

# Workflow

The screenshot shows a Google Slides presentation with the title 'QA Group Project'. The slide contains a hierarchical list of requirements:

- Epic:**
  - Site Navigation
  - Design
  - Movies
  - Cinema
  - User Functionality
- User stories:**
  - Each Movie page - Movies**
    - As a user, I want to view detailed information about the movie, such as reviews and trailers, to make an informed decision about which movies to watch.
      - Task**
        - Create a page for each movie
      - Acceptance criteria**
        - Each movie page should display useful information
    - General design - Design**
      - As a user, I want the background colour of the website to be visually appealing and not white.
        - Tasks**
          - Choose a colour palette for the website
          - Implement the selected background colour

Identified epics and user stories. Each user story has sub tasks and acceptance criteria

The screenshot shows the Jira Backlog interface for the QA-Cinema project. On the left, a sidebar includes links for Planning (Timeline, Backlog, Board, Reports), Development (Code, Project pages, Add shortcut, Project settings), and a note about being a team-managed project. The main area displays the Backlog under the QC Sprint 1 section. It shows a list of tasks categorized by epic: Site Navigation, Design, Movies, Cinema, Functionality, and Bugs. A 'Create issue' button is available. To the right, a detailed view of the QC-6 task is shown, which is assigned to the MOVIES epic and has a priority of TO DO. The task description is: "Create a page for each movie that has the description of the movie". Other tasks in the backlog include QC-7 through QC-14, all assigned to the DESIGN epic.

### Set up Jira board with tasks

This screenshot shows the same Jira Backlog interface as above, but with priority levels assigned to the tasks. The QC-6 task now has a priority of P1. The QC-7 task has a priority of P2. The QC-8 task has a priority of P3. The QC-9 task has a priority of P4. The QC-10 task has a priority of P5. The QC-11 task has a priority of P6. The QC-12 task has a priority of P7. The QC-13 task has a priority of P8. The QC-14 task has a priority of P9. The QC-15 task has a priority of P10. The QC-16 task has a priority of P11. The QC-17 task has a priority of P12. The QC-18 task has a priority of P13. The QC-19 task has a priority of P14. The QC-20 task has a priority of P15. The QC-21 task has a priority of P16. The tasks are color-coded by epic: MOVIES (blue), DESIGN (green), FUNCTIONALITY (orange), and SITE NAVIGATION (purple).

Work in progress: assigning priority to each task and story points. Identifying the best parts of the MVP

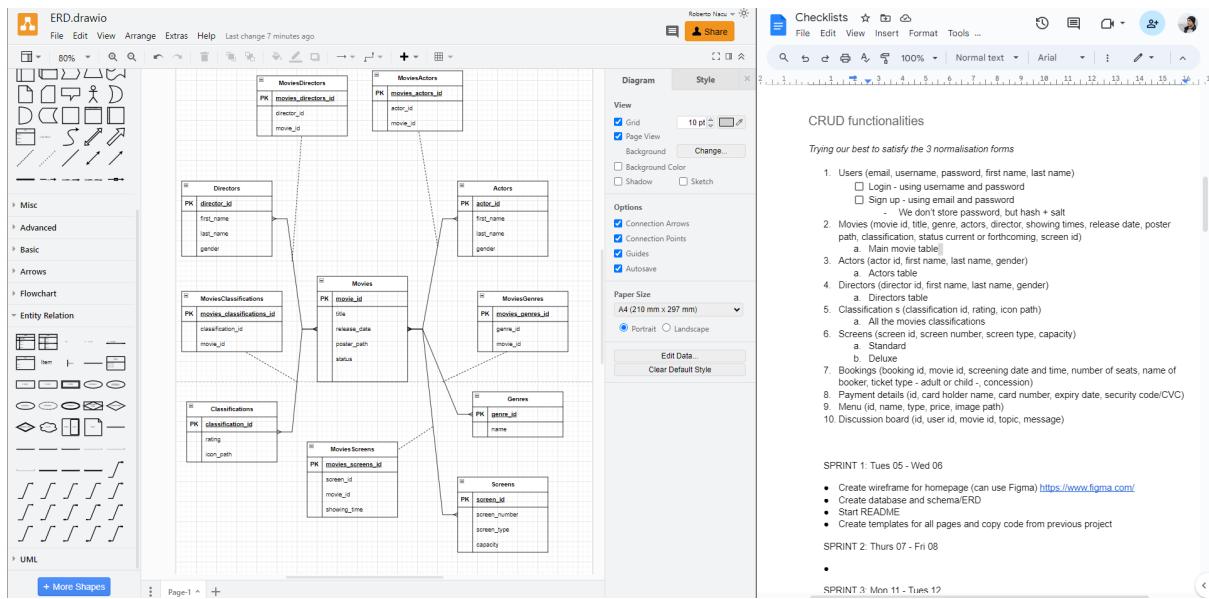
	A	B	C	D	E	F	G	H
	Risk ID	Category	Description	Probability (1 - 5)	Impact (1 - 5)	Severity	Mitigation	
1	2	1	Technical Hardware or software failures in the production environment	2	4	8	Use reliable hosting providers and maintain backups and redundancy measures	
2	3	2	Technical Images or movie posters are broken	2	3	6	Implement adequate tests and always perform code refactoring when needed	
3	4	3	Technical The database fails to load data	2	5	10	Perform database CRUD functionalities testing	
4	5	4	Technical Checkout page does not display accurate total price	2	5	10	Implement adequate tests and always perform code refactoring when needed	
5	6	5	Technical The application does not meet the requirements set in the MVP	2	5	10	Review completed tasks at the end of each sprint and compare them against the initially set user stories, as well as the client requirements set in the project brief	
6	7	6	Complexity Developing the application could introduce technical challenges that the team is not familiar with	4	4	16	Consult with other team members and/or instructors to ensure a smooth integration of DevOps practices	
7	8	7	Security Data breach compromising customer information	2	5	10	Implement strong security measures, encryption, and regular security audits	
8	9	8	Technical Critical bugs in the booking system	3	4	12	Implement a comprehensive testing strategy, including both unit and integration testing	
9	10	9	Competition The cinema industry is competitive, and the success of the platform depends on effectively standing out in the market	3	4	12	Regularly monitor market, adapt to changes, and differentiate the cinema's offerings	
10	11	10	User adoption Insufficient training and support	3	3	9	Develop user-friendly documentation, provide training sessions for QA cinemas' staff, and offer technical support for customers	
11	12	11	Project Management Inadequate project management and communication	4	3	12	Appoint roles and establish clear communication channels among the development team and stakeholders	
12	13	12	Project Management Inappropriate usage of tools (i.e., linking GitHub issues to tasks on Jira)	2	2	4	Research and understand how to use needed tools	
13	14							
14	15							
15	16							
16	17							
17	18							
18	19							
19	20							
20	21							
21								

Performed risk assessment and created risk assessment matrix

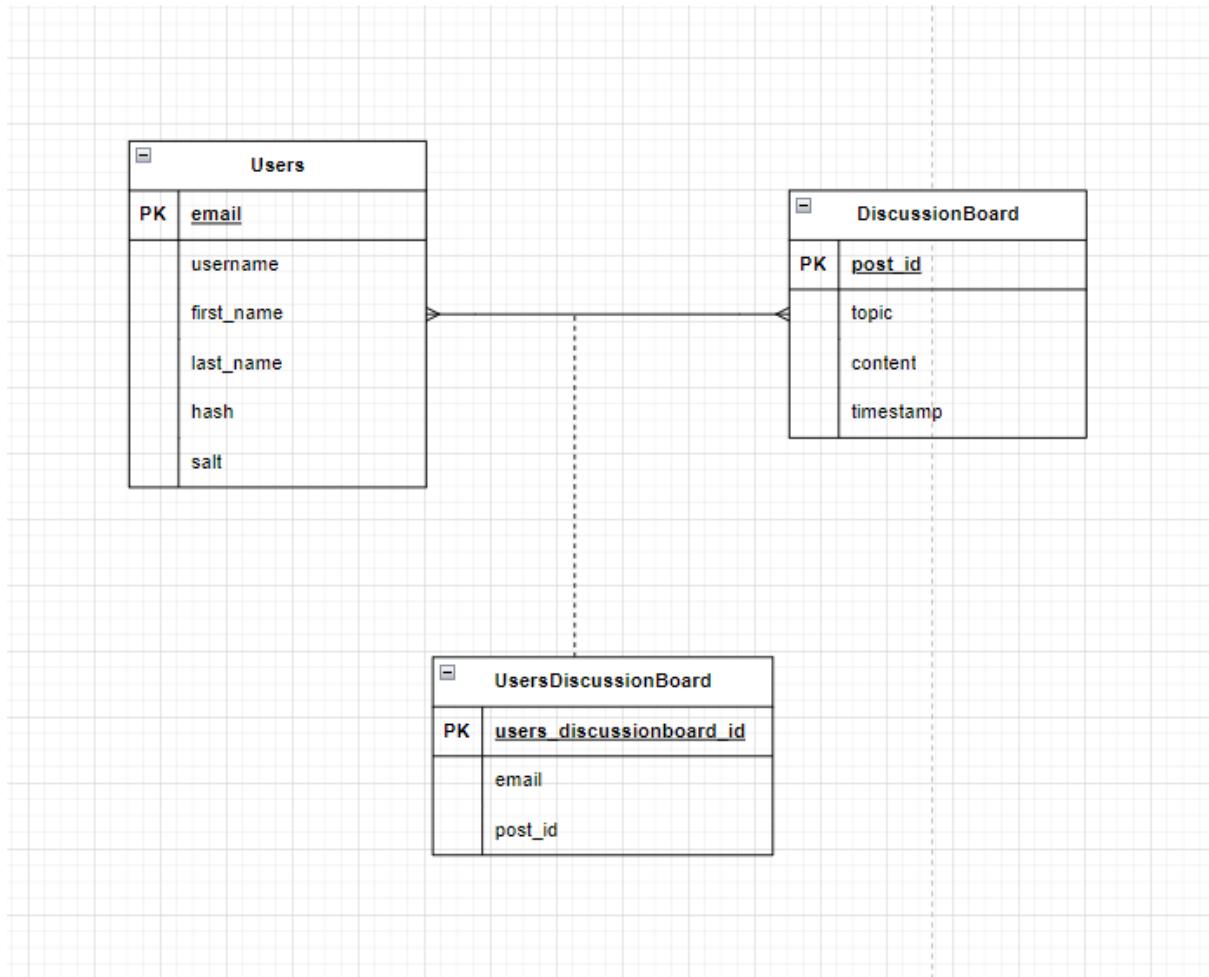
### CRUD functionalities

1. Users (email, username, password, first name, last name)
  - Login - using username and password
  - Sign up - using email and password
    - We don't store password, but hash + salt
2. Movies (movie id, title, genre, actors, director, showing times, release date, poster path, classification, status current or forthcoming, screen)
  - a. Main movie table
3. Actors (actor id, first name, last name, gender)
  - a. Actors table
4. Directors (director id, first name, last name, gender)
  - a. Directors table
5. Classifications (classification id, rating, icon path)
  - a. All the movies classifications
6. Screens (screen id, screen name)
  - a. Standard
  - b. Deluxe
7. Bookings (booking id, movie title, screening date and time, number of seats, name of booker, ticket type - adult or child -, concession)
8. Payment details (id, card holder name, card number, expiry date, security code/CVC)
9. Menu (id, name, type, price, image path)
10. |

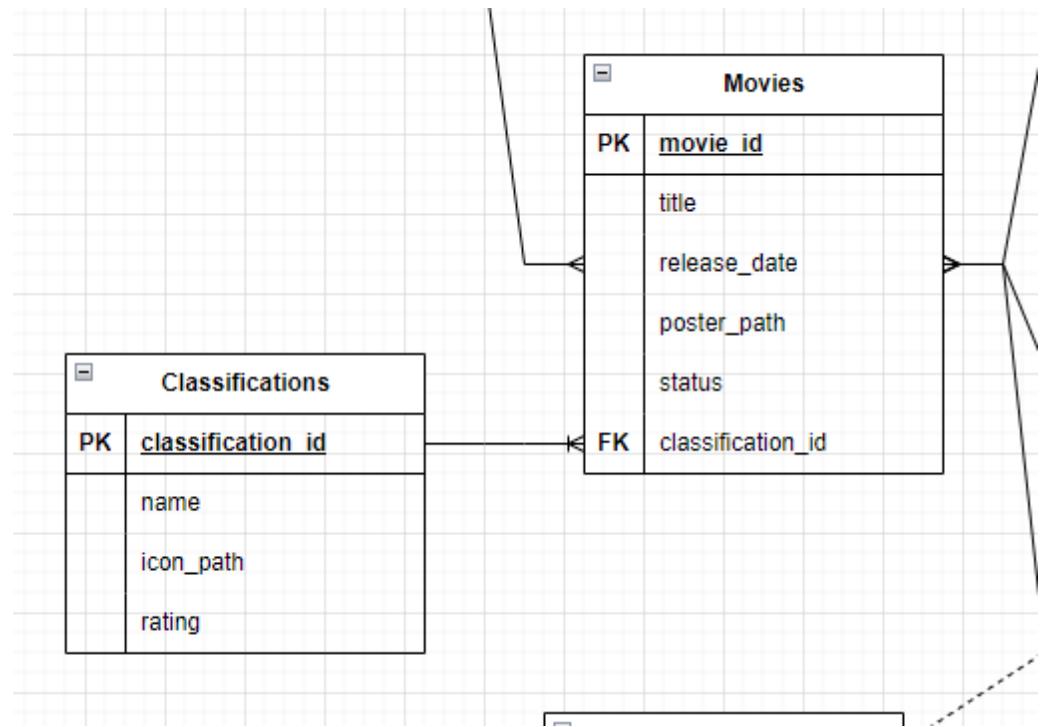
Started working on creating initial database schema



