Title Page

Name: Seainin Keenan

Student No: D00190985

Course: Computing in Cloud Computing

Date: 17/01/2022

**Submitted in accordance with the requirements for the degree of B.Sc. (Hons) in Cloud Computing**

Copyright © 2022, Seainin Keenan

# Acknowledgements

Thank you

# Declaration

|  |
| --- |
| I hereby declare that the work described in this project is, except where otherwise stated, entirely my own work and has not been submitted as part of any degree at this or any other Institute/University.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Seainin Keenan, 2022 |

# Abstract

The main objective of this project was to create an application/website which will track new releases of a different variety of entertainment media, i.e., movies, books, games, and shows. Users will be able to add different entertainment media to their in-app calendar; The user receives a notification when their media is released, by either phone or email

Table of Contents

[Acknowledgements ii](#_Toc103205121)

[Declaration iii](#_Toc103205122)

[Abstract iv](#_Toc103205123)

[Table of Figures viii](#_Toc103205124)

[Introduction 9](#_Toc103205125)

[Aims 9](#_Toc103205126)

[Objectives 9](#_Toc103205127)

[Literature Review 10](#_Toc103205128)

[Existing Applications 10](#_Toc103205129)

[Hosting Platforms 10](#_Toc103205130)

[Amazon Web Services 10](#_Toc103205131)

[Google 10](#_Toc103205132)

[Azure 11](#_Toc103205133)

[Technologies 11](#_Toc103205134)

[What kind of app will be developed? 12](#_Toc103205135)

[Programming Platforms 12](#_Toc103205136)

[Programming Languages 13](#_Toc103205137)

[Web Scraper 13](#_Toc103205138)

[Design 15](#_Toc103205139)

[Brief Overview of Project 15](#_Toc103205140)

[APP 15](#_Toc103205141)

[Login 16](#_Toc103205142)

[Signup 16](#_Toc103205143)

[Feed 16](#_Toc103205144)

[Forum 16](#_Toc103205145)

[Messages 16](#_Toc103205146)

[Profile 16](#_Toc103205147)

[Sidenav 16](#_Toc103205148)

[Chat 16](#_Toc103205149)

[Forum 16](#_Toc103205150)

[MediaApi/Feed 16](#_Toc103205151)

[Firestore 16](#_Toc103205152)

[Notifications 16](#_Toc103205153)

[API 16](#_Toc103205154)

[Models 16](#_Toc103205155)

[Data 16](#_Toc103205156)

[Controller 16](#_Toc103205157)

[Microsoft Packages 16](#_Toc103205158)

[Other Packages 17](#_Toc103205159)

[Azure SQL Database 17](#_Toc103205160)

[Webscaper 17](#_Toc103205161)

[Push Notifications 17](#_Toc103205162)

[Implementation-how I used it physically in the code 19](#_Toc103205163)

[API Development 19](#_Toc103205164)

[Timeline-remember why or what made me do it this way 19](#_Toc103205165)

[Future Work 21](#_Toc103205166)

[Observations 22](#_Toc103205167)

[APP Development 22](#_Toc103205168)

[Timeline 22](#_Toc103205169)

[Future Work 24](#_Toc103205170)

[Observations 24](#_Toc103205171)

[Web Scraper Development 24](#_Toc103205172)

[Timeline 24](#_Toc103205173)

[Future Work 26](#_Toc103205174)

[Observations 26](#_Toc103205175)

[Conclusions 28](#_Toc103205176)

[Source Code 29](#_Toc103205177)

# Table of Figures

[Figure 1 the numpty 7](#_Toc95301044)

# Introduction

This project looks at designing and implementing an application that can be used as a centralized point to find release dates for moves, games, and music. First, users will be able to create an account on the ionic mobile application. Once logged in, they will be able to browse a list of movies, games, shows, and books released or will be released soon. Then, the Media API supplies the mobile application with data for the media release dates. Finally, a web scraper fills and adds the media data to a SQL database connected to the media API.

## Aims

1. Design a mobile application that will allow users to access entertainment release dates.
2. Implement the application backend on a cloud platform.
3. Design a database to maintain user and media information.
4. Use a web scraper to find release dates automatically on the web.
5. Implement push notifications.

## Objectives

1. Understand the tools and applications used to develop mobile applications
2. Research cloud platforms and the different services provided, i.e., SQL database, lambda functions, virtual machines, and notification services.
3. Research database options such as SQL and NoSQL
4. Research the operation of web scrapers to find information on the Internet.

# Literature Review

## Existing Applications

Are not any mobile applications to keep track of different media releases. However, multiple websites show when specific media releases, i.e., movie releases or game releases. They had the information but no straightforward method to keep track of them. They can be tracked manually on a calendar app. There are multiple websites to find information on media releases, including a list. All have their way of showing the data. For example, games and books have a higher quantity of releases compared to movies and shows. So, the information is there, but there could be an easier way to follow these media, mainly if someone follows multiple media, it can be hard to keep track of them.

