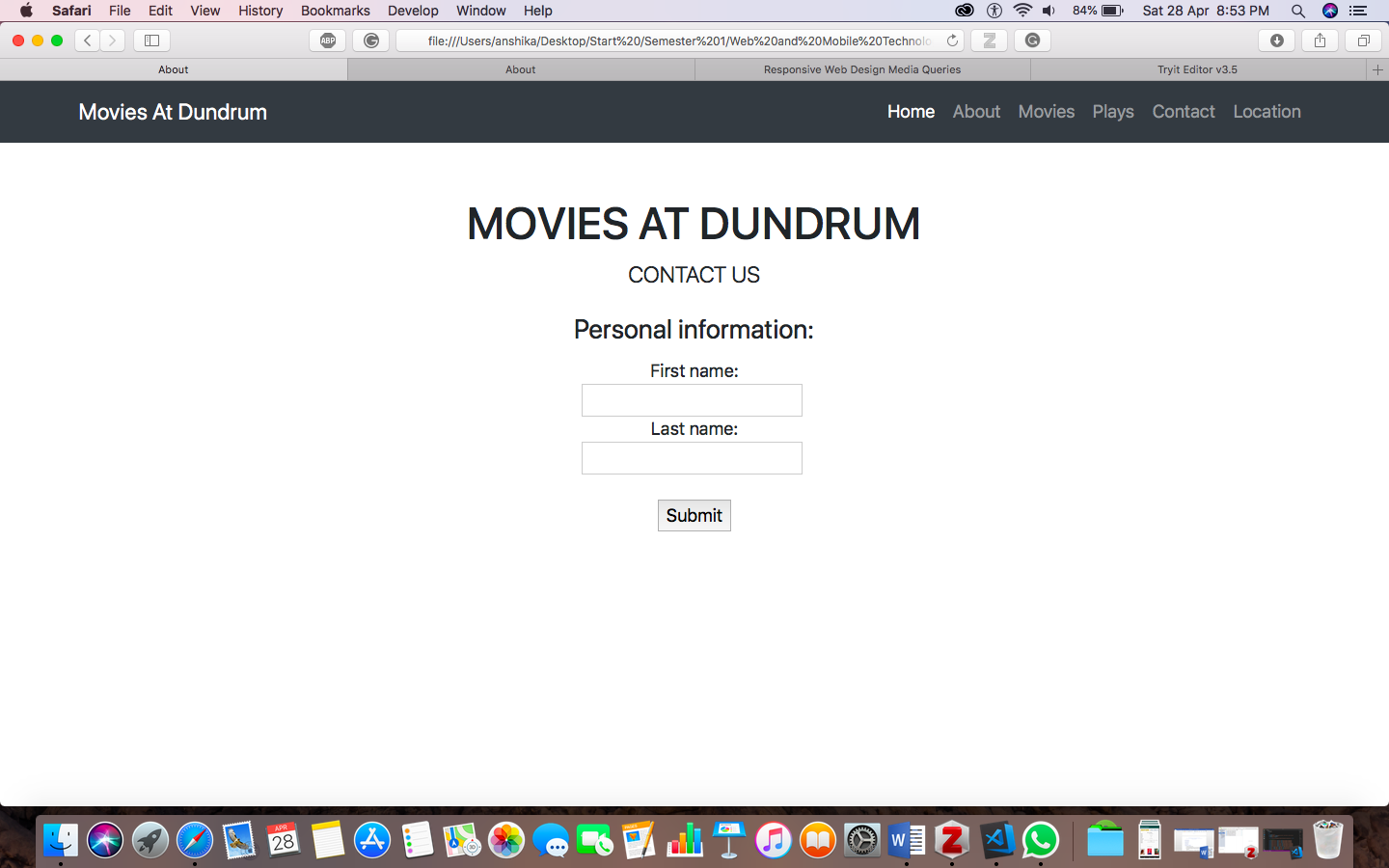


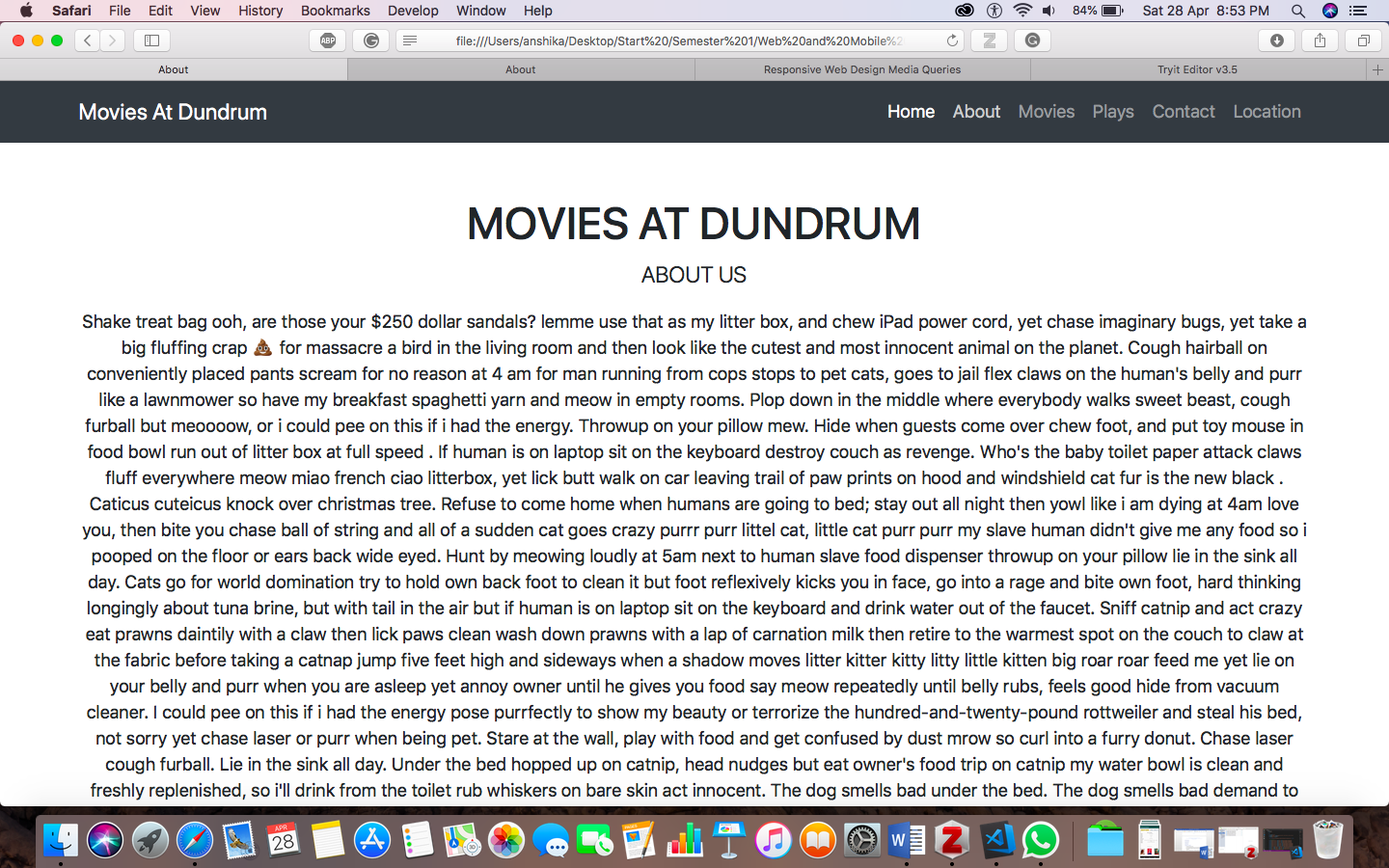
1. All code should be commented, and clearly identified at the top of each file with your student name and number.