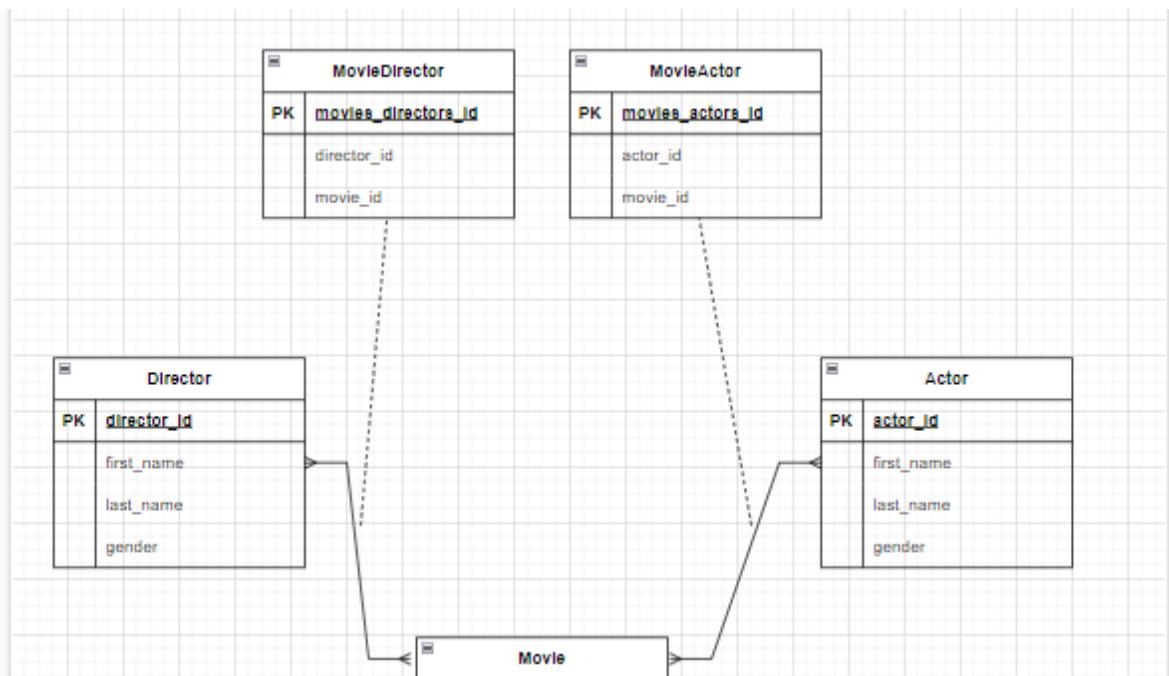
Started working on creation of the ERD

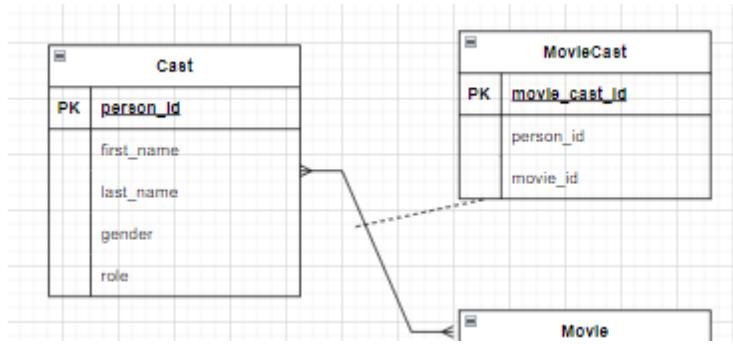


Created discussion board section of the ERD

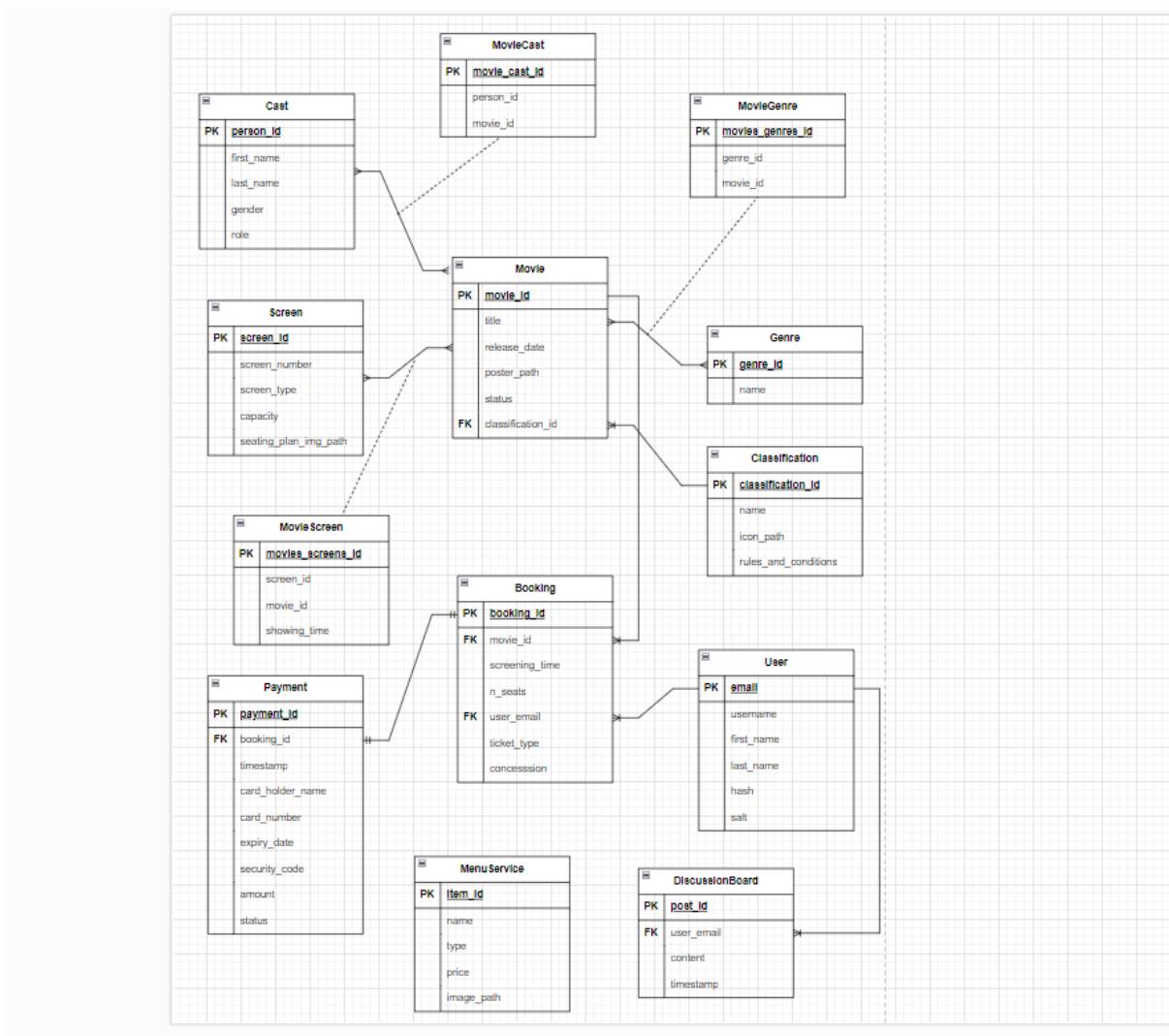


Updated classifications section of the ERD

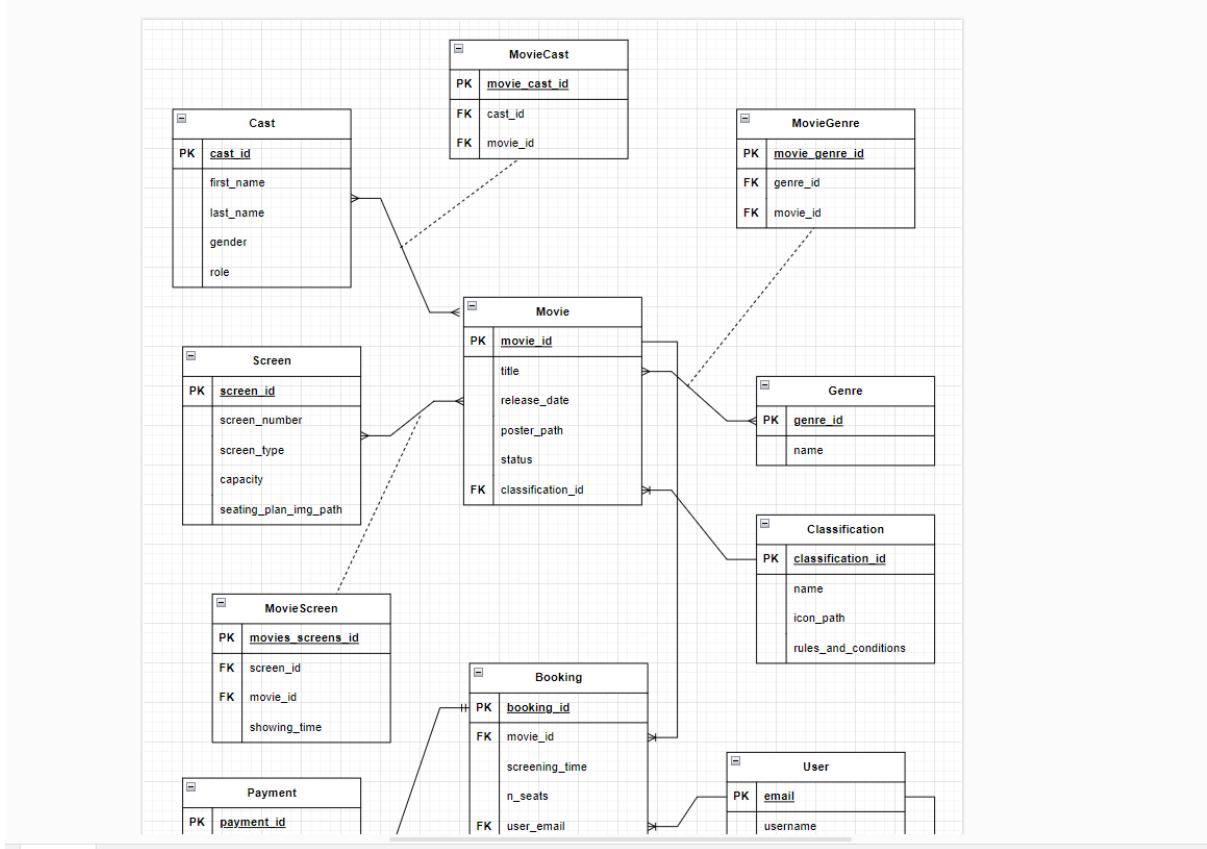




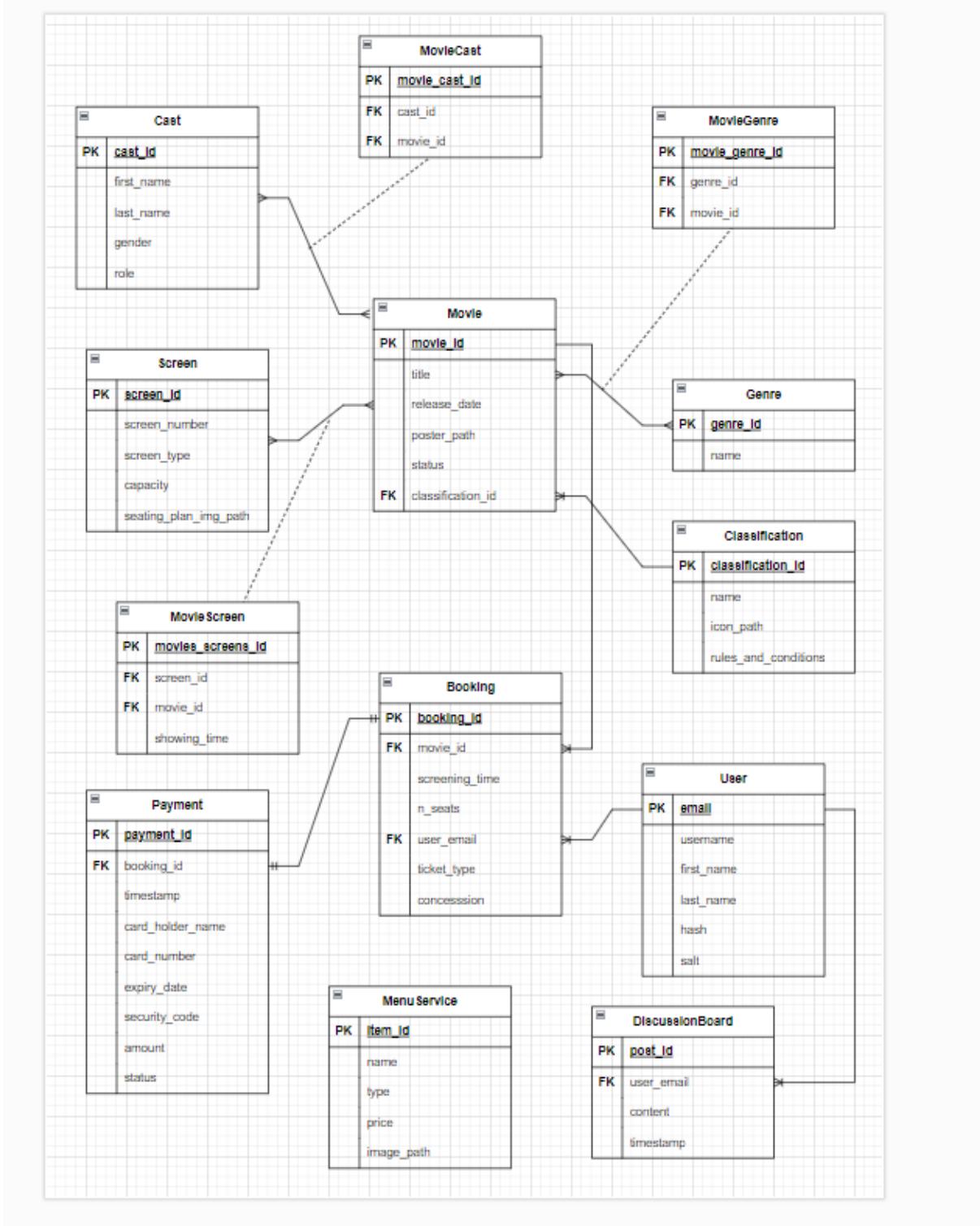
Unified directors and cast under the same table (different role field value - director, actor)



Completed draft one of the ERD - needs to be reviewed



Added constraints in associative tables



Finalised draft 2 of the ERD. Time to work on modules creation (Farah)

The screenshot shows the AWS Management Console homepage. At the top, there's a search bar and a navigation bar with options like 'Services', 'London', and 'qalearner @ qa77'. Below the search bar is a 'Recently visited' section with icons for EC2, CloudWatch, Elastic Kubernetes Service, X-Ray, VPC, Lambda, S3, RDS, IAM, and Elastic Container Service. A 'View all services' link is at the bottom of this list. To the right of this is the 'Welcome to AWS' section, which includes links for 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The 'AWS Health' section shows a heart icon and a message stating 'No health data'. At the bottom of the page are links for 'CloudShell', 'Feedback', 'Language', and copyright information.

Time to set up AWS related things (Terraform, Jenkins, Ansible, Docker, etc)

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'New EC2 Experience' and links for 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', and 'Savings Plans'. The main area shows a table titled 'Instances (1/1)'. The table has columns for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', and 'Alarm status'. One row is visible, showing 'terraform-server' as the name, 'i-0f5e6924ec183cdce' as the Instance ID, 'Running' as the state, 't2.medium' as the type, and 'No alarms' for status and alarm. There are buttons for 'Connect', 'Actions', and 'Launch instances'.

Setting up a terraform server/instance

The screenshot shows the AWS Management Console interface for managing security groups. On the left, there's a sidebar with 'Security groups' and a single item: 'sg-0135ad289768af735 (launch-wizard-1)'. Below this, under 'Inbound rules', there's a table with one row. The table has columns for 'Name', 'Security group rule ID', 'Port range', 'Protocol', and 'Source'. The single rule is: Name '-' (empty), Security group rule ID 'sgr-0974c0a6fc941f372', Port range '22', Protocol 'TCP', and Source '0.0.0.0/0'. A search bar at the top says 'Filter rules'.

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0974c0a6fc941f372	22	TCP	0.0.0.0/0

Created custom security group - only SSH allowed for now

```
ubuntu@ip-172-31-9-139:~$ sudo apt update && sudo apt upgrade -y
```

Initial set up, a must on all instances

```
ubuntu@ip-172-31-9-139:~$ terraform --version
Terraform v1.5.6
on linux_amd64
ubuntu@ip-172-31-9-139:~$
```

Installed terraform

```
ubuntu@ip-172-31-9-139:~/aws$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 116 Sep  6 10:15 credentials
ubuntu@ip-172-31-9-139:~/aws$
```

Created a credentials file for the AWS configuration

```
ubuntu@ip-172-31-9-139: ~/terraform
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 2.0"
    }
  }
}

# configure AWS provider
provider "aws" {
  region           = "eu-west-2"
  shared_credentials_file = "~/.aws/credentials"
}

# create a new resource - testing for now
resource "aws_instance" "EC2" {
  ami           = var.ami-id
  instance_type = var.instance_type
}
```

Created a relatively simple terraform config to verify that everything is working properly

```

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v2.70.4

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-9-139:~/terraform$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions
are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.EC2 will be created
+ resource "aws_instance" "EC2" {
    + ami                               = "ami-0eb260c4d5475b901"
    + arn                             = (known after apply)
    + associate_public_ip_address     = (known after apply)
    + availability_zone                = (known after apply)
    + cpu_core_count                  = (known after apply)
    + cpu_threads_per_core           = (known after apply)
    + get_password_data              = false
    + host_id                         = (known after apply)
    + id                             = (known after apply)
    + instance_state                 = (known after apply)
    + instance_type                  = "t2.micro"
    + ipv6_address_count             = (known after apply)
    + ipv6_addresses                 = (known after apply)
    + key_name                       = (known after apply)
    + network_interface_id          = (known after apply)
    + outpost_arn                    = (known after apply)
    + password_data                 = (known after apply)
    + placement_group                = (known after apply)
    + primary_network_interface_id   = (known after apply)
    + private_dns                     = (known after apply)
    + private_ip                      = (known after apply)
    + public_dns                      = (known after apply)
    + public_ip                       = (known after apply)
    + security_groups                = (known after apply)
    + source_dest_check              = true
    + subnet_id                      = (known after apply)
    + tenancy                         = (known after apply)
    + volume_tags                     = (known after apply)
    + vpc_security_group_ids         = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ ec2_id = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take
exactly these actions if you run "terraform apply" now.

```

```

ubuntu@ip-172-31-9-139:~/terraform$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions
are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.EC2 will be created
+ resource "aws_instance" "EC2" {
  + ami
  + arn
  + associate_public_ip_address
  + availability_zone
  + cpu_core_count
  + cpu_threads_per_core
  + get_password_data
  + host_id
  + id
  + instance_state
  + instance_type
  + ipv6_address_count
  + ipv6_addresses
  + key_name
  + network_interface_id
  + outpost_arn
  + password_data
  + placement_group
  + primary_network_interface_id
  + private_dns
  + private_ip
  + public_dns
  + public_ip
  + security_groups
  + source_dest_check
  + subnet_id
  + tenancy
  + volume_tags
  + vpc_security_group_ids
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ ec2_id = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.EC2: Creating...
aws_instance.EC2: Still creating... [10s elapsed]
aws_instance.EC2: Still creating... [20s elapsed]
aws_instance.EC2: Creation complete after 21s [id=i-0c5ee559766f5251c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

ec2_id = "i-0c5ee559766f5251c"
ubuntu@ip-172-31-9-139:~/terraform$ terraform output
ec2_id = "i-0c5ee559766f5251c"
ubuntu@ip-172-31-9-139:~/terraform$ _

```

	Name	Instance ID	Instance state	Instance type	Status check
	-	i-0c5ee559766f5251c	Running	t2.micro	Initializing

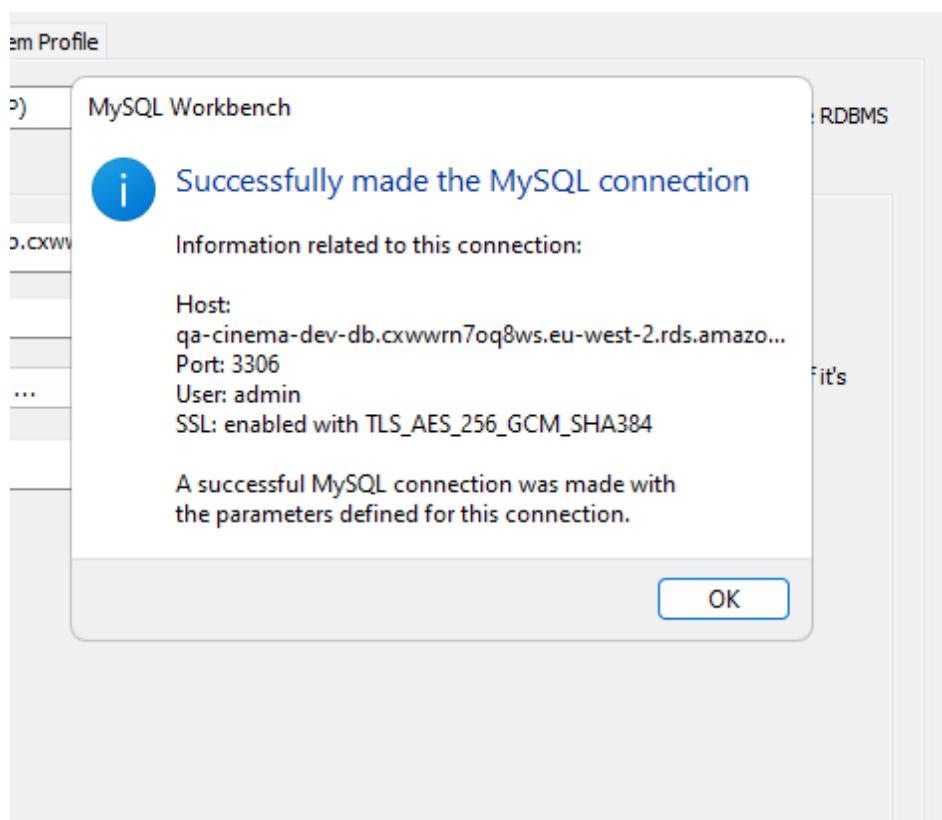
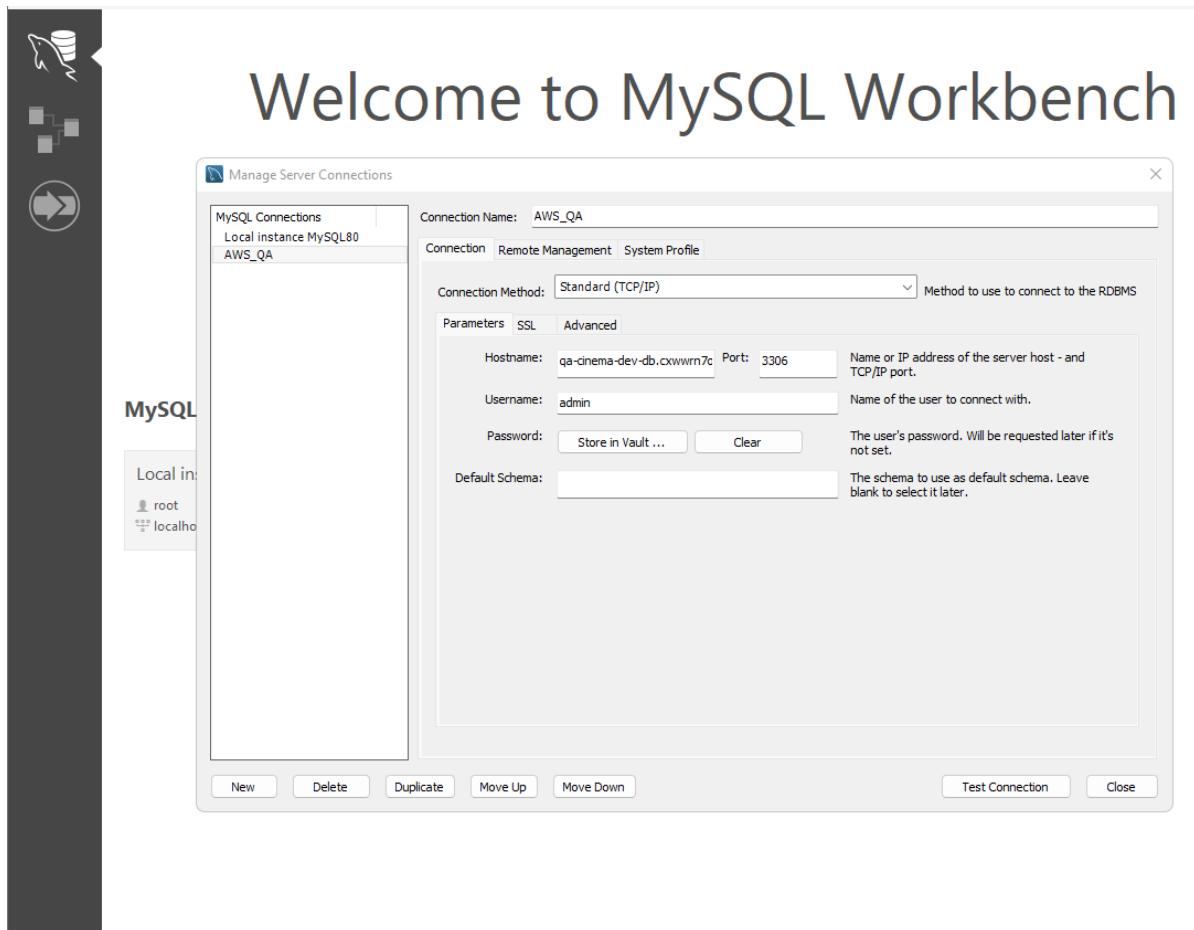
Everything working as it should

```
ubuntu@ip-172-31-9-139: ~/Codebusters/terraform/easy-test
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 2.0"
    }
  }
}

# configure AWS provider
provider "aws" {
  region           = "eu-west-2"
  shared_credentials_file = "~/.aws/credentials"
}

# creat a RDS instance
resource "aws_db_instance" "qa_cinema_dev_db" {
  engine          = "mysql"
  allocated_storage  = 20
  identifier       = "qa-cinema-dev-db"
  storage_type     = "gp2"
  engine_version   = "8.0.33"
  instance_class   = "db.t2.micro"
  username         = "admin"
  password         = "password123"
  skip_final_snapshot = true
  publicly_accessible = true
}
```

Created a very simple db connection, sensitive data is not hidden at the moment, and a lot of things need to be improved