## Hosting Platforms

### Amazon Web Services

Amazon has a lot of simple services it provides, including Simple Notification Service (SNS), Simple Queue Service (SQS), Simple Storage Service(S3), and Elastic Compute Cloud (EC2) instances. All these services can be used independently from one other. However, they require some knowledge to set up and can get confusing. Some of the services I might use include SNS, Relational Database Service (RDS), EC2, and Lambda. SNS is used to push notifications to other services in different formats. These formats include text and email. SNS has clients subscribe to a topic that can then publish/send messages to all clients who subscribe to that topic. RDS is an amazon used to set up different databases, including Microsoft SQL Server, MySQL, aurora, and PostgreSQL. The Media API will store all information in an SQL database, which amazon could provide. EC2 is used to create virtual machines in the cloud. There are many possibilities when using virtual machines, including hosting websites/services, and could benefit any project that works with the cloud. Finally, Lambda provides a service to run code without needing a server.

### Google

Google also provides cloud services, like AWS. Firebase is a platform provided by Google to help create and develop mobile and web applications. They offer both a real-time and a Firestore database. A real-time database is Firebase's original database. It is low latency compared to the Firestore Database. The Firestore database is a newer database that uses faster queries and can scale better than the real-time database (Firebase Documentation, 2022). The Firebase SDK/software development kit can be used with multiply languages like C++, Java, and JavaScript. Firebase is widely used with JavaScript frames like angular and react

### Azure

Microsoft provides Azure's web services. Azure provides SQL databases both on a server and serverless like the other services. Although Microsoft also developed Visual Studio (VS). Azure web services are easily integrated and used in VS applications. In addition, VS provides much help when deploying services, making interaction with VS convenient.

## Technologies

Ionic is an open-source software development kit used to develop hybrid apps, i.e., web applications that are put into a native app shell. They help web designers create fully functional mobile applications by allowing the developer to use HTML, CSS, and JavaScript frameworks to create applications and use mobile functionality (Ionic, 2022). Ionic was founded in 2012 by Ben Sperry and Max Lynch. Ionic sits in a Cordova WebView. Cordova allows Ionic to use mobile device functions, i.e., contacts, and emails. Cordova was developed by Apache (Griffith, 2022)

Angular is a JavaScript framework used to create web applications. It is used as a front-end framework. Helps to create web pages with functionality. It is written mostly in typescript. Angular was developed by Google on September 14, 2016.

Bootstrap structures web pages and allow websites to be responsive. However, Bootstrap can cause trouble with ionic. They both manipulate the DOM. The Bootstrap node package allows Ionic to use Bootstrap. It uses HTML and CSS to do this. Bootstrap splits a webpage into 12 columns, and each HTML element can be formatted using these columns. Bootstrap also supplies more effortless scalability in terms of screen size, i.e., changing the look of the webpage on computer screens and mobile devices

Node Js executes JavaScript code on the server-side. However, it also runs on the front end of the web application. Node Js computes requests on a single thread instead of using a pool that can run out of connections in the connection pool. Node Js handles requests on a single thread. It can handle thousands of requests. (Capan, 2022)

.Net Core is used to build web apps and services in C# and other languages. .Net Core is the latest version of the .Net framework, a software development framework to build and run applications on windows. Microsoft developed the .Net framework

Selenium is a tool used to perform automated tasks on a browser, including testing and automation on a web browser. Many web scrapers use selenium to execute a web page rather than download the web page HTML. Because it can execute the JavaScript on a page, it provides all the availability to a program that a person sees when browsing a webpage. Selenium provides a chrome extension and selenium libraries for different programming languages. A web driver is needed to use selenium on a local machine. the web driver interacts directly with a browser and uses the browsers engine to control the webpage. (tutorialpoint, 2022)

Git is a version control system. It helps keep track of files & folders and stores them in the cloud. Git handles any size project, from enterprise projects to single files. Git was released in April 2005. It makes use of a bash command line to control different resources. GitHub and GitLab are two internet hosts that make use of git.

## What kind of app will be developed?

Ionic provides the capability to create apps on IOS, Android, and the web. Each platform required minor changes to be compatible with ionic. This app will be developed for android as android is more popular than IOS (Statista Research Department, 2022), although there are benefits to developing with both. This app is a utility app with a few social media options. It provides the user with a reminder of when a media is released.

## Programming Platforms

Visual Studio, Visual Studio Code, and Replit are all environments that are used to develop applications/products/code. In addition, they help create a wide range of projects in different languages.

Visual Studio is an Integrated Development Environment used to develop websites, web apps, and APIs. Microsoft developed visual Studio and integrated its cloud services into the IDE, making deploying an application created in Visual Studio easier and integrating Azure services into the applications.