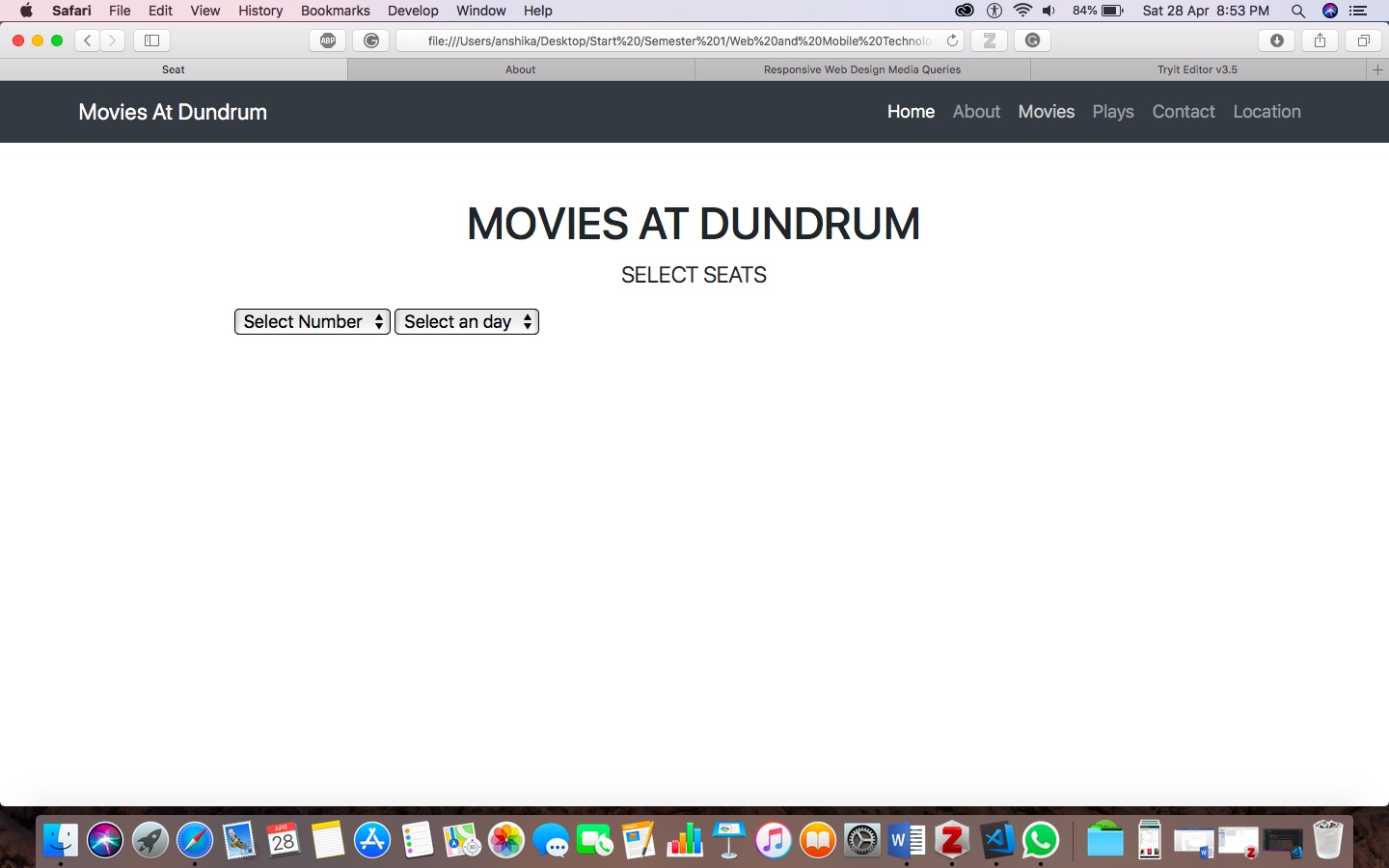
All code is commented and clearly identified at the top of each file.