SQL connection with cloud based db working. Needs a lot of refactoring and further work

```

Enter a value: yes

aws_key_pair.EC2_kp: Creating...
aws_vpc.qa_cinema_vpc: Creating...
aws_key_pair.EC2_kp: Creation complete after 0s [id=EC2_kp]
aws_vpc.qa_cinema_vpc: Creation complete after 1s [id=vpc-0a4c85e1a8244fc3e]
aws_subnet.qa_cinema_public_subnet[0]: Creating...
aws_internet_gateway.qa_cinema_igw: Creating...
aws_security_group.qa_cinema_web_sg: Creating...
aws_subnet.qa_cinema_private_subnet[1]: Creating...
aws_subnet.qa_cinema_private_subnet[0]: Creating...
aws_route_table.qa_cinema_private_rt: Creating...
aws_route_table.qa_cinema_private_rt: Creation complete after 0s [id=rtb-0dae3b2b3da8eca28]
aws_internet_gateway.qa_cinema_igw: Creation complete after 0s [id=igw-0d3356ea1a8b2780]
aws_route_table.qa_cinema_public_rt: Creating...
aws_subnet.qa_cinema_private_subnet[1]: Creation complete after 1s [id=subnet-0da43cd7d5a49be53]
aws_subnet.qa_cinema_public_subnet[0]: Creation complete after 1s [id=subnet-0b3a68db950a5ca58]
aws_subnet.qa_cinema_private_subnet[0]: Creation complete after 1s [id=subnet-09d1cad5c5d1faf5b]
aws_db_subnet_group.qa_cinema_db_subnet_group: Creating...
aws_route_table_association.private[1]: Creating...
aws_route_table_association.private[0]: Creating...
aws_route_table.qa_cinema_public_rt: Creation complete after 1s [id=rtb-0cf29d065a56b097]
aws_route_table_association.public[0]: Creating...
aws_route_table_association.private[1]: Creation complete after 0s [id=rtbassoc-0b9fe02f9761f44d9]
aws_route_table_association.private[0]: Creation complete after 0s [id=rtbassoc-0c81cae92db4c64f]
aws_route_table_association.public[0]: Creation complete after 0s [id=rtbassoc-0f78fa9821f5915ad]
aws_security_group.qa_cinema_web_sg: Creation complete after 1s [id=sg-09271ca4c756837c9]
aws_instance.web[0]: Creating...
aws_db_subnet_group.qa_cinema_db_subnet_group: Creation complete after 1s [id=qa_cinema_db_subnet_group]
aws_instance.web[0]: Still creating... [10s elapsed]
aws_instance.web[0]: Still creating... [20s elapsed]
aws_instance.web[0]: Creation complete after 22s [id=i-0e3068cba4bc274d7]
aws_eip.qa_cinema_web_eip[0]: Creating...
aws_eip.qa_cinema_web_eip[0]: Creation complete after 1s [id=eipalloc-0e01f982b2f5d8560]

9 | Error: "sg-09271ca4c756837c9" is not a valid CIDR block: invalid CIDR address: sg-09271ca4c756837c
|   |
|   with aws_security_group.qa_cinema_db_sg,
|   on main.tf line 183, in resource "aws_security_group" "qa_cinema_db_sg":
| 183: resource "aws_security_group" "qa_cinema_db_sg" {
```

ubuntu@ip-172-31-9-139:~/Codebusters/terraform\$ \_

Working on creating a more advanced terraform configuration. Having problems with CIDR blocks and security groups

```

    protocol      = "tcp"
    security_groups = [aws_security_group.qa_cinema_web_sg.id]
}
```

Fixed the error, instead of specifying the CIDR block, I had to specify the security group (typo I guess)

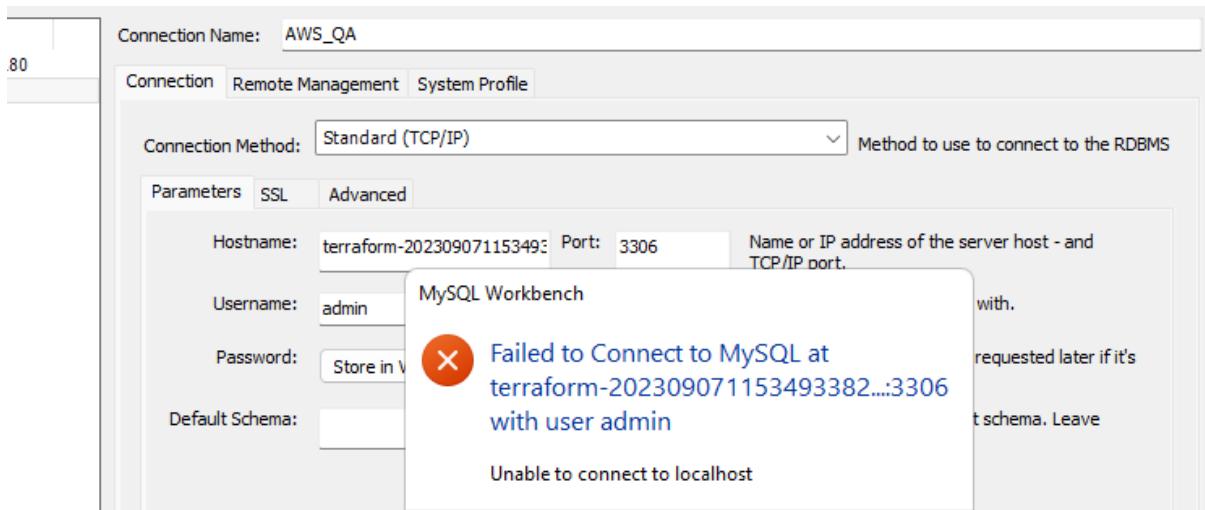
```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

database_endpoint = "terraform-2023090711534933820000001.cxwwrn7oq8ws.eu-west-2.rds.amazonaws.com"
database_port = 3306
web_public_dns = "ec2-13-41-88-70.eu-west-2.compute.amazonaws.com"
web_public_ip = "13.41.88.70"
ubuntu@ip-172-31-9-139:~/Codebusters/terraform$
```

Our Terraform configuration is successfully working, time to test if things were set up properly



Expected behaviour: we cannot connect to the RDS database over the internet

```
ssh -i ~/.ssh/EC2_kp ubuntu@$({terraform output -raw web_public_dns})
```

```
ubuntu@ip-10-0-1-220:~$ ssh -i ~/.ssh/EC2_kp ubuntu@$({terraform output -raw web_public_ip})
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Thu Sep  7 12:47:43 UTC 2023

 System load: 0.080078125      Processes:          98
 Usage of /: 24.2% of 7.57GB   Users logged in:     0
 Memory usage: 26%            IPv4 address for eth0: 10.0.1.220
 Swap usage:  0%

 Expanded Security Maintenance for Applications is not enabled.

 117 updates can be applied immediately.
 66 of these updates are standard security updates.
 To see these additional updates run: apt list --upgradable

 Enable ESM Apps to receive additional future security updates.
 See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Sep  7 12:41:42 2023 from 13.40.136.11
ubuntu@ip-10-0-1-220:~$ ls -l
total 0
```

We can SSH into the newly created EC2 instances through SSH from the terraform server instance

```
mysql -h <ENDPOINT> -P 3306 -u admin -p
```

```

ubuntu@ip-10-0-1-220:~$ mysql -h terraform-2023090712305725930000001.cxwwrn7oq8ws.eu-west-2.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.00 sec)

mysql>

```

We can connect to our RDS database only through the newly created EC2 instances

```

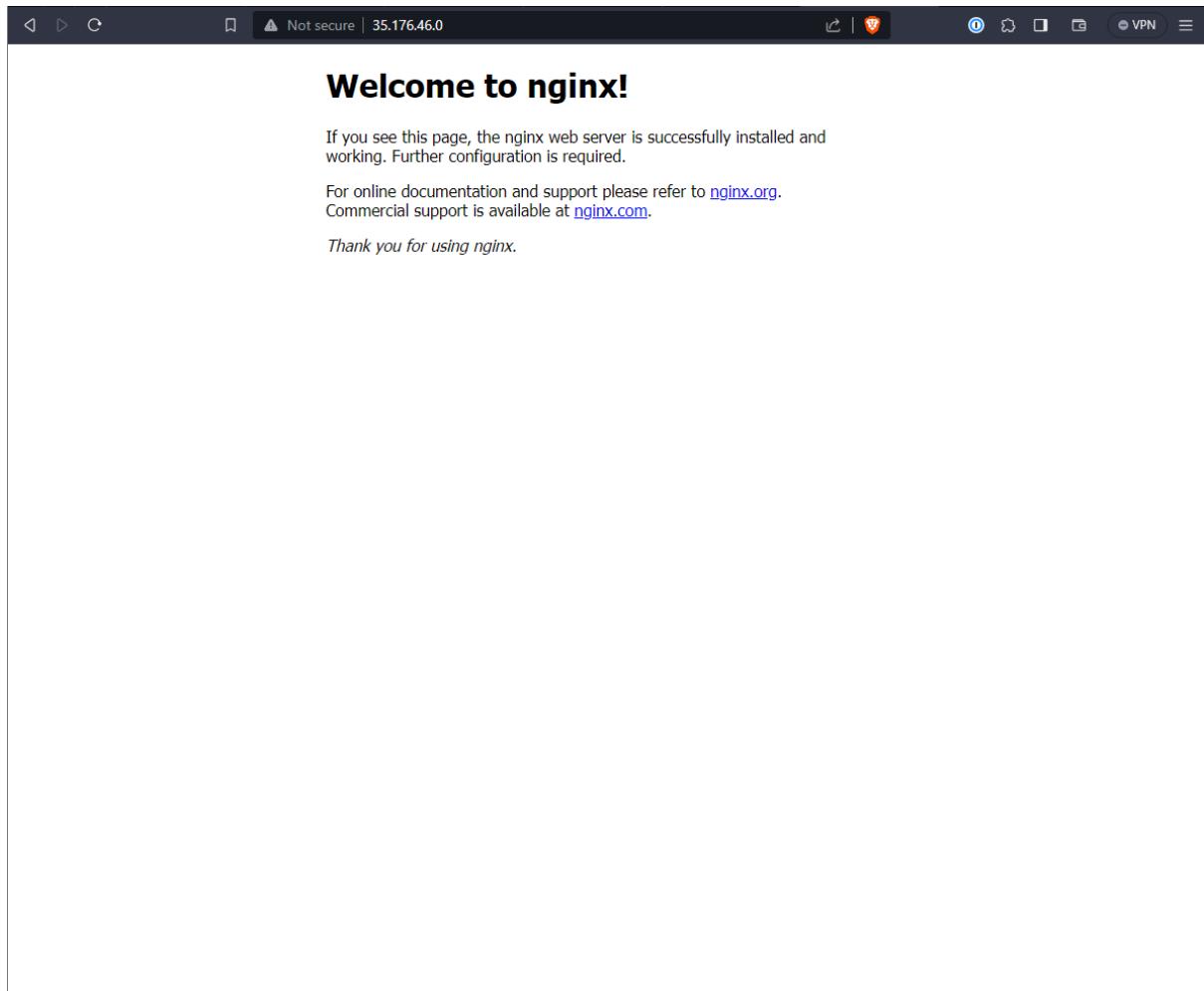
ubuntu@ip-10-0-1-220:~$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-10-0-1-220:~$ _

```

Installed basic NGINX web server on the newly created EC2 instance to check if the network configurations done through Terraform are set up properly or not



Everything works as expected

```

application > models.py
1  from application import db
2  from flask import Flask
3  from flask.ext.sqlalchemy import SQLAlchemy
4  from sqlalchemy import Column, Integer, String, ForeignKey, DateTime, Float, Boolean
5
6
7  # I used a length of 255 for a lot of them as it's common practice (compatibility, flexibility, avoiding truncation)
8
9  class User(db.Model):
10
11      user_email = db.Column(db.String(30), primary_key=True) #Changed email to user_email
12      user_name = db.Column(db.String(30), nullable=False)
13      first_name = db.Column(db.String(30), nullable=False)
14      last_name = db.Column(db.String(30), nullable=False)
15      hash = db.Column(db.String(30), nullable=False)
16      salt = db.Column(db.String(30), nullable=False)
17      #orders = db.relationship('Booking', backref='customer')
18      #all bookings for a specific customer
19      bookings = db.relationship('Booking', back_populates='customer', overlaps='orders.user')
20      orders = db.relationship('Booking', back_populates='user', overlaps='bookings.customer')
21
22
23  class Booking(db.Model):
24
25      booking_id = db.Column(db.Integer, primary_key=True)
26      movie_id = db.Column(db.Integer, db.ForeignKey('movie.movie_id'))
27      screening_time = db.Column(db.DateTime, nullable=False)
28      n_seats = db.Column(db.Integer, nullable=False)
29      user_email = db.Column(db.String(30), db.ForeignKey('user.user_email'))
30      ticket_type = db.Column(db.String(30), nullable=False)
31      concession = db.Column(db.Boolean, nullable=False)
32      #user = db.relationship('User', backref='bookings')
33      # Define the user and customer relationships and specify back_populates
34      user = db.relationship('User', back_populates='orders', overlaps='bookings.customer')
35      customer = db.relationship('User', back_populates='bookings', overlaps='user.orders')
36
37
38
39  class Movie(db.Model):
40
41      movie_id = db.Column(db.Integer, primary_key=True)
42      title = db.Column(db.String(30), nullable=False)
43      release_date = db.Column(db.Date, nullable=False)
44      poster_path = db.Column(db.String(255), nullable=False)
45      status = db.Column(db.String(20), nullable=False)
46      classification_id = db.Column(db.Integer, db.ForeignKey('classification.classification_id'))
47      genre = db.relationship('MovieGenre', backref='movie')
48      # Add this line to create a relationship with Classification
49      classification = db.relationship('Classification', backref='movies')

```

Starting to work on integration, forked and cloned group repo, usage of venv and the requirements.txt file

```

(form.Firstname.data) (form.lastname.data)! You are now able to log in",
login"))
html", form=form)

tml")

html", form=form)

, "POST"])

html")

n.html")

```

\$ git status  
On branch feature-database  
Your branch is ahead of 'origin/feature-database' by 24 commits.  
(use "git push" to publish your local commits)  
  
Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
 deleted: \_\_pycache\_\_/form.cpython-311.pyc  
 deleted: \_\_pycache\_\_/forms.cpython-311.pyc  
 modified: app.py  
 modified: application/\_init\_.py  
 modified: application/\_\_pycache\_\_/\_init\_\_.cpython-311.pyc  
 deleted: application/\_\_pycache\_\_/models.cpython-311.pyc  
 deleted: application/\_\_pycache\_\_/routes.cpython-311.pyc  
 deleted: application/database.db  
 deleted: application/models.py  
 deleted: application/routes.py  
 Modified: application/templates/about.html  
 deleted: application/templates/cart.html  
 modified: application/templates/checkout.html  
 modified: application/templates/classification.html  
 modified: application/templates/home.html  
 modified: application/templates/login.html  
 modified: application/templates/movies.html  
 modified: application/templates/payment.html  
 modified: application/templates/register.html  
 modified: application/templates/search.html  
 deleted: application/templates/services.html  
  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
 application/modules/  
 application/routes/  
 application/static/  
 application/templates/booking.html  
 application/templates/discussion.html  
 application/templates/layout.html  
 application/templates/register.html  
 application/templates/search.html  
 application/templates/services.html  
  
no changes added to commit (use "git add" and/or "git commit -a")  
(venv)  
rober@shizuka MINGW64 ~/Documents/MEGA/work/AMS/SFTA-final-team-project/Codebusters (feature-database)

Performed some code refactoring

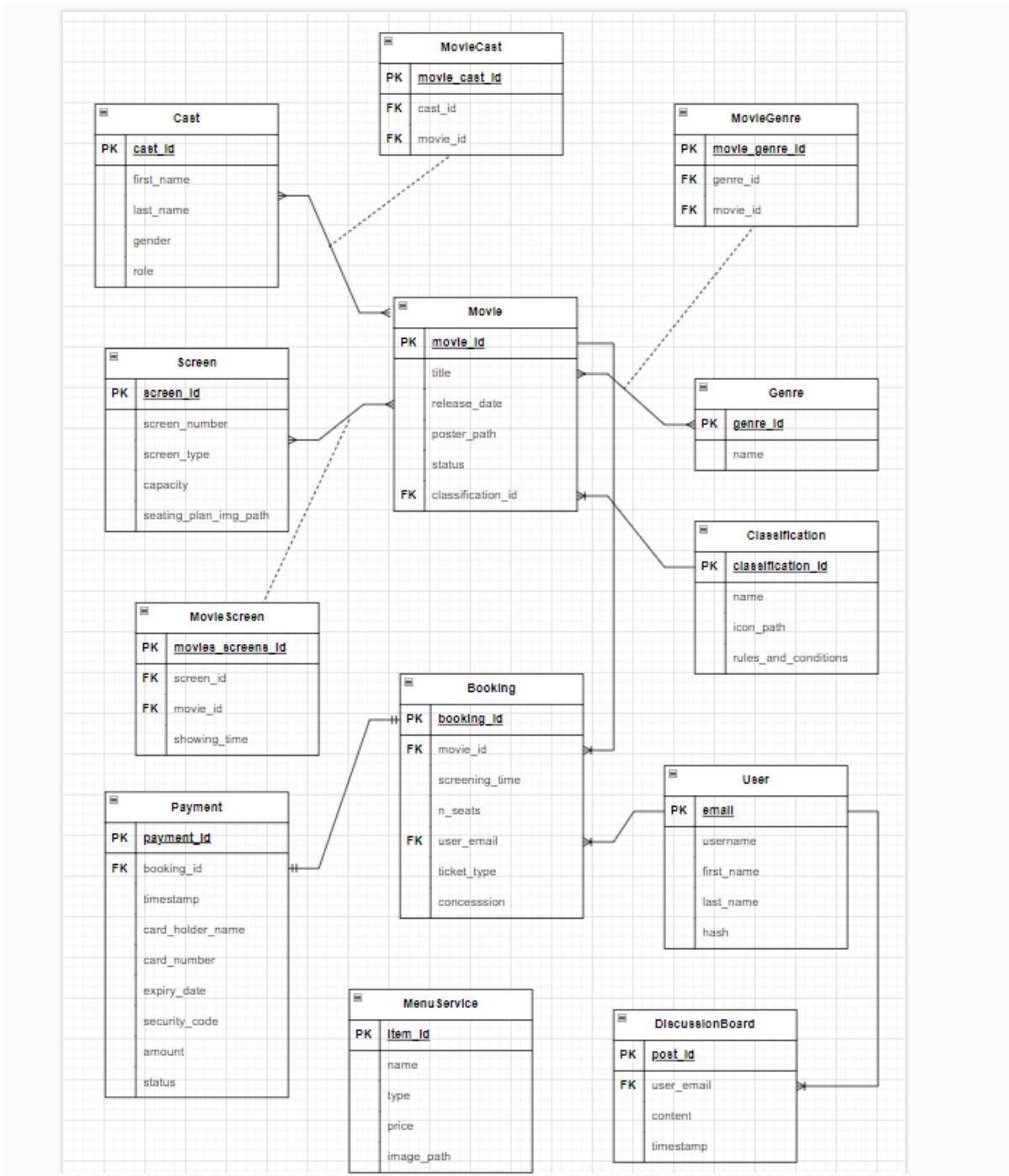
```

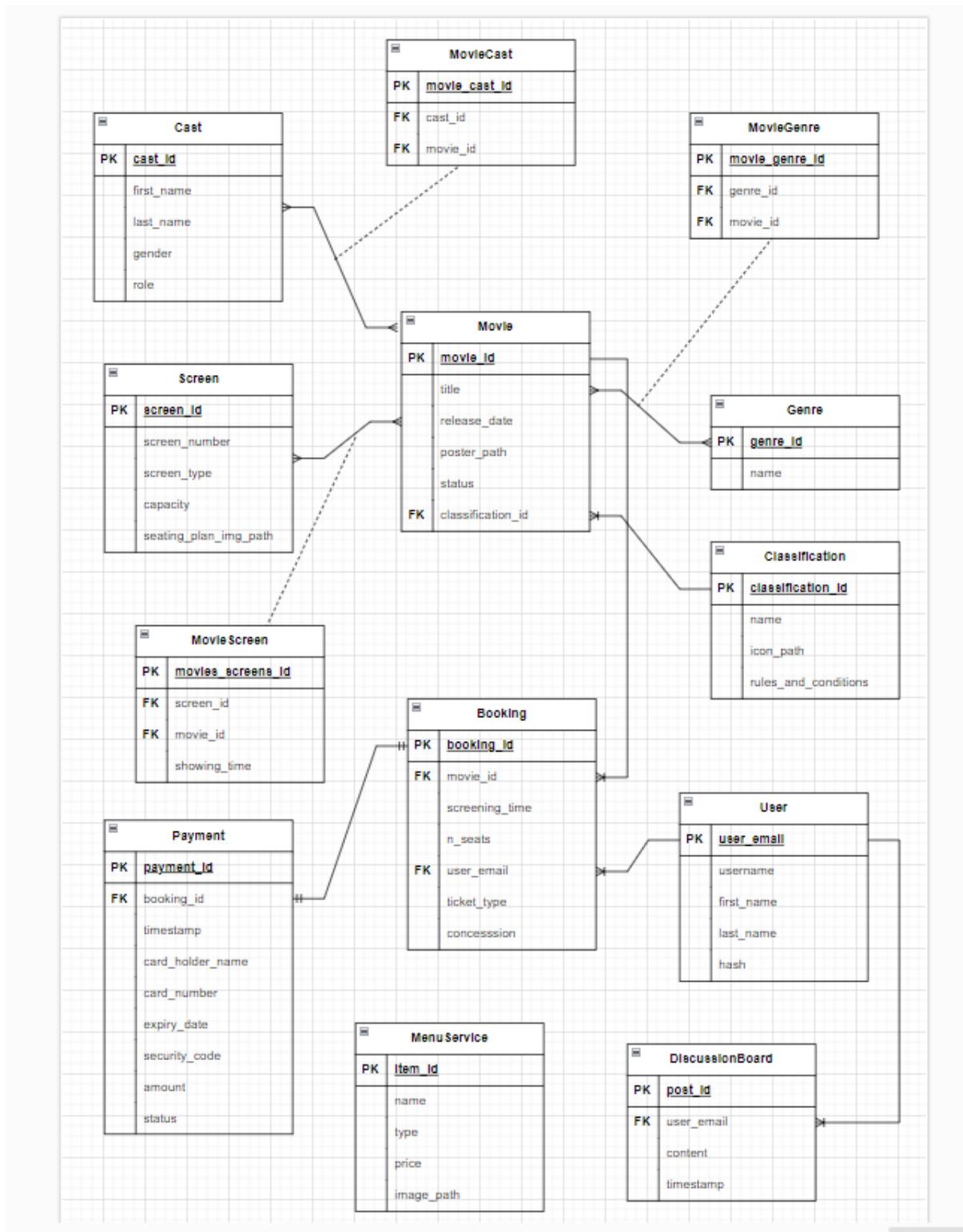
4
5
6 class User(db.Model):
7     user_email = db.Column(
8         db.String(30), primary_key=True
9     ) # Changed email to user_email
10    username = db.Column(db.String(20), nullable=False)
11    first_name = db.Column(db.String(30), nullable=False)
12    last_name = db.Column(db.String(30), nullable=False)
13    hash = db.Column(db.String(60), nullable=False)
14    # this can be removed
15    # -> flask_bcrypt handles the salt for us
16    # salt = db.Column(db.String(30), nullable=False)
17    # orders = db.relationship('Booking', backref='customer')
18    # Define the bookings relationship and specify back_populates
19    bookings = db.relationship(
20        "Booking", back_populates="customer", overlaps="orders,user"
21    )
22    orders = db.relationship(
23        "Booking", back_populates="user", overlaps="bookings, customer"
24    )
25