Visual Studio Code is a code editor. It provides an area where code is written and edited, and it supports multiply different coding languages, with the ability to add extensions, which can help format code to a designated format or add an animated cat to the code-editor

Replit is an online IDE based in San Francisco. Supports multiply languages including **C, C#, Machine Assembly, Node Js, and Python.** Replit can be used to develop applications on the web, suited for weaker laptops, or if the developer moves around a lot, can save CPU and ram. Although Replit does not provide private projects by default, there is a payment option to increase CPU and ram limits, more storage, and make projects private. The free tier provides the user with the tools to create projects, including importing libraries, some drivers, and compiling code in the browser.

## Programming Languages

JavaScript allows developers to create front-end websites. It allows web pages to be more interactive while browsing. A high-level programming language is a language that can be easily read by humans and is abstract from the computer running it (Beal, 2022).

C# can be used with .Net core to create web applications and APIs. It is also a high-level programming language.

Typescript allows developers to add type support to JavaScript, creating classes and objects. Some JavaScript frameworks are written in typescript, including Angular and Vue, while react supports it. Typescript simplifies JavaScript code to read and modify more easily.

HTML is a language used to design displays for websites in a browser. CSS assists in formatting HTML pages. They care about a web page's design and what the user sees when interacting with a web page.

JSON is a format used to store data. It helps store objects in a key-value pair and lists of different objects. It can be easily read and written by humans. JSON stands for JavaScript object notation. It can be beneficial for sending data across HTTP

Python is a high-level, general-purpose programming language. It can be lightweight but not used for mobile or game dev because slow. Python was developed to be easily read by humans (geeksforgeeks, 2022).

SQL stands for Structured Query Language. It is used to query databases. It is capable of other processes such as stored procedures which automatically perform SQL queries on a database

## Web Scraper

Web scrapers are used to get/scrape data from websites, getting the data from html elements or browsing webpages.

Web scrapers are used to get/scrape data from websites by getting the data from HTML elements or browsing webpages.

Creating a web scraper is C++ with gumbo or using python and selenium to interpret dynamic web pages. Basic formatted of a python web scraper would be. First, import libraries into python file, then initiate selenium driver and add parameters, then load the webpage and query the elements of the webpage.

# Design

### Brief Overview of Project

The client will have a mobile application in which they can create an account and log in. The main page will be a feed that will display a list of different types of newly released or upcoming media, including movies, games, books, and shows. The user will then add the media to their in-app calendar. The user will also be able to message other users and review media. The mobile application will store its user data in a Firestore database and other information specific to the user, i.e., the calendar entries. Users will be notified when an upcoming media is released via text or email.

The mobile application will get the media data from an API (Application Protocol Interface). This API will store its data in a SQL database on the cloud. The API will perform CRUD (Create, Read, Update, Delete) on the data in the SQL database. The API will have multiple endpoints, conform to the REST API criteria, be stateless, and be designed using the model view controller pattern. Splitting all the data in models and used by their controller/endpoints to improve the readability of the code.

A web scraper will fill the SQL database with media. The web scraper will use python to scrape specific websites and store the data in the SQL database. The libraries used will be selenium, pyodbc, UUID, web driver manager, and Date Time, and the drivers used are chrome web driver and ODBC Driver 18 for SQL Server. pyodbc python library allows python to communicate with an MSSQL database using the ODBC Driver for SQL Server. The web driver’s manager library downloads the required web driver for selenium when the python script is run. UUID Library creates universally unique identifiers, like the guide, globally unique identifiers.

## APP

The mobile application will be written in angular, HTML, CSS, and Ionic. The mobile application will communicate with the mobile using Cordova. The following pages will be added to the mobile application:

#### Login

Allow a user to login if they already have an account, use fire auth. Html page that will give the user an option to log in with their email and password, or sign up if they don’t have an account. Users will be stored in a Firestore database. This will allow the mobile application keep track of user specific data i.e. using the user data to keep track of followed media

#### Signup

Allow a user to create an account that they will be able to sign up with which will be stored in a Firestore database. An email, username and password are required to make an account

#### Feed

The application was originally going to have a separate page to list each media based on type i.e., movies, games, shows and books. The application will instead make use of a single feed page that will display a list of all media. This should make the application more engaging by decreasing the number of pages the user needs to use. The user will also have the option to search by media type as well as search by media name, date, theme, genre or actor.

The media API will supply the media entries for the media app. When a request is made to the API, the API will return a list of media ordered by release date.

If a user wants more information on a media, they will be able to click on a media, they will be brought to another page which will have more information on the media

#### Forum

The forums page will have a forum for each media in the MediaAPI which should be automatically generated. The user will be able to create posts on forums. The user will also be able to comment on posts.

#### Messages

The messages page will allow the user to message other users. The user will be able to search for other users and open a chat page with another user. The user will be able to send messages and receive message from other user. Message will be stored in a Firestore database along with the sender and receiver. The user will also have an option to block other users.

#### Profile

The profile page will allow users to create a profile. The profile will including a username a profile picture, a short description and a list of medias they like/follow.