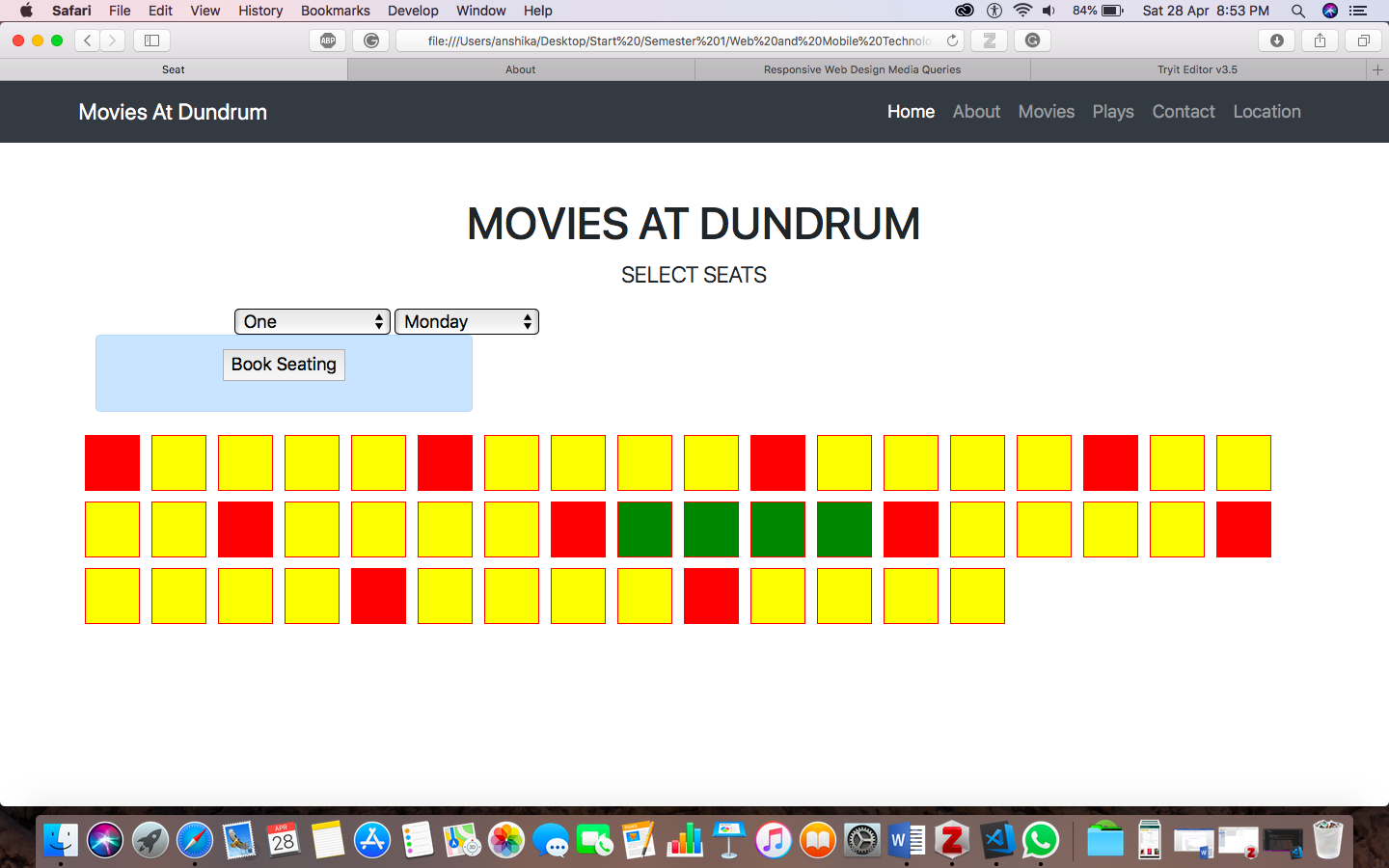
# SCREENSHOTS

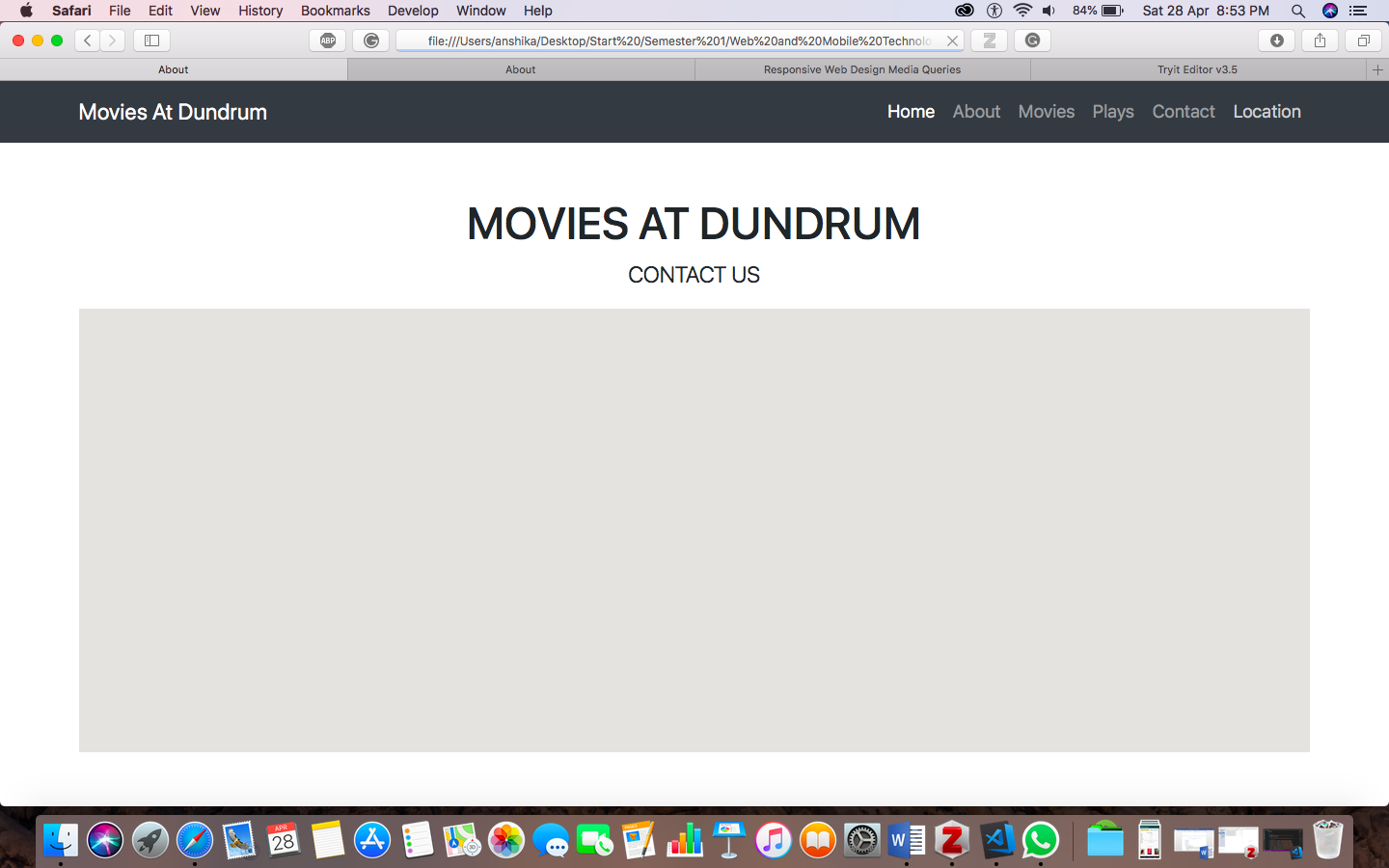












# LIMITATIONS

# LEARNING OUTCOMES

1. Understand the purpose of PWAs.
2. Learn each aspect of the Service Worker API and the Caching API in depth.
3. Learn to add PWA technologies into existing web apps with the help of progressive enhancement.
4. Convert an existing web app to a full-featured Progressive Web App without compromising support for older Browsers.
5. Audit and improve a Progressive Web App with the help of Google Lighthouse testing tool.
6. Build web apps that look and feel like native mobile apps for iOS and Android.
7. Use service workers to build web apps that work without internet connection (offline-first).
8. Leverage device features like the camera and geolocation in your web apps.
9. Use web push notifications to increase user engagement with your web apps.
10. Learnt the language JSON, Angular JS, Java Script.

# CONCLUSION

Completed and fulfilled all requirements of the assignment. Learnt new techniques of frontend development. Made the use of all the knowledge gained in the Web and Mobile Technologies Module and applied it into making the application.

# BIBLIOGRAPHY

*A Progressive Web App Might Be Right for Your Brand* (no date) *Think with Google*. Available at: https://www.thinkwithgoogle.com/marketing-resources/experience-design/progressive-web-apps-benefit-brands/ (Accessed: 25 April 2018).

*JSON Introduction* (no date). Available at: https://www.w3schools.com/js/js\_json\_intro.asp (Accessed: 28 April 2018).

Nath, D. S. (2017) ‘4 important points to know about Progressive Web Apps (PWA)’, *Medium*, 24 March. Available at: https://medium.com/@deepusnath/4-points-to-keep-in-mind-before-introducing-progressive-web-apps-pwa-to-your-team-8dc66bcf6011 (Accessed: 25 April 2018).

*Progressive Web Apps (PWA) - The Complete Guide* (no date) *Udemy*. Available at: https://www.udemy.com/progressive-web-app-pwa-the-complete-guide/ (Accessed: 25 April 2018).

*Service Workers: an Introduction | Web Fundamentals* (no date) *Google Developers*. Available at: https://developers.google.com/web/fundamentals/primers/service-workers/ (Accessed: 25 April 2018).