```

Removed salt field from User table

TODO: ERD needs to be updated → Done





Also updated the primary key name of the User table from email to user\_email

```

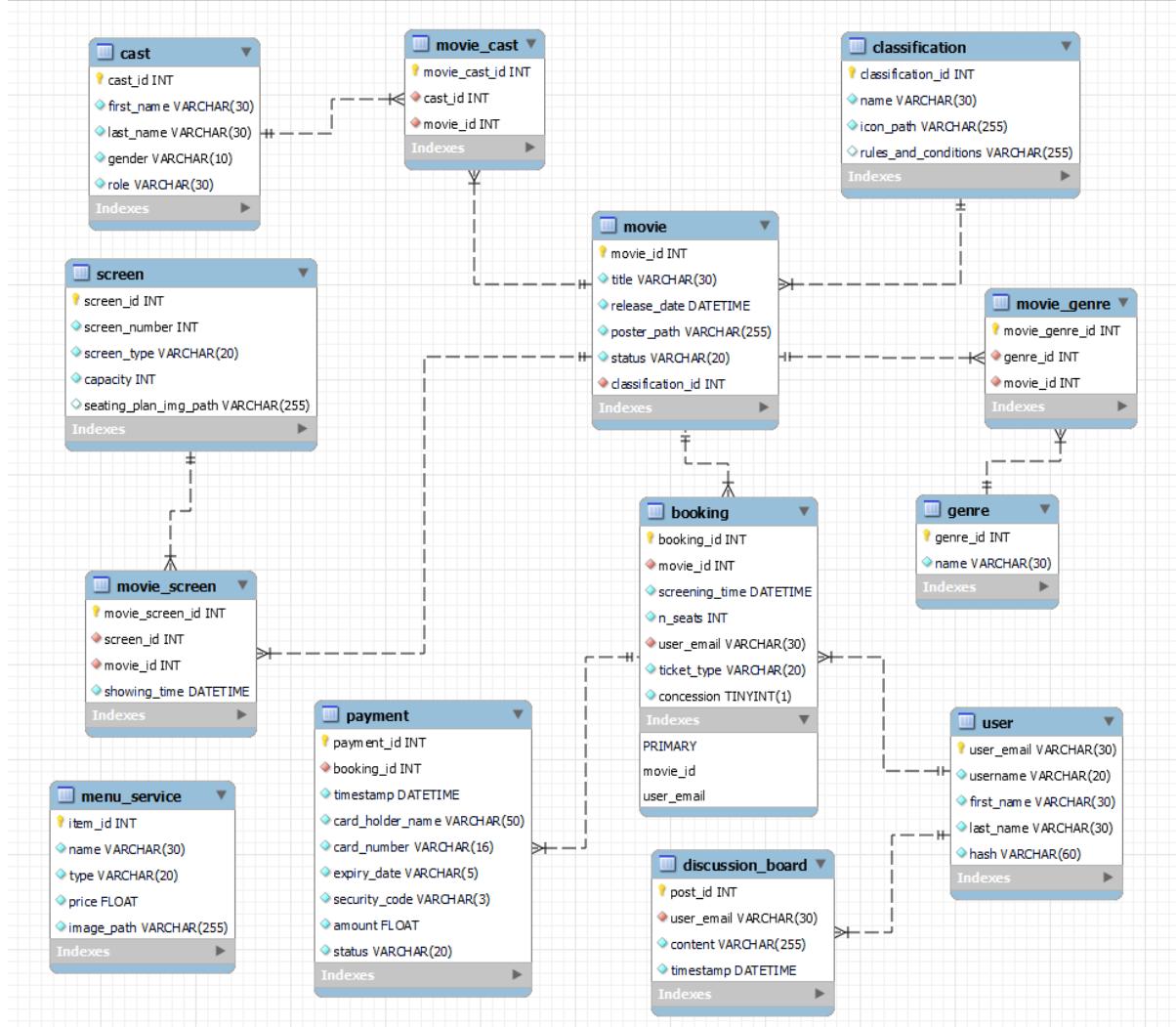
routes.py | 34 createpy M  _init_.py M
application > # _init_.py ...
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3 from flask_bcrypt import Bcrypt
4 import os
5 from dotenv import load_dotenv
6
7 # Create the Flask application
8 app = Flask(__name__)
9
10 # loading environment variables
11 load_dotenv()
12
13 # retrieving environment variables from .env file
14 DB_TYPE = os.getenv("DB_TYPE")
15 DB_USER = os.getenv("DB_USER")
16 DB_PASSWORD = os.getenv("DB_PASSWORD")
17 DB_HOST = os.getenv("DB_HOST")
18 DB_NAME = os.getenv("DB_NAME")
19
20 app.config["SQLALCHEMY_DATABASE_URI"] = (
21     f'{DB_TYPE}://'
22     f'{DB_USER}:{DB_PASSWORD}@{DB_HOST}/{DB_NAME}'
23 )
24
25 # form security
26 SECRET_KEY = os.getenv("SECRET_KEY")
27 app.config["SECRET_KEY"] = SECRET_KEY
28
29 # Create the SQLAlchemy database object
30 db = SQLAlchemy(app)
31 bcrypt = Bcrypt(app)
32
33 # Import your routes and models
34 from application.routes import routes

```

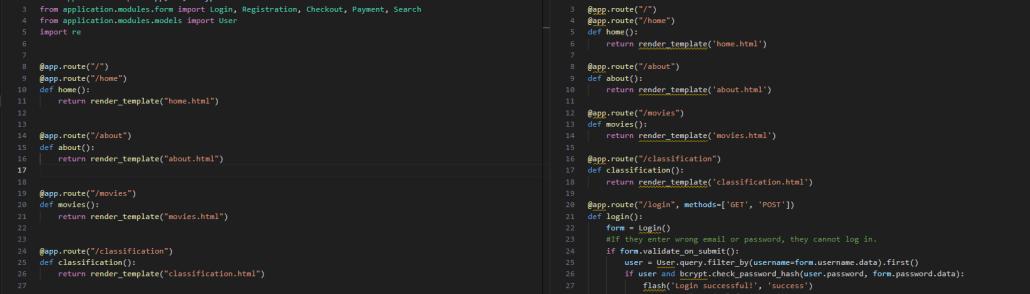
(venv)  
rober@zhizi: MINGW64 ~/Documents/MEGA/work/AMS/SFIA-final-team-project/Codebusters (feature-database)  
\$ py create.py  
(venv)  
rober@zhizi: MINGW64 ~/Documents/MEGA/work/AMS/SFIA-final-team-project/Codebusters (feature-database)  
\$ py create.py  
(venv)  
rober@zhizi: MINGW64 ~/Documents/MEGA/work/AMS/SFIA-final-team-project/Codebusters (feature-database)  
\$ git status  
On branch feature-database  
Your branch is ahead of 'origin/feature-database' by 2 commits.  
(use "git push" to publish your local commits)  
Changes not staged for commit:  
(use "git add" and/or "git commit -a")  
(use "git restore <file>..." to discard changes in working directory)  
modified: application/\_init\_\_.py  
modified: create.py  
no changes added to commit (use "git add" and/or "git commit -a")  
(venv)  
rober@zhizi: MINGW64 ~/Documents/MEGA/work/AMS/SFIA-final-team-project/Codebusters (feature-database)  
\$

More refactoring regarding anything database related. Switched to perform some tests with MySQL rather than the initial sqlite db. This will be needed later on when we will transition to a cloud based

RDS



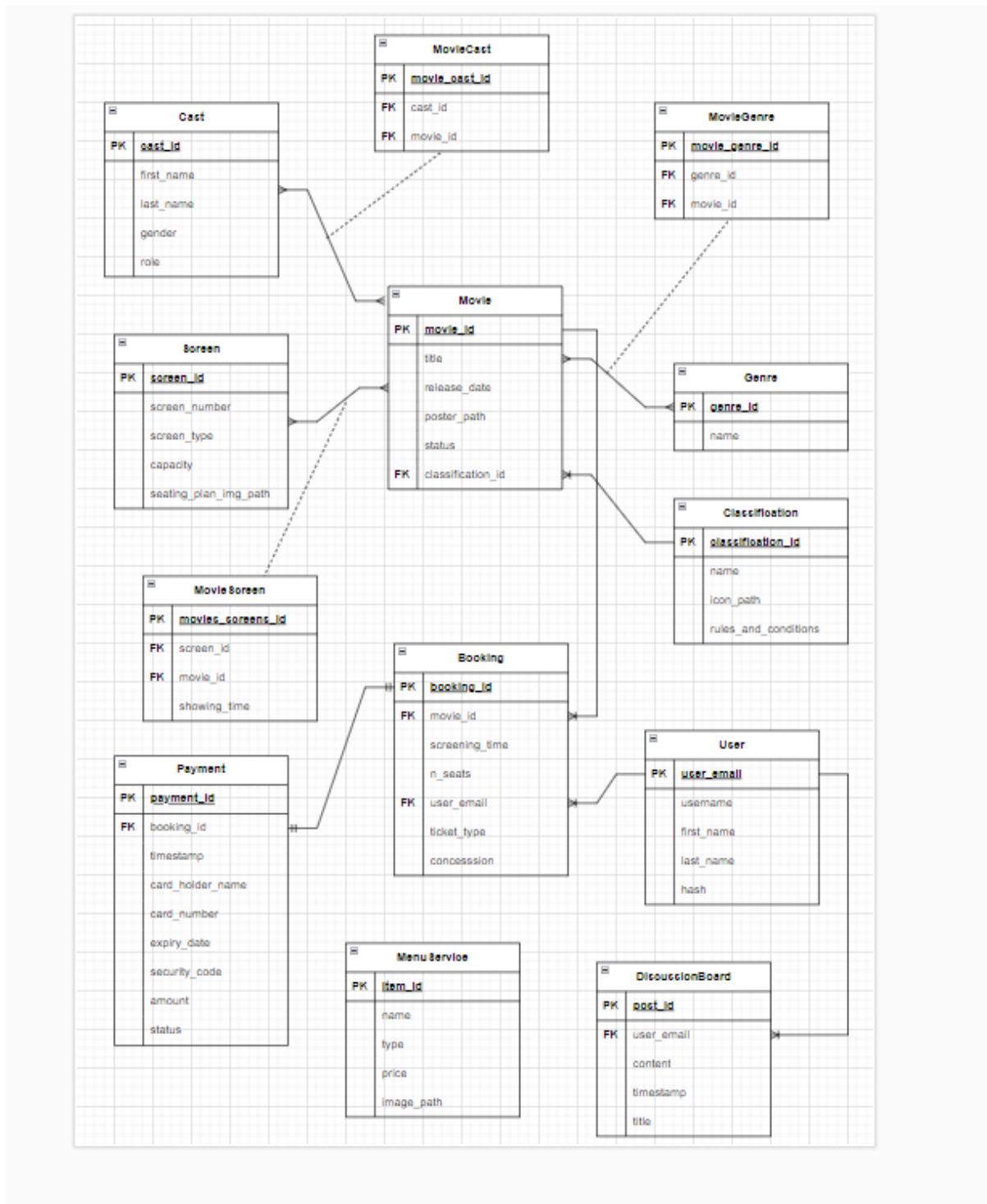
Use the reverse engineering feature provided by MySQL workbench to verify the creation of our models and compare constraints and relationships with the latest ERD and design choices



The screenshot shows a code editor with two tabs open: `routes.py` and `routes.sv`. The `routes.py` tab contains Python code defining routes for a Flask application. The `routes.sv` tab contains shell commands for booting the application.

```
routes.py (C:\Users\noben\Documents\MEGA\work\AMS\SHIA-final-team-project\CodeBusters)
application > routes > routes.py ...
1 from flask import render_template, url_for, flash, redirect
2 from application import app, bcrypt
3 from application.modules.form import Login, Registration, Checkout, Payment, Search
4 from application.modules.models import User
5 import re
6
7 @app.route("/")
8 @app.route("/home")
9 def home():
10     return render_template("home.html")
11
12 @app.route("/about")
13 def about():
14     return render_template("about.html")
15
16 @app.route("/movies")
17 def movies():
18     return render_template("movies.html")
19
20 @app.route("/classification")
21 def classification():
22     return render_template("classification.html")
23
24 @app.route("/login", methods=["GET", "POST"])
25 def login():
26     form = Login()
27
28     if form.validate_on_submit():
29         user = User.query.filter_by(username=form.username.data).first()
30         if user and bcrypt.check_password_hash(user.password, form.password.data):
31             flash("Login successful!", "success")
32             return redirect(url_for("home"))
33
34         else:
35             flash("Login unsuccessful. Please check username and password.", "danger")
36
37     return render_template("login.html", form=form)
38
39
40 @app.route("/register", methods=["GET", "POST"])
41 def register():
42     form = Registration()
43
44     if form.validate_on_submit():
45         existing_user = User.query.filter(
46             (User.user_email == form.email.data) | (User.username == form.username.data))
47
48         if existing_user:
49             flash("Email address is already registered. Please sign in or choose a different email address.", "warning")
50
51
52 @app.route("/booking")
53 def booking():
54     return render_template("booking.html")
55
56
57 @app.route("/movie")
58 def movie():
59     return render_template("movie.html")
60
61
62 @app.route("/classification")
63 def classification():
64     return render_template("classification.html")
65
66
67 @app.route("/about")
68 def about():
69     return render_template("about.html")
70
71
72 @app.route("/movies")
73 def movies():
74     return render_template("movies.html")
75
76
77 @app.route("/login")
78 def login():
79     form = Login()
80
81     if form.validate_on_submit():
82         user = User.query.filter_by(username=form.username.data).first()
83         if user and bcrypt.check_password_hash(user.password, form.password.data):
84             flash("Login successful!", "success")
85             return redirect(url_for("home"))
86
87         else:
88             flash("Login unsuccessful. Please check username and password.", "danger")
89
90     return render_template("login.html", form=form)
91
92
93 @app.route("/register")
94 def register():
95     form = Registration()
96
97     if form.validate_on_submit():
98         existing_user = User.query.filter(
99             (User.user_email == form.email.data) | (User.username == form.username.data))
100
101         if existing_user:
102             flash("Email address is already registered. Please sign in or choose a different email address.", "warning")
103
104             if existing_user.username == form.username.data:
105                 flash("Username is already registered. Please choose a different username.", "warning")
106
107             if existing_user.user_email == form.email.data:
108                 flash("Email address is already registered. Please sign in or choose a different email address.", "warning")
109
110             return redirect(url_for("register"))
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
238
239
239
240
241
242
243
244
245
246
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
367
368
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
667
668
669
669
670
671
672
673
674
675
676
677
678
678
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
828
829
829
830
831
832
833
834
835
836
837
838
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
915
916
916
917
918
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
946
947
948
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
966
967
968
968
969
969
970
971
972
973
974
975
976
977
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1015
1016
1017
1017
1018
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1026
1027
1028
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1036
1037
1038
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1046
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1056
1057
1058
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1066
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1076
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1105
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1115
1115
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1124
1125
1125
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1136
1137
1138
1138
1139
1139
1140
1141
1142
1143
1144
1145
1145
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1156
1157
1158
1158
1159
1159
1160
1161
1162
1163
1164
1165
1165
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1174
1175
1175
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1185
1185
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1194
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1204
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1214
1215
1215
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1225
1225
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1235
1235
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1245
1245
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1255
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1264
1265
1265
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1274
1275
1275
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1285
1285
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1294
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1304
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1314
1315
1315
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1325
1325
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1335
1335
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1345
1345
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1355
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1364
1365
1365
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1374
1375
1375
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1385
1385
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1394
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1404
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1414
1415
1415
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1425
1425
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1435
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1445
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1455
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1464
1465
1465
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1475
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1485
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1495
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1504
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1514
1515
1515
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1525
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1535
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1545
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1555
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1565
1565
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1575
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1585
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1595
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1604
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1614
1615
1615
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1625
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1635
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1645
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1655
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1665
1665
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1675
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1685
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1695
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1704
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1714
1715
1715
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1725
1726
1727
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1735
1736
1737
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1745
1746
1747
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1755
1756
1757
1757
1758
1758
1759
1759
1760
1761
1762
1763
1764
1765
1765
1766
1767
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1775
1776
1777
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1785
1786
1787
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1795
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1804
1805
1806
1806
1807
1807
1808
1808
1809
1809
1810
1811
1812
1813
1814
1815
1815
1816
1817
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1825
1826
1827
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1835
1836
1837
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1845
1846
1847
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1855
1856
1857
1857
1858
1858
1859
1859
1860
1861
1862
1863
1864
1865
1865
1866
1867
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1875
1876
1877
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1885
1886
1887
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1904
1905
1906
1906
1907
1907
1908
1908
1909
1909
1910
1911
1912
1913
1914
1915
1915
1916
1917
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1925
1926
1927
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1935
1936
1937
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1945
1946
1947
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1955
1956
1957
1957
1958
1958
1959
1959
1960
1961
1962
1963
1964
1965
1965
1966
1967
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1975
1976
1977
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1985
1986
1987
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2004
2005
2006
2006
2007
2007
2008
2008
2009
2009
2010
2011
2012
2013
2014
2015
2015
2016
2017
2017
2018
2018
2019
2019
2020
2021
2022
2023
2024
2025
2025
2026
2027
2027
2028
2028
2029
2029
2030
2031
2032
2033
2034
2035
2035
2036
2037
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2045
2045
```