#### Sidenav

The application will make use of a Sidenav to organise all the pages into an easy accessible side menu which will be displayed on every page to allow the user to access all other pages

The following services will be added to the mobile application:

#### Chat

This service will be used by the messages page. Allow user to add data to database. Get data from database and delete data if they are the owner/creator of it.

#### Forum

Similar to chat service. The forum service will be used by the forum page. The forum service with add post and comment data, get post and comment data and delete poat and comment data by owner/creator

#### MediaApi/Feed

Service to allow app to use the MediaAPI. Allow feed app to create dynamic requests to get specific information from the API including actors, themes,genres,sellers and creating properties of medias. These requests will return a list of medias associated with the above parameters

#### Firestore

Allow users to use Firestore to save login and signup detail. The login and sign up page will make use of this service

#### Notifications

Send api request to mediaApi to add users email,number and the media they wish to follow. This data will be saved a SQL table. The media topic arn will also be save into the table.

## API

There will be many different parts of this media API. There are not as clear to split as the media App. But here is a list of the different components of the Media API

### Model

Text

Description automatically generatedWhat are models and how they will be used in this project

The models are the objects that the API uses to structure and hold the data passing from the SQL database to the mobile application.

This is an example of one of the media model. Each model will used as a template fro how the data will be stored in the SQL database.

### Interface

Text

Description automatically generatedThe Data will be how the media API gets the information from any source; in this case it will be a SQL database. But what if I wanted to get the data from some other type of database or some other area? To make the media more reusable, I create a interface for each controller used in this API and a SQL connection class that makes use of the interface. An interface allows developers to create methods with all the information that they require i.e. The parameter that the method needs to function and the return type.

Other classes will be able to use these interfaces, and create the methods how they would like to be implemented while still returning the required information. In the case of this API the controller uses these methods when it needs data for a response to a API request.

### Controller

Text

Description automatically generatedControllers to create api endpoints, what plan

Adding images

### Data

Text

Description automatically generatedThe SQLData classes are classes that interact with the SQL database. They use the Interfaces so that there is a standard way of getting the information required by the controller regardless of how a class is created. A scope is added into the start up c# script, to let the controller know which class implementing the data interface the program should use.

### Microsoft Packages

#### Entity Framework

Tools sql server,design

What each package provides

Allow models to be used to create database

Add notation so database can understand

Migrations adding to models

### Other Package

System.data.sqlclient,

What package is used for and where I used it, provided the functions to interact with the database

## Azure SQL Database

### Models/schems

All models including series/creatingpartry/genre/theme/persons/userreviews/usersub

Db schema

Diagram

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

### Stored procedures

What I will use. Sort sql, delete replicas, delete old sql

Storedprocedure pic

Graphical user interface, text, application

Description automatically generated

### Triggers

What triggers stored procedures

Example of triggers

Graphical user interface, text, application

Description automatically generated

## Web Scraper

### Load Website

Create driver using selenium and webdriver

Load webpage

### Parse Website

Get list of information and parce data

### Add to database

Create connection add data in one go or for each

### Optional

Send user information added

### Cloud

Stored on aws lambda

## Push Notifications

Simple Notification Service (SNS) sends messages via email or text; This will notify the users when a media they follow has been released. The SNS will be run in an AWS lambda function using python. The libraries used were pyodbc, time, and boto3. Boto3 is a python library that allows developers to use AWS services in their python code. In this case, it would be used to set up the SNS. each media will have its topic. Every time a user follows a media, they send an API request to the media API with their email, number, and the media they wish to follow. The API will then add the email and number into a table with the topic Amazon Resource Name (ARN) of the game they wish to follow. The SQL table will be called 'userSub.' The lambda function will be triggered daily. It will subscribe to all users in the 'userSub' table with the topic that matches their ARN if they are not yet subscribers. Then the function will get all media from the database where the date is today's date. There topic ARN will be used to publish a message to all subscribers, letting them know their followed media has been released. Then all users in the ‘userSub’ that match that topic will be deleted to clear up the table

# Implementation

Step by step, even or especially when things don’t work, it demonstrates how you spent your time.

## API Development

Main focus, most time spent

### Timeline-remember why or what made me do it this way

Create model/models, form on what information I wanted in my api

Create data interfaces

Create controller for separate model,added crud to media controller

Create dummy data to get from in a class.

