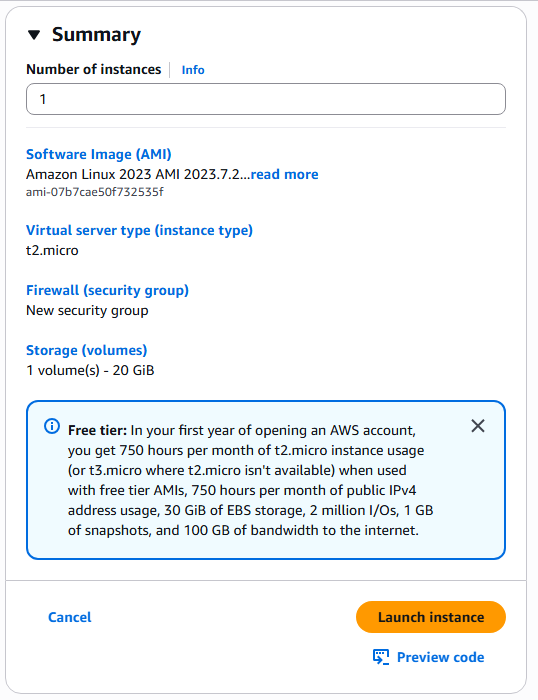
LAUNCH EC2 INSTANCE

1. In EC2 find instances and click “Launch instances”.
2. Select Ubuntu 24.04 LTS for the operating system
3. Instance type is t2.micro
4. Create new key pair
5. Create security group, click “edit” and add a rule, select http for the type
6. Then create another rule in the same security group but this time select https for the type
7. Add a 20GiB SSD
8. Check that the instance is in the free tier it should look like this, then click “Launch instance”



GET STATIC IPV4 ADDRESS

1. Go to EC2 dashboard
2. From there navigate to Elastic Ips
3. Select your newly created instance
4. Then click Allocate Elastic IP address
5. Click Allocate

GET DNS ADDRESS AND CONFIGURE

1. On namecheap purchase your domain name
2. Go to Domain list, then Advanced DNS
3. Add new host record, type A record, and enter the EC2 Static IP Address

SET UP SSH TO SERVER  
I find that creating a .bat file that automatically logs in to my ec2 instance is very convenient and can save time.

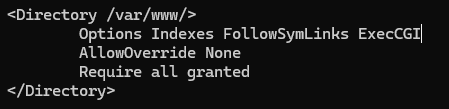
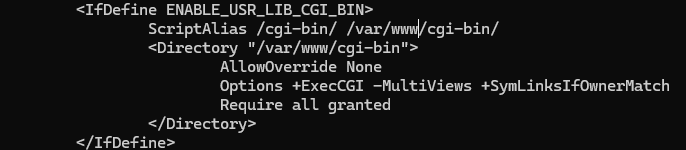
1. Enable OpenSSH in the windows features settings if OpenSSH is not already installed
2. In a new text file write “ssh -i ‘file\_name\_of\_your\_key\_pair’ ‘username@ec2\_instance\_ip\_address’”
3. Save as a .bat file ideally in the same folder as your key pair
4. **Setting Key Pair Permissions**
   1. Go to the key pair file -> properties -> security -> advanced
   2. Click disable inheritance and then convert inherited permissions to explicit permissions on this object
   3. At the second line of the window change owner to your account
   4. Finally, remove all permission entries except for the ones for your account
   5. Apply changes and then click ok
5. This file can now be run and will automatically open terminal and log into your ec2 instance

GET AND CONFIGURE APACHE2

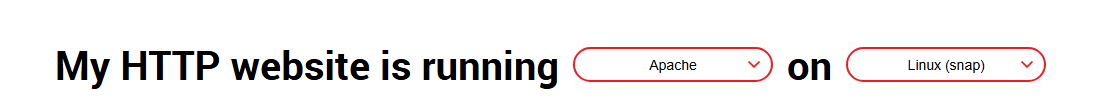
1. First use command “sudo apt update”
2. Then use “sudo apt install apache2”
3. Check if website is live in browser
4. If not use “sudo systemctl start apache2”

CONFIGURE CGI

1. Use the command “sudo a2enmod cgi” to enable cgi module
2. In etc/apache2/mods-available/mime.conf uncomment AddHandler cgi-script .cgi so it looks like thisA screenshot of a computer program

   AI-generated content may be incorrect.
3. Then in etc/apache2/apache2.conf scroll down to Directory /var/www/> and add ExecCGI to the end of the line below so it looks like this
4. In var/www/html make a new directory called cgi-bin and put your cgi files there
5. Then in etc/apache2/conf-available/serve-cgi-bin.conf change the bottom to look like this
6. Then use command “systemctl restart apache2”