More code refactoring needed

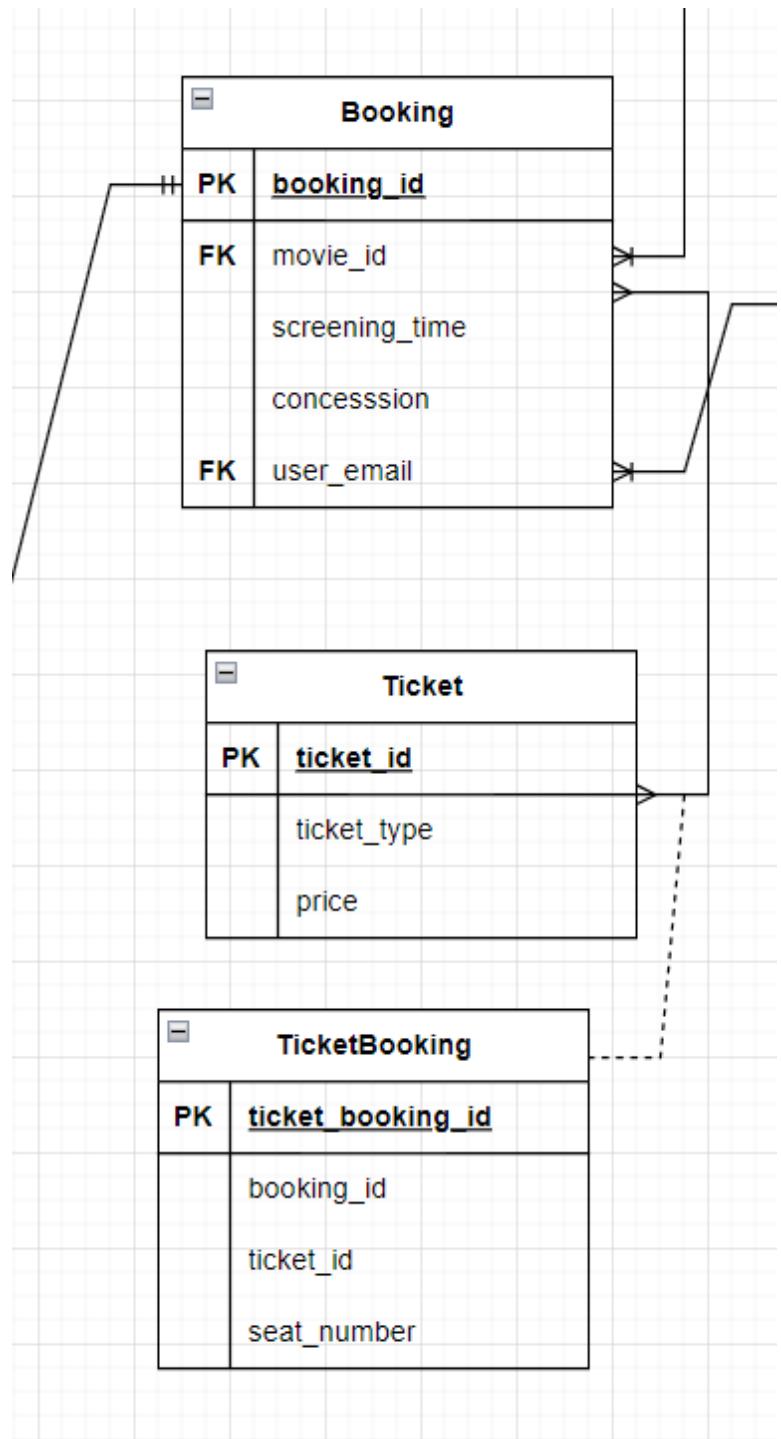


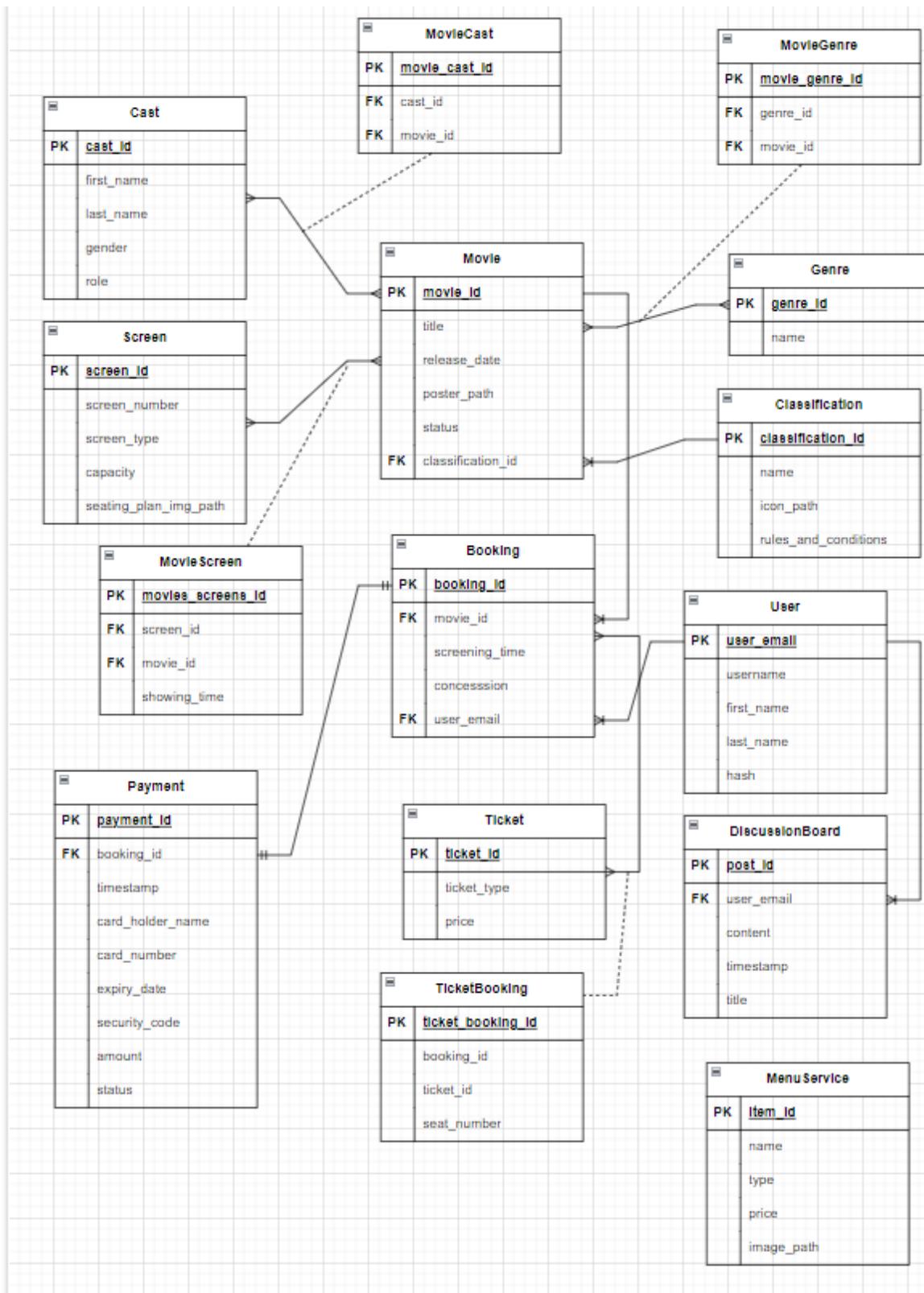
Add title to DiscussionBoard table → Done

**PROBLEM:** We actually do not store the price of the tickets anywhere in the database

**PROBLEM 2:** `n_seats` in `Booking` table is quite redundant

→ moreover, new field `seat_number` can be created after correctly managing tickets (types and prices)





Updated ERD to latest version

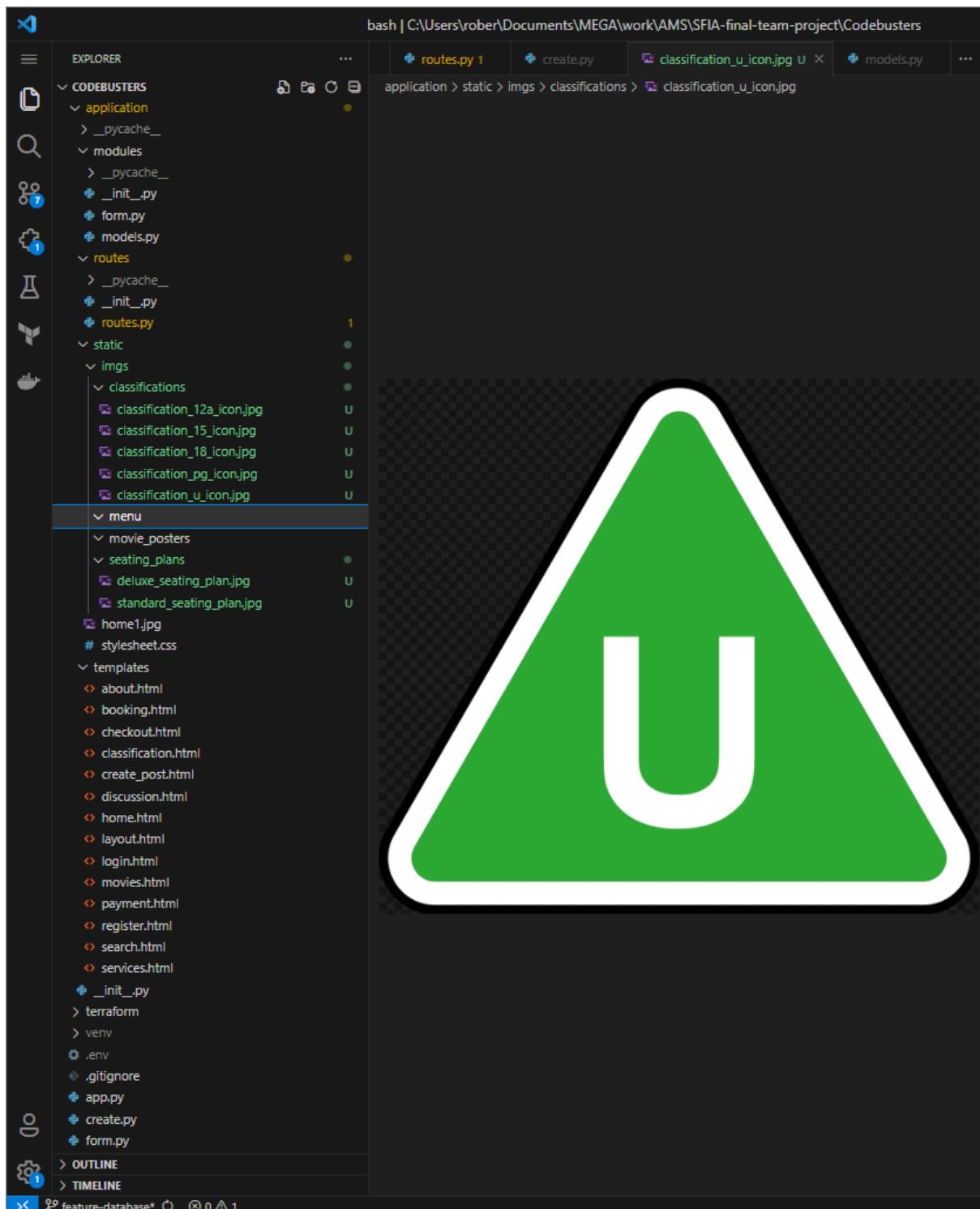
```

17     booking = db.relationship("Booking", backref="user")
18     discussion_board = db.relationship("DiscussionBoard", backref="user")
19
20
21 class Booking(db.Model):
22     booking_id = db.Column(db.Integer, primary_key=True)
23     movie_id = db.Column(db.Integer, db.ForeignKey("movie.movie_id"), nullable=False)
24     screening_time = db.Column(db.DateTime, nullable=False)
25     user_email = db.Column(
26         db.String(30), db.ForeignKey("user.user_email"), nullable=False
27     )
28     concession = db.Column(db.Boolean, nullable=False)
29     payment = db.relationship("Payment", backref="booking")
30     ticket_booking = db.relationship("TicketBooking", backref="booking")
31
32
33 class Ticket(db.Model):
34     ticket_id = db.Column(db.Integer, primary_key=True)
35     ticket_type = db.Column(db.String(30), nullable=False)
36     price = db.Column(db.Float, nullable=False)
37     ticket_booking = db.relationship("TicketBooking", backref="ticket")
38
39
40 class TicketBooking(db.Model):
41     ticket_booking_id = db.Column(db.Integer, primary_key=True)
42     ticket_id = db.Column(db.Integer, db.ForeignKey("ticket.ticket_id"), nullable=False)
43     booking_id =
44         db.Column(db.Integer, db.ForeignKey("booking.booking_id"), nullable=False)
45     seat_number = db.Column(db.String(10), nullable=False)
46
47
48
49 class Movie(db.Model):
50     movie_id = db.Column(db.Integer, primary_key=True)
51     title = db.Column(db.String(30), nullable=False)
52     release_date = db.Column(db.DateTime, nullable=False)
53     poster_path = db.Column(db.String(255), nullable=False)
54     status = db.Column(db.String(20), nullable=False)
55     classification_id = db.Column(

```

Also updated classes as well as relationships and backrefs

Performed full refactoring of the mock database creation using `create.py`. Now, the database contains a comprehensive amount of data absolutely needed for the site to work and meet the MVP requirements.

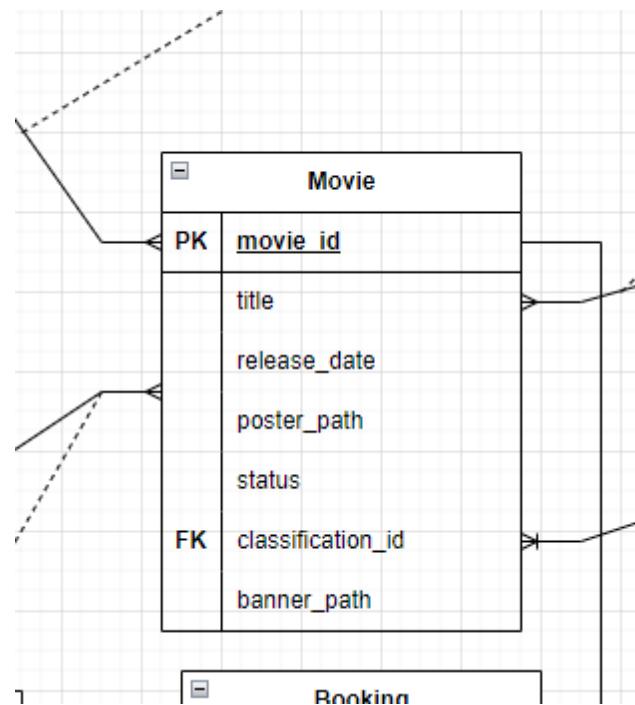


Working on retrieving and referencing all the images which are needed for the frontend (paths stored in the database for easy retrieval as well as other technical reasons) - menu, classifications, movie posters, seating plans

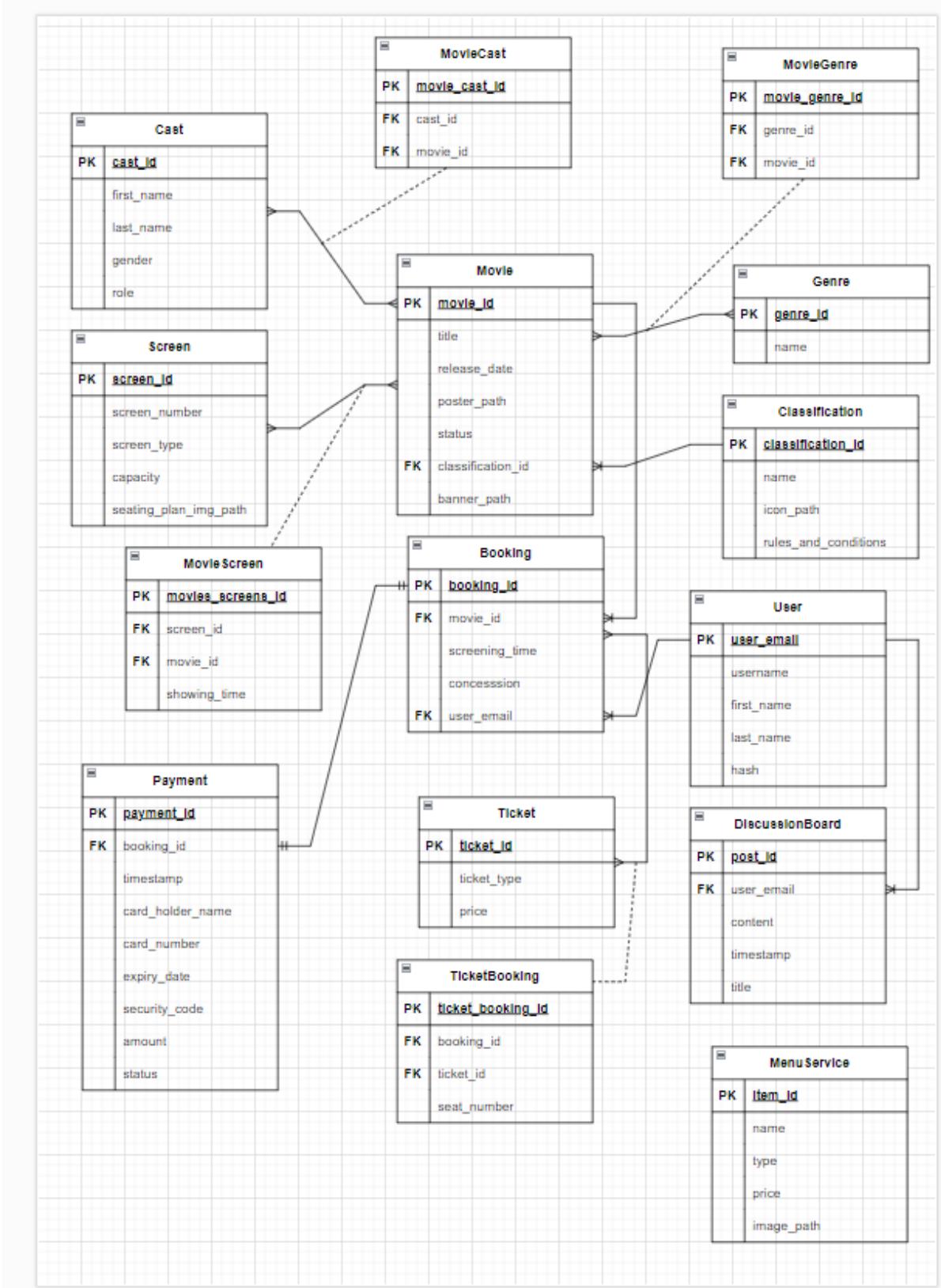
The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'qa\_cinema' is selected. The Schemas list includes tables such as movie, poster, user, screen, payment, movie\_screen, movie\_genre, movie\_cast, movie, menu\_service, genre, discussion\_board, classification, cast, and bookings. The Results Grid pane shows a table with columns: movie\_id, title, release\_date, poster\_path, status, and classification\_id. The SQL Editor pane contains a query for movie 1.