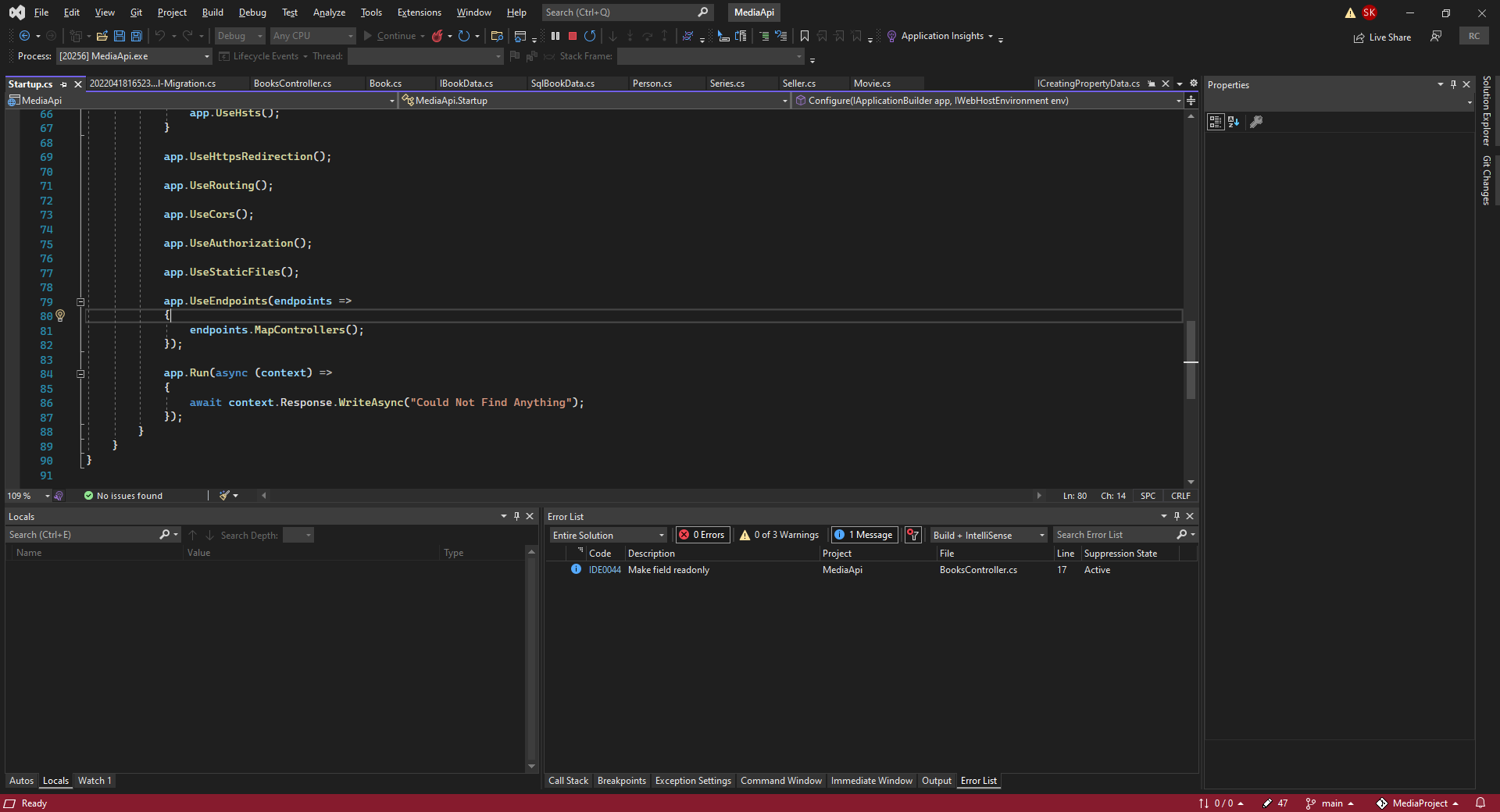
Had a problem with my personal computer, wasn’t able to use it for a 2 weeks. This happened near the start of development so wasn’t too big of a problem

Text

Description automatically generatedStart again with better results-changed what I thought might need to be changed, because I had just written it before was easy enough to remember plus I hadn’t done too much up to that point.

Implement Entity frameworks and classes, downloaded Microsoft entity frameworks. Use this to create table in database. Had to add notations so models translated to database tables [key],[required]

Adding migrations, problem with models as all medias were children of media class. Wouldn’t work with ef.i tried looking up a solution but couldn’t find one. Endned up seperateing each child of class media into their own separate class, didn’t like this but I wanted to make it work. Later on in development found out you could add children classes to database



A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Problem connecting to apache database, wasn’t sure what format the connection string had to be but then found azure provided the string similar to wha tive used before. Also had to add dbcontext in startup.cs and add scope for each interface and sql class pair for controllers to work

Another problem used apache to test with mysql, use mysql package with api to crate and connect to database. Tried to connect with azure mssql didn’t realize why it wouldn’t work, finally coped on, realize that there are different sql database

Create sql database in azure, set up database was simple enough as I had done it before. Just bit of setup. Called server mysql because I didn’t realize there was a differnece

Graphical user interface, text

Description automatically generated

When I tried to connect was rejected because my ip adrees wasn’t accept. Figure out how to add ip address to security firewall

Start using git and continue to used for version control, what I learned here, little bit of experience.