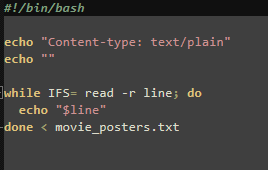
GET DIGITAL CERTIFICATE

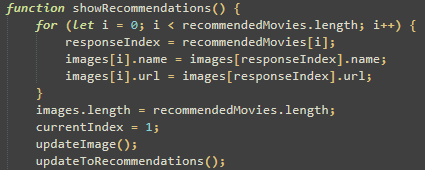
1. Go to certbot.eff.org
2. Select Apache and Linux (snap) like in the image below
3. Follow the instructions to get the domain on https

INSTALL PYTHON3 AND NUMPY FOR SCRIPTS

1. Use command “sudo apt install python3-numpy”

HTML CREATION  
The html will be a single index.html file with javascript being utilized. The goal is to display images of movie posters that is provided via a bash file that reads from a .txt file. The html will temporarily store the user’s ratings whenever a thumbs up or thumbs down button is pressed. When the user wants to submit ratings, the html should send the user’s rating to a .cgi file that will return a string of indexes. That returned string of indexes can be used to display movie posters of unwatched movies, ordered by the likelihood the user will enjoy, which is determined by the .cgi file.

1. Create a custom html index file, include a basic layout and design.
2. Make 2 buttons, position them side by side and in the centre of the page, one to scroll left and one right.
3. Make 3 image tags, position them side by side and in between the scroll buttons.
4. Make 2 buttons for each image tag, the thumbsUp and thumbsDown buttons, position each pair shoulder to shoulder just beneath the corresponding image tag.
5. Make 1 button, this will be our submit button, position it below thumbsUp and Down buttons in the centre of the page.
6. Fetch the movie poster links with a simple bash file.  
   
7. Read the movie poster links into an array.
8. Set the 3 image tags to the first 3 movies.
9. Whenever scroll button is pressed display next 3 movies, I use an index to keep track of which movie’s we are displaying.  
   A screen shot of a computer code

   AI-generated content may be incorrect.
10. Whenever a thumbsUp or thumbsDown button is pressed store 1 (thumbsUp) or 0 (thumbsDown) to an array with the same index as current displayed movies.
11. Whenever submit button is pressed send the array of user movie ratings to cgi file via post and await response.
12. Response will be a string of integers with each integer representing the index of the array of movie poster links.
13. Convert the response string into an array and overwrite the existing array of movie poster links with each movie poster link in accordance with the response array.  
      
    (Here recommendedMovies is the response array and images is the array of movie poster links)
14. Then hide the thumbsUp, thumbsDown and submit buttons and refresh the displayed posters.

CGI CREATION  
The goal of the cgi script (get\_recommendations.cgi) is to take a string of 1’s, 0’s and null’s and calculate the user’s favourite genres and to order genres by user’s favourites preference. Then find the movie that scores the highest on the user’s favourite genres and store the index of that movie into the first index of the result array. Then find the second highest movie and store that movie into the second index of the result array. Continue this process while ignoring duplicate movies until the result array has 10 movie indexes. Then return the result array.

1. Read received string and put it into an array (This will be our dataArray)
2. Open movie\_data.txt (the file containing each movie’s genre scores) and read the first line into an array. (Each line is a list of integers between 0 and 9 corresponding to how much the specific movie is considered that genre)
3. If the line read index of dataArray has a rating, 1 or 0, either, add or subtract the line array from another array called genrePoints
4. If the line read index of dataArray has no rating add that index to an array called unratedMovies
5. Loop through all lines of movie\_data.txt
6. genrePoints should now according to this method be an array of points, each index represents how much a user like a genre. unratedMovies should now be an array of all movies the user has not rated
7. Make a new array movieGenrePreference and write the index of the highest scoring genre in genrePoints to the first index of movieGenrePreference, then the second highest scoring genre to the second index and so on until genres are sorted by the user’s preference.
8. Read movie\_data.txt into a 2d array and then use unratedMovies and movieGenrePreference to find the top 3 unrated movies with the highest score in the user’s favourite genre. Write the top 3 of each genre into a 2d array
9. Loop through the top 3 of each genre and write the first entry into a 1d array called topTenSorted, then the second, until there are 10 entries, make sure to ignore any duplicates
10. Turn topTenSorted into a string and return the string

SET FILE PERMISSIONS

1. Make sure to set get\_poster.sh and get\_recommendations.cgi so they are executable, use “sudo chmod 755 ‘path\_to\_file’”