Verifying with MySQL workbench whether everything was added and works as expected

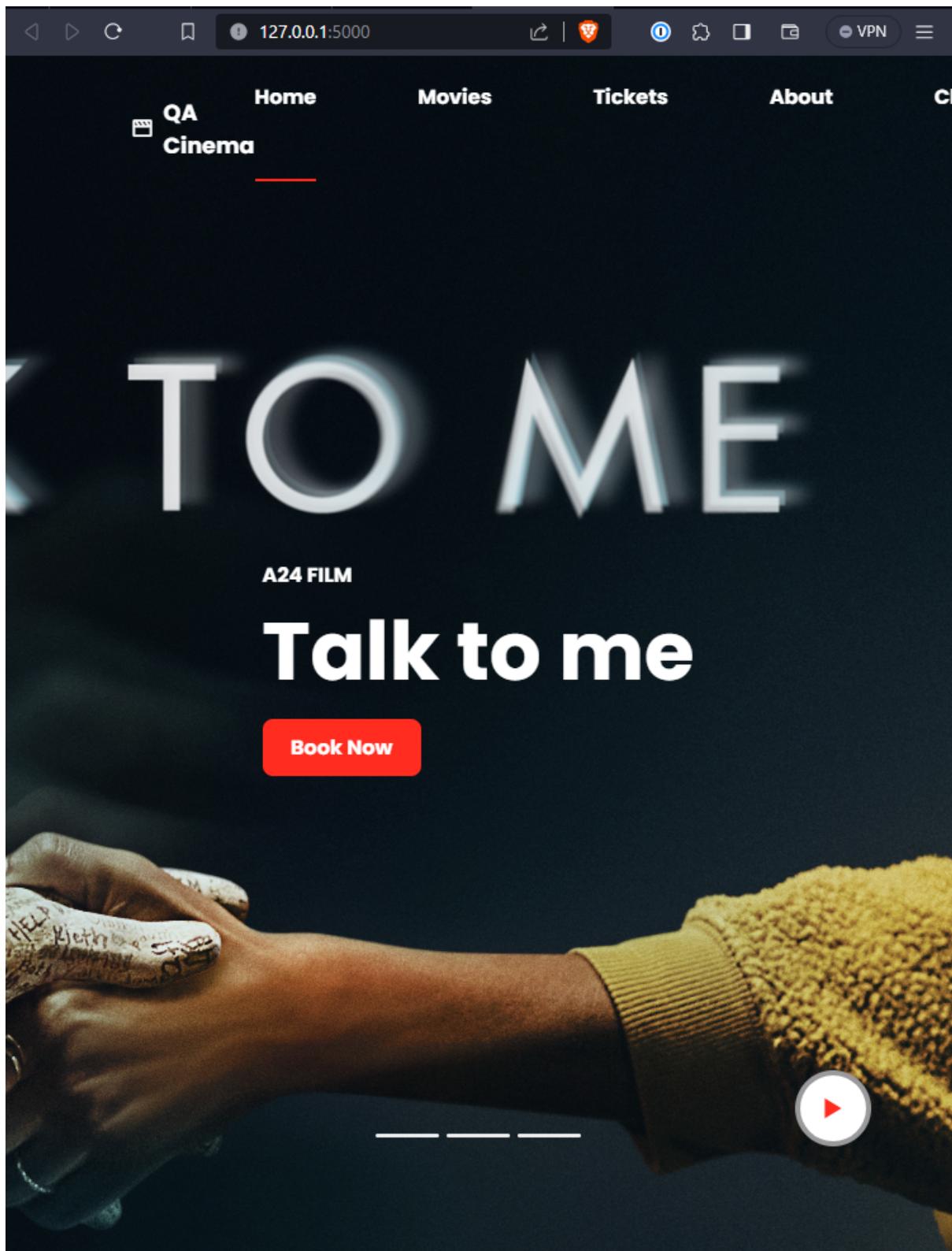
**PROBLEM:** Might be useful to have a banner path in the Movie database (mainly for aesthetic purposes)



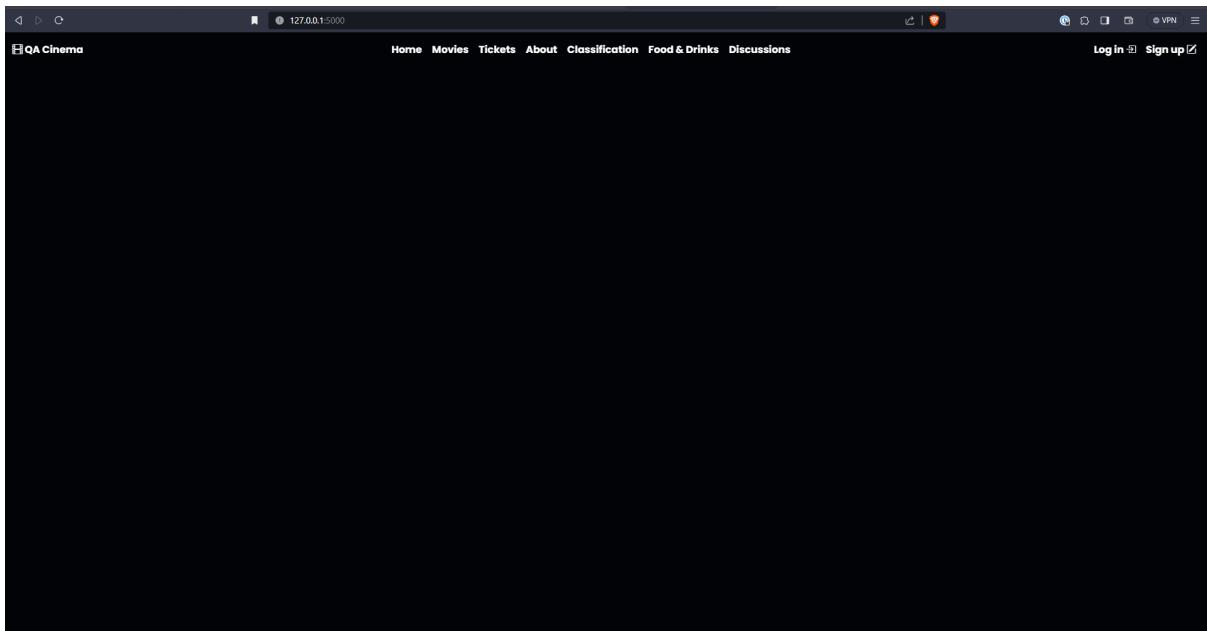
Added banner\_path field to Movie table



Updated ERD: draft 4



Started working on refactoring of certain elements related to the front end (i.e., making navbar responsive, home page design, displaying movies from the mock database on the home page)



Front-end refactoring in progress

The screenshot shows a code editor with the file `home.html` open. The code is as follows:

```

<application> <templates> <home.html> </templates> </application>
  application > templates > home.html > script
  1  {% extends "layout.html" %}
  2  {%-- Setting title of the page -->
  3  (% block title %) Home (% endblock title%)
  4  (% block head %) All the links and styles needed for the page -->
  5  (% block links_refs %)
  6  <link
  7   rel="stylesheet"
  8   href="https://cdn.jsdelivr.net/npm/swiper@10/swiper-bundle.min.css"/>
  9  (% endblock links_refs %)
 10  (% block content%)
 11  <!-- Displaying main content of the page -->
 12  (% block content%)
 13  <section class="home_swiper" id="home">
 14    <div class="swiper-slide">
 15      <div class="container">
 16        
 19      </div>
 20    </div>
 21    <div class="home_text">
 22      <span>Detective Comics</span>
 23      <h1>Blue beetle</h1>
 24      <a href="((url_for('booking')))" class="btn">Book Now</a>
 25      <div class="play">
 26        <i class="bx bx play"></i>
 27        <a href="((url_for('booking')))">Book Now</a>
 28      </div>
 29    </div>
 30    <!-- Add more swiper-slide elements as needed -->
 31    <div class="swiper-slide container">
 32      
 35    </div>
 36  </div>
 37  
```

To the right of the code editor, the QA Cinema homepage is displayed. It features a large banner for the movie "Blue beetle". The banner includes the text "DETECTIVE COMICS" and "Blue beetle". A red "Book Now" button is visible below the banner.

Home page refactoring continued

```

<section class="swiper-slide" %>
    <div style="background-image: url('{{ url_for('static', filename='imgs/movie_banners/' + movie.banner_path) }}'); background-size: cover; background-position: center;>
        <div class="custom-overlay"></div>
        <div class="container-fluid">
            <div class="home-text">
                <h1>{{ movie.title }}</h1>
                <p>{{ movie.release_date.strftime('%Y') }}</p>
                <!-- Need to pass the ID of the movie to the booking page -->
                <a href="{{ url_for('booking', movie_id=movie.movie_id) }}" class="btn">Buy tickets</a>
            </div>
        </div>
    </div>
<% endfor %>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<% endblock content %>
<!-- Can add scripts specific to this template here -->
<% block scripts %>
<script src="https://cdn.jsdelivr.net/npm/swiper@10/swiperBundle.min.js"></script>
<script src="{{ url_for('static', filename='js/swiper.js') }}></script>

```

Home page refactoring complete?

```

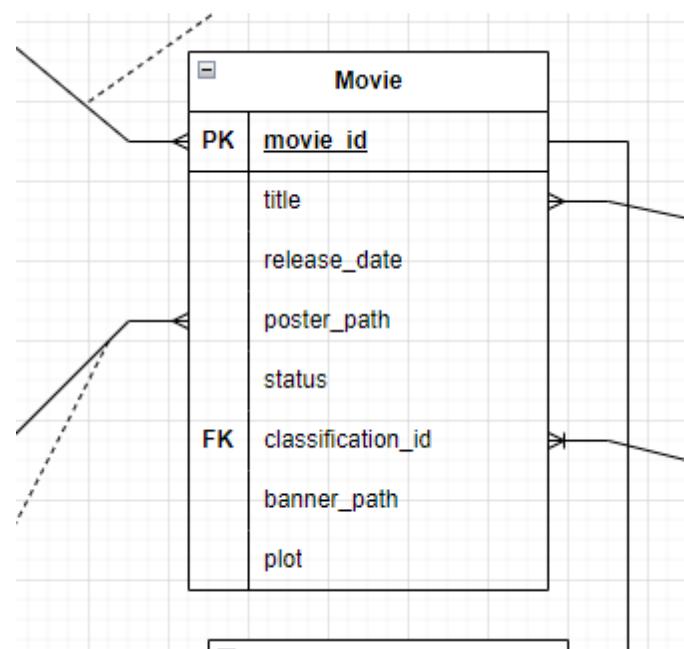
<div class="swiper-slide" %>
    <div class="banner">
        <div class="swiperSwiper">
            <!-- Additional required wrapper -->
            <div class="swiper-wrapper">
                <div class="swiper-slide">
                    <img alt="https://image.tmdb.org/t/p/original/l1BQgactT3HKEsgBckQlyrATVelt.jpg" />
                </div>
            </div>
        </div>
    </div>
<% endblock content %>

```

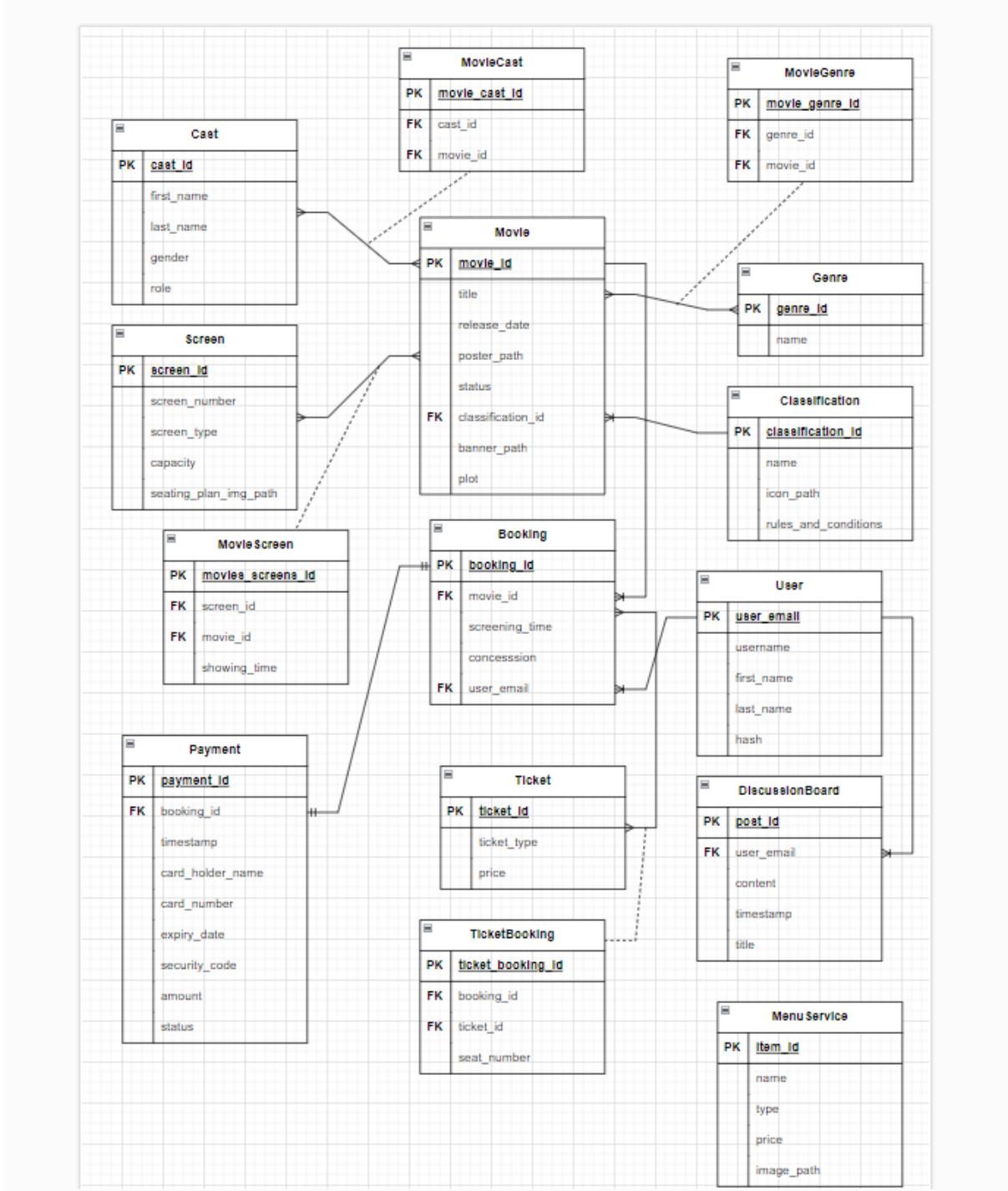
Started working on refactoring movie listings page

**PROBLEM:** Movie table does not have a `plot` field

- need to update ERD
- as well as models and create.py



Added to ERD



```

class Movie(db.Model):
    movie_id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(30), nullable=False)
    release_date = db.Column(db.DateTime, nullable=False)
    poster_path = db.Column(db.String(255), nullable=False)
    banner_path = db.Column(db.String(255), nullable=False)
    status = db.Column(db.String(20), nullable=False)
    plot = db.Column(db.Text, nullable=False)
    classification_id = db.Column(
        db.Integer, db.ForeignKey("classification.classification_id"), nullable=False
    )
    movie_genre = db.relationship("MovieGenre", backref="movie")
    movie_cast = db.relationship("MovieCast", backref="movie")
    movie_screen = db.relationship("MovieScreen", backref="movie")
    booking = db.relationship("Booking", backref="movie")

```

Added plot field to Movie table + movie plots data to database within the create.py file

```

application > static > css > stylesheet.css
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
CCCCCCCC HEAD application/static/css/stylesheet.css (Current Change)

/*
 * Basic styling setup
 */
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@700&display=swap');

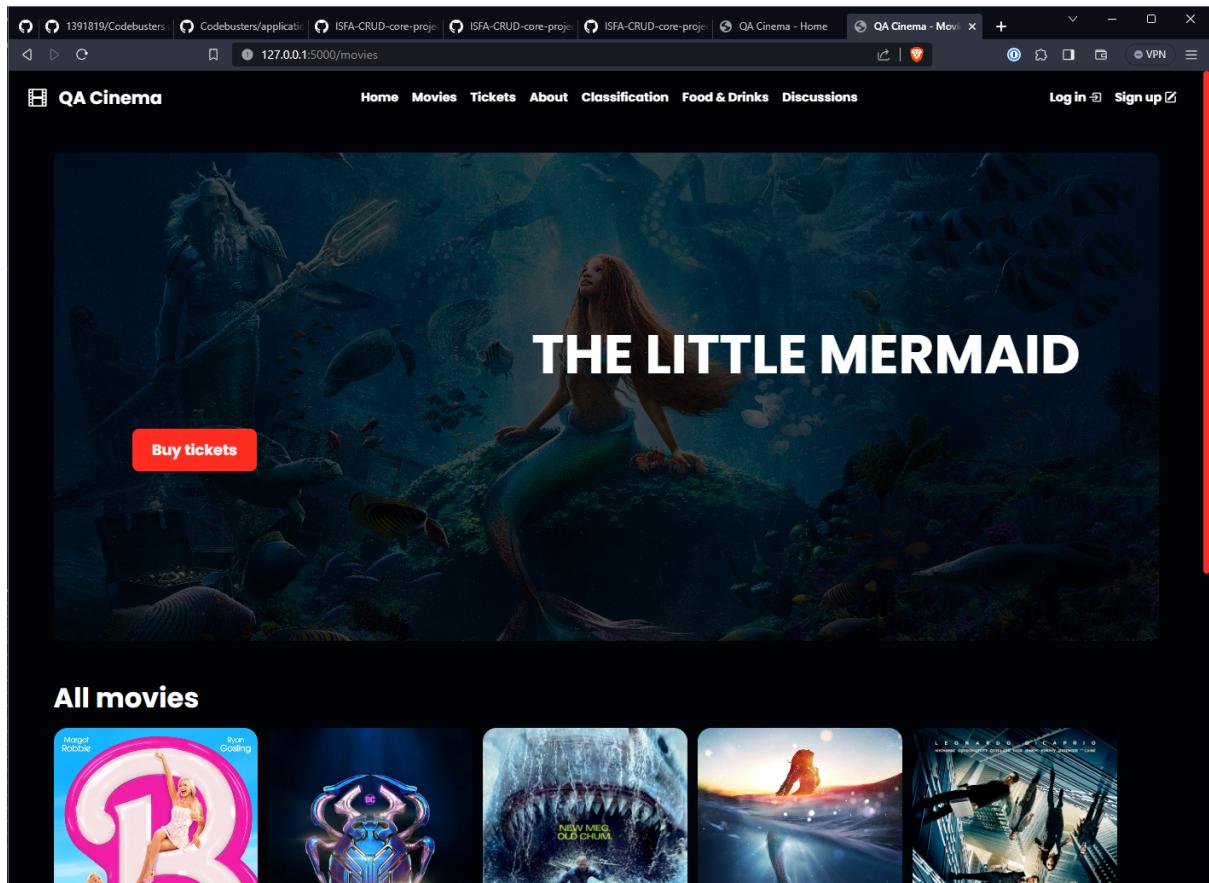
html, body {
    font-family: 'Poppins', sans-serif;
    scroll-padding-top: zrem;
    scroll-behavior: smooth;
}
/* Specifies a prioritized list of font family names or generic family names. A user agent iterates through the list of family names until it matches an available font that contains a glyph for the character to be rendered.
   (Edge 12, Firefox 1, Safari 1, Chrome 1, IE 3, Opera 3)
*/
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    list-style-type: none;
    text-decoration-line: none;
    scroll-padding-top: zrem;
    scroll-behavior: smooth;
}

/* Main styles
   https://www.w3.org/Style/Examples/007/scrollbar.html
   --main-color:#f2f2f2;
   --text-color:#000000;
   --bg-color:#fff;
   ::-webkit-scrollbar {
       width: 0.5rem;
       background: var(--text-color);
   }
   ::-webkit-scrollbar-thumb {
       background: var(--main-color);
       border-radius: 5rem;
   }
*/
body {
    --main-color:#f2f2f2;
    --text-color:#000000;
    --bg-color:#fff;
}
::-webkit-scrollbar {
    width: 0.5rem;
    background: var(--text-color);
}
::-webkit-scrollbar-thumb {
    background: var(--main-color);
    border-radius: 5rem;
}

/* Form styling
   https://www.w3.org/Style/Examples/007/scrollbar.html
   deleted: form.py
   modified: requirements.txt
*/
form {
    border: 1px solid #ccc;
    padding: 10px;
    width: 300px;
    margin: 0 auto;
}
input[type="text"] {
    width: 100%;
    height: 30px;
    border: 1px solid #ccc;
    padding: 5px;
}
input[type="submit"] {
    width: 100px;
    height: 30px;
    border: 1px solid #ccc;
    background-color: #f0f0f0;
    color: #000000;
    border-radius: 5px;
    cursor: pointer;
}

```

Merging of entire work done by everyone > Merging complete



Movie page refactoring in progress. See below for more

Major refactoring: Decided on creating a separate movie details page for each movie instead of displaying everything in the /movies page

The terminal shows the following Python code for a movie details route:

```

application > routes > routes.py > movie_details
# retrieving genres of the movie
genres = ", ".join(
    [
        genre.name
        for genre in Genre.query.join(MovieGenre)
            .filter(MovieGenre.movie_id == movie_id)
            .all()
    ]
)

# retrieving cast of the movie
directors = ", ".join(
    [
        f'{cast.first_name} {cast.last_name}'
        for cast in Cast.query.join(MovieCast)
            .filter(MovieCast.movie_id == movie_id)
            .filter(Cast.role == "Director")
            .all()
    ]
)

actors = ", ".join(
    [
        f'{cast.first_name} {cast.last_name}'
        for cast in Cast.query.join(MovieCast)
            .filter(MovieCast.movie_id == movie_id)
            .filter(Cast.role == "Actor")
            .all()
    ]
)

# rendering appropriate template
return render_template(
    "movie_details.html",
    movie=movie,
    genres=genres,
    directors=directors
)

```

The browser shows the movie details page for 'The Little Mermaid' with the following information:

- Directors:** John Musker, Ron Clements
- Actors:** Halle Bailey, Jonah Hauer-King, Javier Bardem
- Release year:** 2023
- Status:** Released
- Genres:** Romance, Musical
- Classification:** PG
- Plot:** The Little Mermaid follows a mermaid princess Ariel who is fascinated by the human world and makes a deal with a treacherous sea witch Ursula to trade her voice for human legs to impress Prince Eric, whom she saved from a shipwreck before time runs out.