Computer fixed, so started using it again, same SSD so no real trouble

Compatibility issues between computer and laptop net5.0 on laptop net6.0 on computer. Not sure how this happened. I think I tried to update my old project using git with my new project . when I got pack to my laptop it wouldn’t work,told me I needed a newer version orto change the project version.Updated laptop

Added migration and updated database

A screenshot of a computer

Description automatically generated with medium confidence

Use postman to add and get models from api. Graphical user interface, text, application, email

Description automatically generated Tested out the api which help create and redesign the controllers and data interfaces and data sql classes. Used methods provide by data context instead of raw sql queries.learned more about http requests and how they are formed. Postman give a lot of information. Form-data to add multiple bits of data

Used entity framework to add not mapped lists, which would later be used to fill the list in the api. These lists stored lists of objects. I had an object called series which would have a list of medias. I wasn’t able to stoe the list in the sql database so I found you could use [NotMapped] to ignore the list field A screenshot of a computer

Description automatically generated with medium confidencewhen adding migrations. I used it with medias as well as each media has a list of themes and genres and actors associated with them.

Reworked season models from series-season-episode to series-shows-season-epsodes. Now had movies,books,shows and games where all apart of series. Is a bit canfusing as I changed the meaning of series in the context of the api.

Used post man to add dummy data to api so there would be data to test mobile app. So all the data in media models where filled. Which turned out different when using the webscraper. Graphical user interface, text, application, email

Description automatically generated

I had a problem with the mobile app and api communicating due to cors. After looking for a solution and finding that it could be solved by adding app.useCors() to the startup.cs file. Explain cors in app section A screenshot of a computer

Description automatically generated with medium confidence

Change api each controller to accept files/wwwroot folder Using json form. Watched tutorial on adding images. Using IWebHostEnviroment, A screenshot of a computer

Description automatically generated with medium confidence which save the images in a folder in the project. They can be accessed on the web when the api is deployed. A screenshot of a computer

Description automatically generated with medium confidence

Added one to many tables to connect medias with lists of people, themes and genres in sql database, although only themes is used atm as not all data was gotten from the websites scraped(should have built sraper first, why did I make the api first. Because I knew how to)Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceUpdate models to store date as date in models and database instead of as string. Removed dummy data when I created and got the webscraper working. Used A screenshot of a computer

Description automatically generated with medium confidencekept reading that I should never store dates as strings in a sql database. Also thought I would have a problem in the future if I tried to sort/order by date.

Updated models to store extra data that could be gotten from the webscraper.deleted dummy data/decided not to use api to add to sql database and add straight to database. Would have required a lot of repetitive classes . Better to try another methods to show I can do more? Not sure what I would be marked on so I tried use as much technologies that I could, which has negatively affected the project. No part is fully developed and only does the bare minimum.

Added feed api to get multiple medias with one api call. A screenshot of a computer

Description automatically generated with medium confidence Merge all medias(because the database already order then in there individual tables by date) ordered by date when feed endpoint was requested. Thought this methods was better than sorting them while in an api response request. Cut down time

### Future Work

#### Biggest issues

Another problem was that the mobile app has to wait around 10 seconds for the data to be sent back. Will have to fix this in the future. Which is kind of ridiculous

Not all tables are used yet

Not all endpoint have proper information returned/status codes

Why I think these problems happen and why I haven’t fixed them yet.

#### Other work

Plan to add api keys, shouldn’t be too hard and how I plan to implement them, use user id generated by firebase? No because then other people wont be able to use the api. How about send api request with your email, use sns to automatically send you an api key associate with your email address, if you confirm with that email. An api key would be good if I wanted to monitor how individually user(anoumously, don’t need to know who they are) use the api(data which could be used to improve the api). Could also use api keys to restrict amount of requests so I amnit charged a furtune if somebody makes thousands of requests.

Refactor api to be similier to <https://the-one-api.dev/v2>. Really liked the design of the api and how one request can be formatted different ways, using different search perametters.

Deployed to azure using visual studios allows to add dbsets

### Observations

Why create api before webscraper? Different data found than system was created for had to refactor. Bit off more than I could chew. Added functionality that I never used because it would require a lot of time to used right, more complicated webscraper to get all the data I wanted, not impossible just more time than I had to complete this project again bit off more than I could chew. Although it will be easier to scale the project as some aspects are already in place, but are they in the right place?

## APP Development

Took backseat because I though other part were more important and that the app was fickle in design, could put a lot of work in and decide to change it, which I did do

### Timeline

Generated application. Generated using ionic, I tried using different templates including sidebar and tab. Deleted and restarted as I kept running into problems when I change some aspect of the routing. Just deleted them instead of figuring out the problem

Generated tabs. Decide to used this template. Have a tab for each media\*\*\*media list page for each type of media\*\*\*\*. This was later change to a feed that would get all medias with a parameter to search specific media

Added pages for the different medias. At the start this was all I thought I needed. And a calendar. Didn’t do much after that

Had a problem with my personal computer, wasn’t able to use it for a 2 weeks. This happened near the start of development so wasn’t too big of a problem. Was able to use a laptop in the mean time

Computer fixed.

Compatibility issue between laptop and computer. Laptop had up to date ionic while computer had older version(from december). Used version of computer on both. So downgraded on laptop to ionic 5.

Redone app to use feed idea, what is it instead of a page for each media

Added calendar,details-media,forums,login,sighup,media/feed,messages,profile,sidenav. A screenshot of a computer

Description automatically generated with medium confidence Added a place holder for each of these pages. Calendar would have a calendar using a module specifically for android. Details-media was to bring up more detail on a media when clicked on. As of writing this thought of a solution? Feed api gets less information so it is faster pagination\*\*\*need to look into that\*\*\*? ,might be faster??Just title, image, release date and media id\*\*\*possible solution\*\*\*. Then when loading details-media use a more specific api call to get the media data. Would cut down wait time.

Created a login and signup page utilizing the firestore database. I added the firebase api information into environment.ts A screenshot of a computer

Description automatically generated with medium confidenceand added the AngularFiremodule. A screenshot of a computer

Description automatically generatedCreated Firestore service to login in sign up and get user information. Sign up = the information email,password and generated uid is added to a firestore database. Then a user could sign up and login successfully.

A screenshot of a computer

Description automatically generated with medium confidence Text

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

forums and messages were place holders which were going to be used to message other users on the app and create posts on different forums

Problems with sidenav. Graphical user interface

Description automatically generatedNew releases of ionic quickly, hard to find solution on ionic version I am using. Different methods to navigate in ionic between ionic 4 5 and 6. So it was hard to find a tutorial for ionic 5. decided to move on to more important parts. Still having trouble with it. Tried using mwthods from ionic 4 and 6. And watch many tutorials and I could only get it t work once and only once. Think it is something to do with the folder placement or the sidenav only loading once as all pages are loaded inside sidenav. Can figure out solution. Low priority because

I tried adding the media api I create as a service in ionic but I coulnt get it working so I moved the request to the feed page. The problem wasn’t in the service it was with how is used the observable. Haven’t movies the service back yet but could see how that could be useful A screenshot of a computer

Description automatically generated

When I got the api connected I then ran into another problem. Which was cors. Cross-origin resource sharing is a mechanism that restricts http requests from scripts. From what I understand it send a request to an api, the api returns the response and the domain is not the same as the request the response is rejected

I tried setting up requests that would get rid of the error but they didn’t seem to work. I Ended finding a solution to fix this cors problem on the api instead.

Added functionality on media.page.html to show data received from the api. A screenshot of a computer

Description automatically generated with medium confidenceAlso added the ability to get more details by saving the tapped media in a state and bringing that state to another page where more information was show as well as more medias in the same series as the current media

A screenshot of a computer

Description automatically generated

A computer screen capture

Description automatically generated with medium confidenceI tried using different if statements in the html to sort the list of data into different media types. As each media had different bits of information, this didn’t matter too much as the information wont show up if doesn’t have data.inforamtion wanted on the feed page. The problem came when I tried to dislay the iamages. The page failed to load as the image usrls where store using just their name i.e. picture.jpeg. I added the url in the ionic app. Each image for each image was stored in a different folder in the wwroot file so I had to check what type the media was to figure out the folder its image was stored in. could have prevented this by adding the whole url straight into the database, would have been a simple fix why didn’t I do this. The user interface didn’t look too good after I spent a good amount of time developing it and didn’t look to pretty in the code either so I deleted it and decide to use bootstrap.

Later found out that ionic and bootstrap don’t work well together as both of them using the DOM in different ways. But found out after after that ionic has a bootstrap package

### Future Work

#### Biggest Issues

Add messages,forum,user reviews api keys per account, better css as app doesn’t look pleasant at all.

Make better use of Firestore and user experience

#### Other work

Add calendar functionality to keep track of media added to calendar. Just so the user could see them. Plan to add soon but wasn’t completed by the time of this document

Allow users to submit correction to media data. Either have someone manually check and add correct information or cross examine submits of the same media see if they can add if they are all the same???

\*\*\*Figure out how to use sns with specific users. Went through few ideas. Decide to use python again, as I had created the webscraper by this point. Add table to database which stores mobile number and media id. Run python script daily if media date in today create topic and get all mobile numbers or email using media id. Send text to all with details without subscriping first as text messages can be sent without subscribing \*\*\*

### Observations

I found ionic to be useful in that it has a lot to modules that can be used to enchance to application but I have always found css,html and a lot of ionic to be tedious. Typescript can be hard to understand by just looking at it. Although I can seem why it is well used as it can be extremely powerful when used properly.

## Web Scraper Development

Turned out the best as was the smallest prt but a very important piece

### Timeline

Watch a tutorial about using a webscraper with python and selenium

Use replit to create webscraper using python. As I didn’t want to install all the python,pi and dependencies locally especially when the final webscraper would be running on a aws lambda.