The screenshot shows a movie booking application interface. On the left, there's a code editor with Python code for a movie classification page. The main area displays a movie poster for 'Barbie' with the release date 'July 21'. Below the poster, a dropdown menu lists various showtimes and ticket types. A red button at the bottom right says 'Buy tickets'.

```

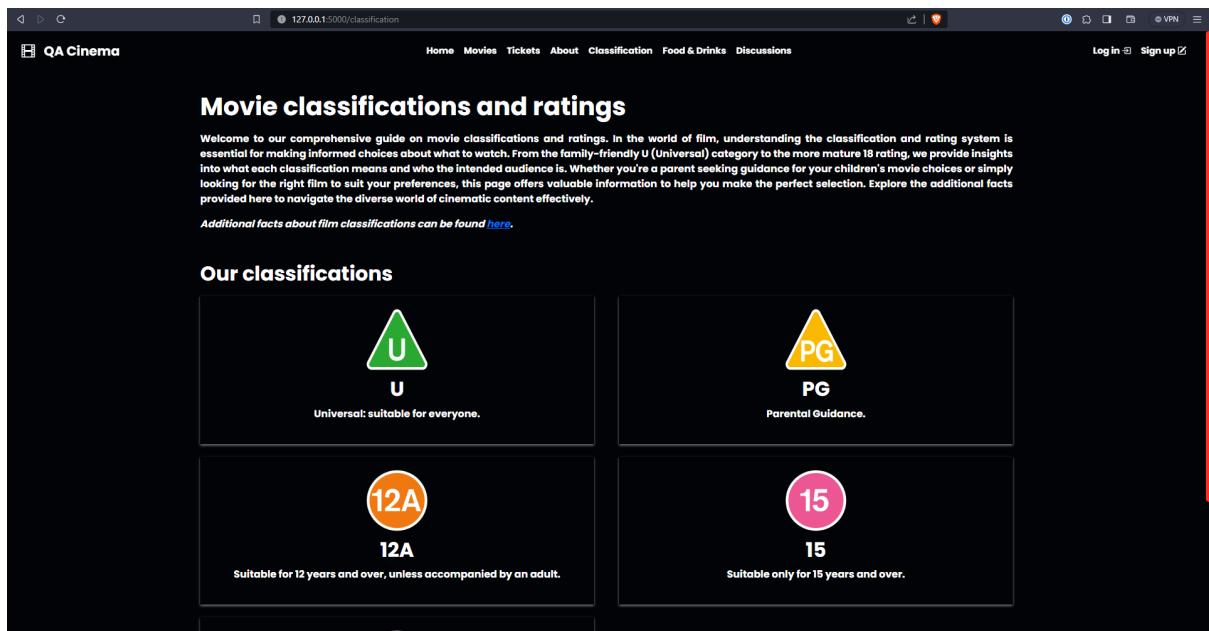
    <h2>Directors</h2>
    <p>{{ directors }}</p>
    <h2>Actors</h2>
    <p>{{ actors }}</p>
    <h2>Release year</h2>
    <p>{{ movie.release_date.strftime('XX') }}</p>
    <h2>Status</h2>
    <p>{{ movie.status }}</p>
    <h2>Genres</h2>
    <p>{{ genres }}</p>
    <h2>Classification</h2>
    
    <p class="small">Read more on movie classifications <a href="/classification">here</a>.</p>
    </div>
    <h2>Plot</h2>
    <p>{{ movie.plot }}</p>
    <h2>Showing times</h2>
    <!-- Displaying all showing times -->
    <select id="showing-time" name="showing-time">
        <% for time in showing_times %>
            <option value="{{ time }}">{{ time }}</option>
        <% endfor %>
    </select>
    <a href="{{ url_for('booking', movie_id=movie.movie_id) }}"
        class="button">Buy tickets</a>

```

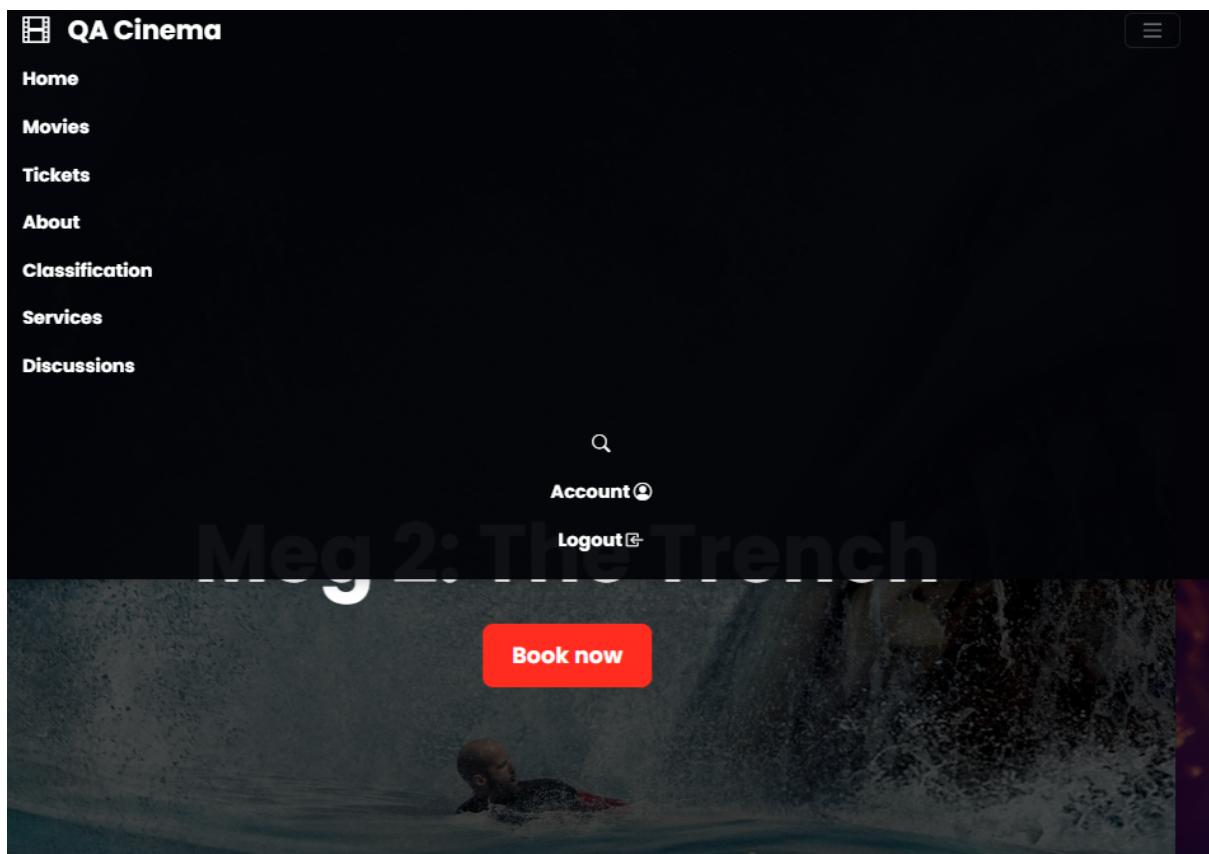
Adding more functionalities to meet the requirements set in the MVP and performing continuous refactoring

The screenshot shows a classification page with a sidebar containing movie ratings: U, PG, 12A, 15, and 18. The main content area displays a list of movies with their posters and classification details. The URL in the browser bar is 127.0.0.1:5000/classification.

Refactoring of classification page



Refactoring of the classification page completed



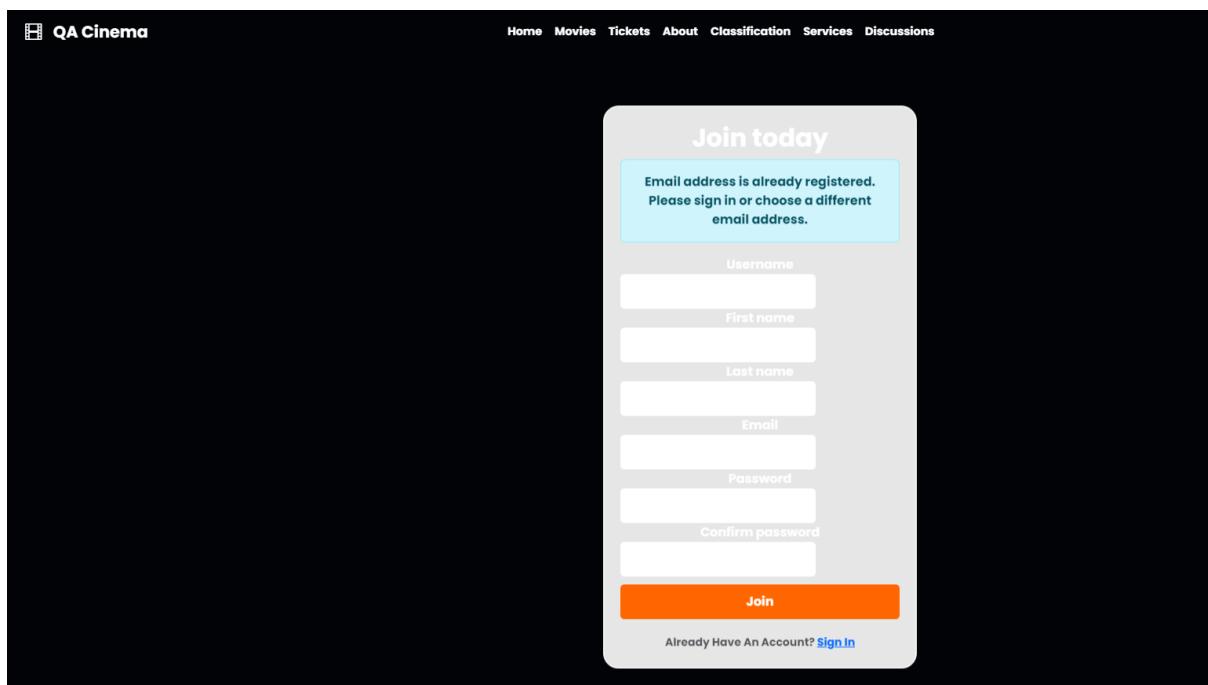
Refactoring of front-end minor elements (i.e., margins, icons, etc)

## AttributeError

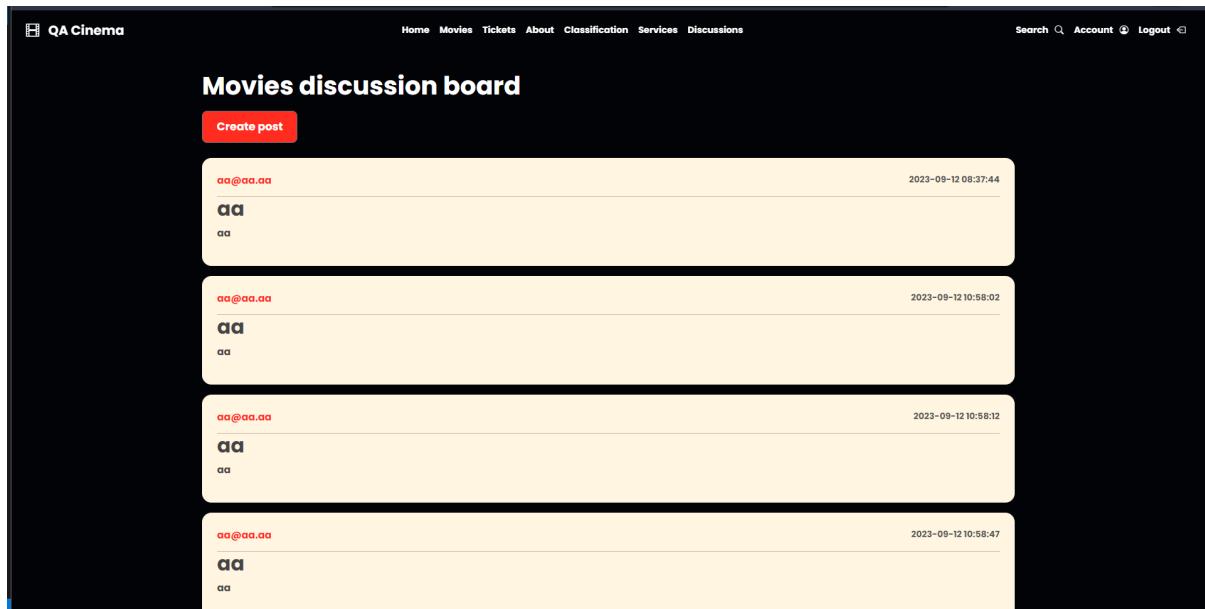
```
AttributeError: 'User' object has no attribute 'email'
```

```
Traceback (most recent call last)
```

```
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 2213, in __call__
    return self.wsgi_app(environ, start_response)
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 2193, in wsgi_app
    response = self.handle_exception(e)
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 2190, in wsgi_app
    response = self.full_dispatch_request()
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 1486, in full_dispatch_request
    rv = self.handle_user_exception(e)
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 1484, in full_dispatch_request
    rv = self.dispatch_request()
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\venv\Lib\site-packages\flask\app.py", line 1469, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
           ^^^^^^^^^^
File "C:\Users\rober\Documents\MEGA\work\AMS\SFIA-final-team-project\Codebusters\application\routes\voutes.py", line 174, in register
    if existing_email_user.email == form.email.data:
           ^^^^^^^^^^
AttributeError: 'User' object has no attribute 'email'
```



Fixing if statement bug on email field (should have been user\_email)



Refactoring discussion board front end

```

310
311
312 @app.route("/discussion")
313 def discussion():
314     # return all posts from newest to oldest
315     posts = DiscussionBoard.query.order_by(desc(DiscussionBoard.timestamp)).all()
316     return render_template("discussion.html", posts=posts)
317
318

```

Refactored discussion board back end to get posts from newest to oldest

```

353
354
355 @app.route("/discussion/new_post", methods=["GET", "POST"])
356 @login_required
357 def create_new_post():
358     form = CreatePosts()
359
360     if form.validate_on_submit():
361         current_time = datetime.utcnow()
362
363         post = DiscussionBoard(
364             title=form.title.data,
365             content=form.content.data,
366             user_email=current_user.user_email,
367             timestamp=current_time,
368         )
369         db.session.add(post)
370         db.session.commit()
371
372         # TODO: This message could be displayed if the client wants it in future iterations
373         # flash("Your post has been created!", "success") # debug
374         return redirect(url_for("discussion"))
375     # else:
376     #     flash("You must be logged in to create a post.", "danger")
377     # return render_template("create_post.html", title="New Post", form=form)
378
379
380     # passign stuff to navbar
381
382

```

Refactoring create new post route - back end

```

@app.route("/discussion/new_post", methods=["GET", "POST"])
@login_required
def create_new_post():
    form = CreatePosts()

    if form.validate_on_submit() and request.method == "POST":
        # getting current time
        current_time = datetime.now()

        # creating post in the db
        post = DiscussionBoard()
        title=form.title.data,
        content=form.content.data,
        user_email=current_user.user_email,
        timestamp=current_time,
    )

        # committing to db
        db.session.add(post)
        db.session.commit()

        # TODO: This message could be displayed if the client wants it in future iterations
        # flash("Your post has been created!", "success") # debug
        return redirect(url_for("discussion"))
    # else:
    #     flash("You must be logged in to create a post.", "danger")
    return render_template("create_post.html", title="New Post", form=form)

```

Refactoring complete

```

5
6 @app.route("/logout")
7 def logout():
8     # store current url in the session
9     session["previous_url"] = request.referrer
0
1
2     # redirect user to previous URL they were on
3     previous_url = session.pop("previous_url", None)
4
5     if previous_url:
6         return redirect(previous_url)
7     else:
8         # if there is no previous URL, redirect to the homepage
9         return redirect(url_for("home"))
0

```

Updated logout button to redirect to previous URL if there was any

The screenshot shows the 'Account' section of the QA Cinema website. At the top, there's a navigation bar with links for Home, Movies, Booking, About, Classification, Services, and Discussions. On the right side of the header are search, account, and logout buttons. The main content area is titled 'Account' and contains fields for 'Username' (aa), 'Email' (aa@aa.aa), 'First name' (aa), and 'Last name' (aa). The background of the page is black.

Updated the account template to display basic user information

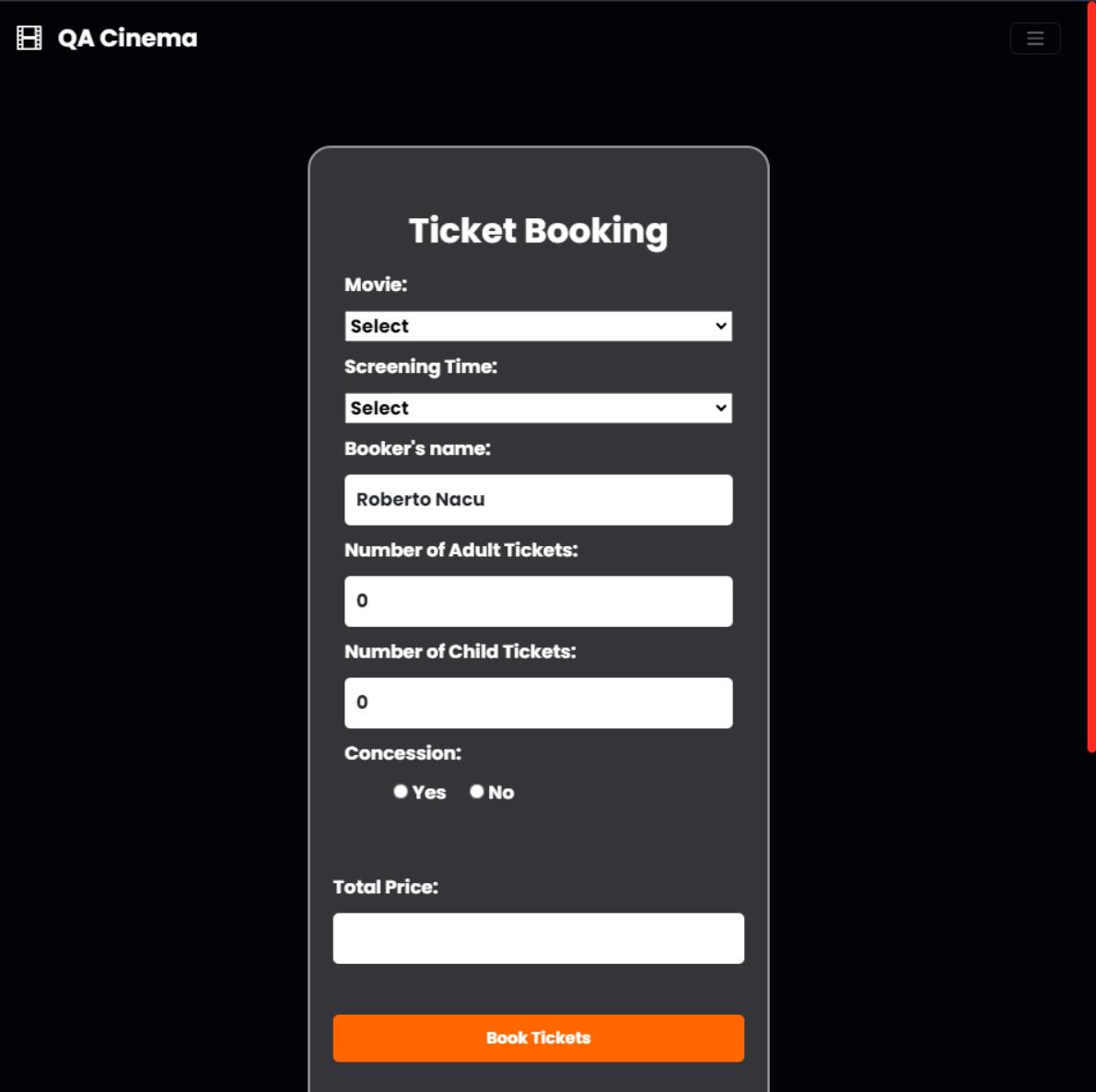
The screenshot shows the 'Movies discussion board' section of the QA Cinema website. At the top, there's a navigation bar with links for Home, Movies, Booking, About, Classification, Services, and Discussions. On the right side of the header are search, account, and logout buttons. The main content area is titled 'Movies discussion board' and features a 'Create post' button. Below it, there are four discussion posts from different users:

- u.r.s.tupid** (aa) posted at 2023-09-12 14:59:36
- aastupi.daa** (aa) posted at 2023-09-12 14:59:26
- normal** (stupi) posted at 2023-09-12 14:58:00
- 1** (aa@aa.aa) posted at 2023-09-12 13:32:53

The background of the page is black.

Testing limits of content moderation in discussion board

Working on booking + payment page/CRUD functionality



A screenshot of a mobile application interface titled "Ticket Booking". The background is black, and the form is contained within a light gray rounded rectangle. At the top center, the title "Ticket Booking" is displayed in bold white font. Below the title are several input fields and controls:

- Movie:** A dropdown menu with the placeholder "Select".
- Screening Time:** A dropdown menu with the placeholder "Select".
- Booker's name:** An input field containing the text "Roberto Nacu".
- Number of Adult Tickets:** An input field containing the number "0".
- Number of Child Tickets:** An input field containing the number "0".
- Concession:** A radio button group with two options: "Yes" and "No".
- Total Price:** An input field that is currently empty.