Imported selenium with webdrivers, these are automatically installed with replit so I didn’t have to worry about setting up the selenium webdriver.

Tested getting a page, and printed the results, after getting the webpages information, I then started looking for websites that had the realease dates of movies,books,games and shows, I also wanted them to be reoccurring as if they weren’t it would kind of defeat the reason of this webscraper. So if they were reoccurring I could scrape then once a day and have up too date data in my sql database. I got two or three websites from each media. Text

Description automatically generated

I tried out all on the movie website metacritc first before implementing it with other media i.e. repeat with books, shows, games. After setting up the driver with chrome using selenium I was able to get the pageA screenshot of a computer

Description automatically generated. Then used the find\_elements(By.TAG\_NAME,’tr’). This would get every tag,and everything in that tag and return a list. The tutorial I was following was using a find\_elements method that was deprecated so I had to search up the current way to usethe method, luckily it hadn’t changed much. I have never formolly learned python only in small past project so I wasn’t familiar with all the syntax. I did watch a few basic tutorials so I would understand what I was doing a bit better. Someof the syntax used I the webscraper tutorial I didn’t understand at all, but by the time I finished the webscraper I understood exactly what I was doing. Although I do think there are glaring gaps in my usage of python.

To parse the medias I created a function for each media. Parse\_movie\_metacritic would take in one moviei.e. element from the webpage and return a python object I could use.the webscraper tutorial returned a data structure I didn’t understand and python threw no exceptions. Weird that python didn’t need me to manually initiate object as return method otr oject so culdnt find out what it was. Found out it was called a dictionary store string key object value pair. Parse\_movie\_metacritic threw exception because each second tr was empty with other empy one randomly. Had problems to figure out how to handle exceptions try except­A screenshot of a computer

Description automatically generated with medium confidence

Had 2 loops one went through list of trs and removed empty tags. Then other loop parse the data into a list of dictionarys. Refactor using if is not None(learning new things) split time in half especially when all web pages where queriedA screenshot of a computer

Description automatically generated with medium confidence

Turned list of dictioary into csv file using panda dataframe. A screenshot of a computer

Description automatically generated with medium confidence

Some movies didn’t have release dates, so I didn’t want to use them, added another if statement to handle if the dictionary[date] had None

I ran into this problem with both shows and games websites.

Then I had to researched how to add the data to an sql database.

Replit doesn’t support pyodbc as it needs a sql driver

Set up python and pip,on my computer and laptop,Download odbc driver for sql server. Now have to configure selenium on pc. Install webdriver manager to install driver during runtimeA screenshot of a computer

Description automatically generated with medium confidence

Little problem writing add\_to\_db methods. With sanitization and dictionary(which is what I was adding to the data base) being null

Created method to change string date to type date so it would be stored properly in the database. Methods can change multiple date string formats into suitable date formatA screenshot of a computer

Description automatically generated with medium confidence

Problem with null fields and unique ids for mediaId. Imported uuid<> which is the same as guid<>. Learned what these both do<detail>.

A computer screen capture

Description automatically generated with medium confidenceThe method is inside a loop so for each element add to database. A lot of sql queries should try do less. Could have let the add\_to\_db method take in a list and all the data to be inserted together and let the sql database handle it as one sql statement. However I didn’t want to set up a second loop to handles the now full set(no null columns) data. Would have increased the runtime by a good amount according to the big O(n) this shouldn’t matter. Lambda has a 15-minute limit so wanted to keep the time down especially if I want to add more to this webscrpare, although I could just create another separate webscraper.

Created stored procedure to delete duplicates if title release date and description are the same. Little bit complicated. What if media changes date? Or two different media by the same name ae coming out??

Created triggers for stored procedure, to trigger when insert into media/movies/shows/games/books. (if so then stored procedure is triggered every time there is a request, will need to refactor if I want to be more effecient) Graphical user interface, text, application

Description automatically generated

### Future Work

Code doesn’t look neat and hard to read/ refator or redesign

Add profanity filter

Make another wenscraper to search for trailers for medias in sql and add them to sql databaseand find information to fill the rest of the database

Add another stored procedure to order all medias by release date. Faster than order every request.

Deploy webscrpaer on aws lambda(steps to take)

### Observations

Learned a lot about python and enjoyed it

## Push Notifications

Created prototype, haven’t added in yet

Used aws rds sql server database

Connect using python pyodbc

Successfully added to lambda

Demo gets information fro db where media has game release date and topic arn

User sub as an email a number and a topic arn, both email and number will be used but can be left empty.

Goes through games to check if game date is todays date, if so publish message to all subscribers that message is released today

### Future Work

Add information to azure database,

Allow user in mobile app decide if they want to be reminded when a media is releasing 1 day before 3 days before and or 7 days before. All this information to be added to database and python will easily be able to format the right stringto be published based on the users preferancce.

# Conclusions

The application …

# Source Code

|  |  |
| --- | --- |
|  |  |