At the bottom of the form is a large orange button labeled "Book Tickets".

Current status: design is there, some functionality as well → need to complete this and check if CRUD functionality works

Work in progress

The screenshot displays the QA Cinema application's user interface and its underlying code structure.

**Application Structure:**

- application
- ↳ \_\_pycache\_\_
- ↳ modules
- ↳ \_\_pycache\_\_
- ↳ movies
- ↳ payment
- ↳ routes
- ↳ static
- ↳ css
- ↳ stylesheet.css
- ↳ img
- ↳ classifications
- ↳ menu
- ↳ menu\_hot\_dogs.py
- ↳ menu\_ic\_crea.py
- ↳ menu\_nachos.py
- ↳ menu\_pasta.py
- ↳ menu\_salad.py
- ↳ menu\_soda.py
- ↳ movie\_banners
- ↳ movie\_posters
- ↳ payment
- ↳ setting\_plan

**Booking Form (Right Side):**

Movie

Booking

Screen type: Standard

Screening time: 18.09.2023 - 19:00

Booker email: bb@bb

# Adult tickets - £14.99: 2

# Child tickets - £9.99: 0

Concession: ● Yes ○ No

Total Price: 29.98

**Console (Bottom):**

```
iR Elements Recorder Console Sources Network Performance Memory Application Security >> Default level: 1 Issue 1
```

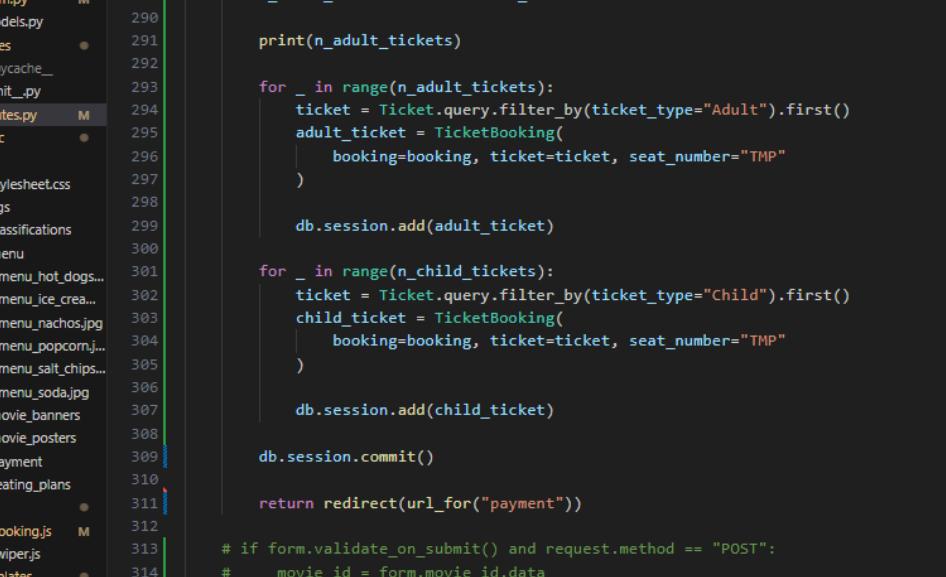
Network Requests (Bottom Left):

- GET /booking/movie\_id=1 HTTP/1.1 200
- GET /seats/available HTTP/1.1 200
- GET /seats/join\_booking/1 HTTP/1.1 304
- GET /seats/join\_booking/1 HTTP/1.1 304
- GET /seats/price(HTTP/1.1, 200)
- GET /seats/price(HTTP/1.1, 200)
- GET /get\_screening\_times/1/standard HTTP/1.1 200
- GET /get\_screening\_times/1/selected HTTP/1.1 200
- GET /get\_ticket\_prices HTTP/1.1 200
- GET /get\_tickets/1 HTTP/1.1 302
- GET /get\_tickets/1 HTTP/1.1 200
- GET /get\_tickets/1 HTTP/1.1 404
- GET /get\_tickets/1 HTTP/1.1 304
- GET /get\_tickets/1 HTTP/1.1 404
- GET /image/vina.png HTTP/1.1 404
- GET /statistic/movie\_id=1 HTTP/1.1 304
- GET /booking/movie\_id=1 HTTP/1.1 200

Got the booking form to add bookings to the Booking table

```
15 •   SELECT * FROM cast;  
16 •   SELECT * FROM booking;  
17 •   SELECT * FROM ticket;
```

Result Grid		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content	
	booking_id	movie_id	screening_time	user_email	concession				
▶	1	1	2023-11-10 00:00:00	aa@aa.aa	1				
	2	2	2023-09-18 22:00:00	bb@bb	1				
	3	1	2023-09-21 16:00:00	bb@bb	1				
	4	1	2023-09-21 16:00:00	bb@bb	1				
*	NULL	NULL	NULL	NULL	NULL				



```
# we need to keep track of the tickets and their types for each booking
# TODO: In future iterations, the seat_number field could also be used (TicketBooking tab)
n_adult_tickets = form.adult_tickets.data
n_child_tickets = form.child_tickets.data

print(n_adult_tickets)

for _ in range(n_adult_tickets):
    ticket = Ticket.query.filter_by(ticket_type="Adult").first()
    adult_ticket = TicketBooking(
        booking=booking, ticket=ticket, seat_number="TMP"
    )

    db.session.add(adult_ticket)

for _ in range(n_child_tickets):
    ticket = Ticket.query.filter_by(ticket_type="Child").first()
    child_ticket = TicketBooking(
        booking=booking, ticket=ticket, seat_number="TMP"
    )

    db.session.add(child_ticket)

db.session.commit()

return redirect(url_for("payment"))

# if form.validate_on_submit() and request.method == "POST":
#     movie_id = form.movie_id.data
#     user_email = current_user.user_email
#     concession = form.concession.data
```

Added tracking of tickets for each booking

```
12 •   SELECT * FROM discussion_posts;
13 •   SELECT * FROM classification;
14 •   SELECT * FROM cast;
15 •   SELECT * FROM booking;
16 •   SELECT * FROM ticket;
17 •   SELECT * FROM ticket_booking;
18 •   TNSFRIT INTO honking VALUES(1, "2023-11-11", "aa@aa.aa", TRUE);
19 •
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

ticket_booking_id	ticket_id	booking_id	seat_number
1	1	8	TMP
2	1	9	TMP
3	1	9	TMP
4	1	9	TMP
5	2	9	TMP
6	2	9	TMP
NULL	NULL	NULL	NULL

Administration Schemas Information ticket\_booking20 x Apply Revert

## Checking everything with workbench

The screenshot shows a developer's environment with multiple windows open:

- QA Cinema**: A browser tab displaying a movie booking interface.
- Ticket Booking**: A browser tab showing a form for ticket booking.
- routes.py**: An open Python file containing routes for movie screening times and ticket prices.
- booking.html**: An open HTML file representing the movie listing page.
- Terminal**: A terminal window showing command-line output related to the project.

The **routes.py** code includes:

- Routes for getting screening times and movie details.
- A function to filter screens by movie ID and screen type.
- A function to sort screening times by date.
- A route for getting ticket prices.

The **Ticket Booking** form contains fields for:

- Movie: The Little Mermaid
- Screen type: Deluxe
- Screening time: 18.09.2023 - 16:00
- Booker email: admin@test.com
- Adult tickets - £14.99: 1
- Child tickets - £9.99: 0
- Concession: Yes (radio button selected)
- Total Price: (empty input field)

The **booking.html** code includes:

- Logic for displaying screening times based on movie ID and screen type.
- Handling of disabled select options.
- Display of screen types for a given movie ID.

The **Terminal** shows command-line navigation and file status.

Refactoring of front-end and back-end of the booking system

The screenshot shows a code editor with a sidebar containing a project tree. The tree includes:

- CODEBUSTERS
- application > templates > payment.html
- application
- modules
- routes
- static
- css
- imgs
- classifications
- menu
- movie\_banners
- movie\_posters
- payment
- chip.png
- visa.png
- seating\_plans
- js
- booking.js
- swiper.js
- templates

  - about.html
  - account.html
  - booking.html
  - checkout.html
  - classification.html
  - create\_post.html
  - discussion.html
  - home.html
  - layout.html
  - login.html
  - movie\_details.html
  - movies.html
  - payment.html M
  - profile.html
  - register.html
  - screen.html
  - search.html
  - services.html

The main pane displays the content of the payment.html file, which contains Jinja2 template code. The code includes logic for cardholder name, card number, expiration date, and payment methods (chip and visa).

```
{% for error in form.cardholder_name.errors %}  
    <p>{{ error }}</p>  
{% endif %}  
</div>  
  
<!-- Card Number field -->  
<div class="form-group">  
    <label for="{{ form.card_number.id }}">  
        {{ form.card_number.label.text }}</label>  
    >  
    {{ form.card_number(class="form-control") }} {% if  
form.card_number.errors %}  
        <div class="errors">  
            {% for error in form.card_number.errors %}  
                <p>{{ error }}</p>  
            {% endif %}  
        </div>  
    {% endif %}  
</div>  
  
<!-- Expiration date field -->  
<div class="form-group">  
    <label for="{{ form.expire.id }}">  
        {{ form.expire.label.text }}</label>  
    >  
    {{ form.expire(class="form-control") }}  
    <!-- Display errors -->  
    {% if form.expire.errors %}  
        <div class="errors">  
            <!-- Going through each error -->  


At the bottom of the editor, there are tabs for py, bash, and a terminal window showing command-line logs.


```

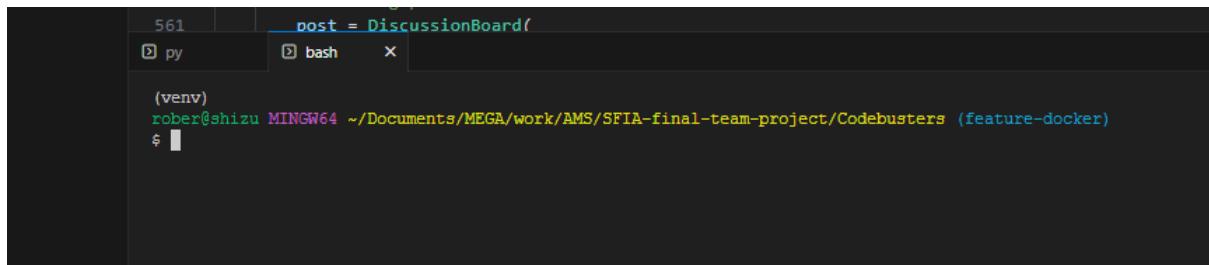
Started refactoring of payment page

## Important refactoring:

→ bookings not tracked/info not stored until payment is successful

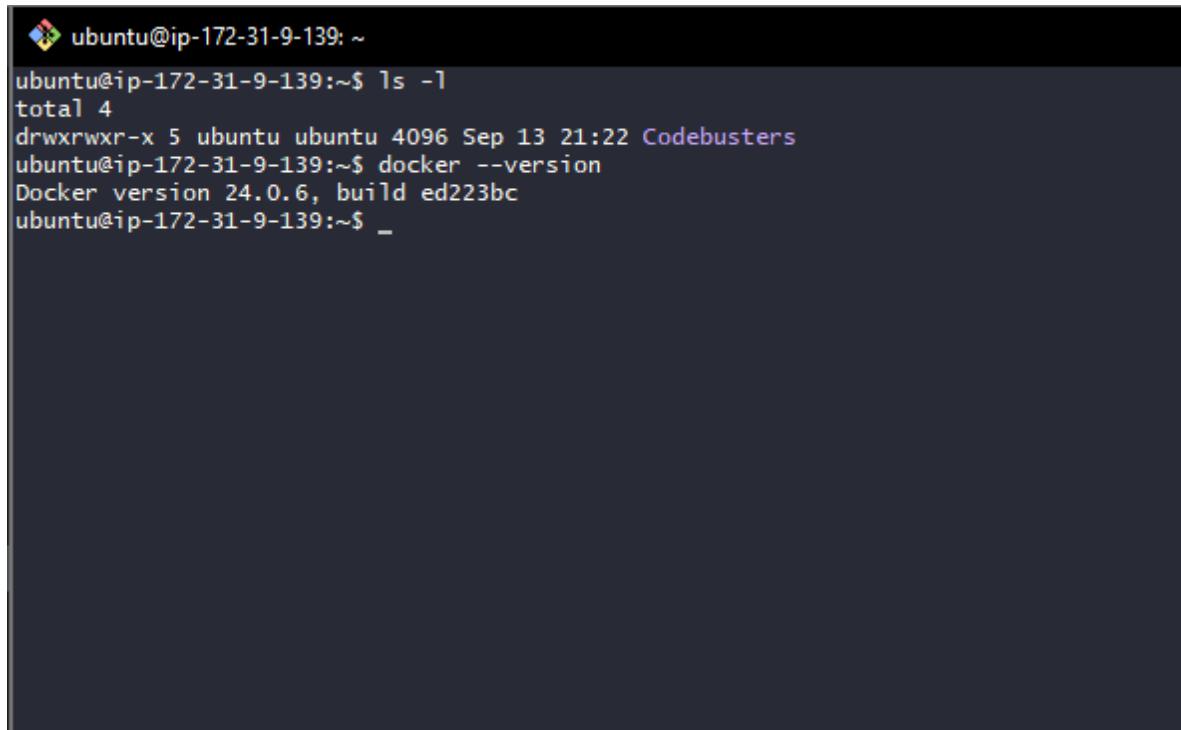
Unfixable problem due to limited time:

- we do not track the capacity of the screens for every screening time, we just have a general capacity field which holds the amount of seats
- a new capacity field in the MovieScreen table is needed in order to track the capacity at each screening time, and then also perform capacity management



A screenshot of a terminal window titled 'post = DiscussionBoard()' containing the following text:  
(venv)  
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/SFIA-final-team-project/Codebusters (feature-docker)  
\$

Time to use Docker to create image/container of the almost finished app → this will be done on the Terraform server instance on AWS (reasoning is pretty obvious, so that we can then push our docker image to a container registry



```
ubuntu@ip-172-31-9-139:~  
ubuntu@ip-172-31-9-139:~$ ls -l  
total 4  
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 13 21:22 Codebusters  
ubuntu@ip-172-31-9-139:~$ docker --version  
Docker version 24.0.6, build ed223bc  
ubuntu@ip-172-31-9-139:~$ _
```

Installed Docker

```
NO containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-9-139:~/Codebusters$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34-Ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.00 sec)

mysql> CREATE DATABASE qa_cinema;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| qa_cinema      |
| sys            |
+-----+
5 rows in set (0.01 sec)

mysql>
```

Testing configuration set up and other things before containerising the application

```
ubuntu@ip-172-31-9-139:~/Codebusters
ubuntu@ip-172-31-9-139:~/Codebusters$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34-Ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| qa_cinema      |
| sys            |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Set up root user (this should be changed in later dev stages)

```

❖ ubuntu@ip-172-31-9-139: ~/Codebusters
(venv) ubuntu@ip-172-31-9-139:~/Codebusters$ ls -l
total 80
-rw-rw-r-- 1 ubuntu ubuntu 348 Sep 13 22:55 Dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 712 Sep 13 21:22 README.md
-rw-rw-r-- 1 ubuntu ubuntu 108 Sep 13 21:22 app.py
drwxrwxr-x 7 ubuntu ubuntu 4096 Sep 14 07:12 application
-rw-rw-r-- 1 ubuntu ubuntu 50217 Sep 13 21:22 create.py
-rw-rw-r-- 1 ubuntu ubuntu 446 Sep 13 21:22 requirements.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 13 21:22 terraform
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 14 07:14 venv
(venv) ubuntu@ip-172-31-9-139:~/Codebusters$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34-Ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE qa_cinema;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

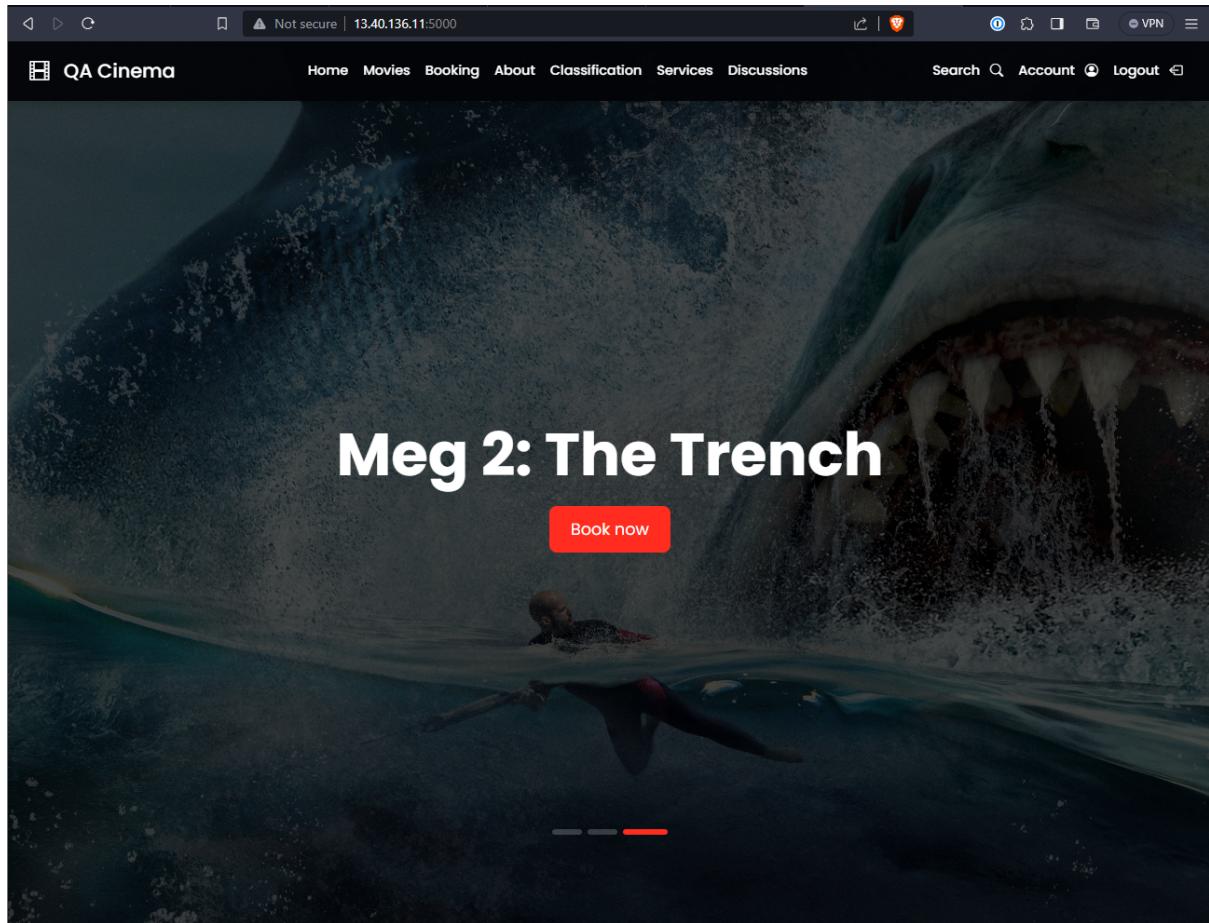
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_qa_cinema |
+-----+
| booking           |
| cast              |
| classification   |
| discussion_board |
| genre             |
| menu_service      |
| movie             |
| movie_cast        |
| movie_genre       |
| movie_screen      |
| payment           |
| screen            |
| ticket            |
| ticket_booking    |
| user              |
+-----+
15 rows in set (0.01 sec)

mysql> SELECT * FROM ticket;
+-----+-----+-----+
| ticket_id | ticket_type | price |
+-----+-----+-----+
|      1 | Adult      | 14.99 |
|      2 | Child      | 9.99  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> _

```

Set everything up to check that things still work on the terraform server EC2 instance (from windows to ubuntu) → creation of db using create.py is all good



Exposed port 5000 for the flask app, and everything works fine locally on the terraform EC2 instance

A screenshot of the AWS Management Console. The top navigation bar shows the AWS logo, 'Services', a search bar, and a keyboard shortcut '[Alt+S]'. Below the navigation, the breadcrumb trail shows 'Amazon ECR &gt; Repositories &gt; Create repository'. The main content area is titled 'Create repository'. Under this, there is a section titled 'General settings'. Within 'General settings', there is a 'Visibility settings' section with two options: 'Private' (selected) and 'Public'. The 'Private' option is described as 'Access is managed by IAM and repository policy permissions.' The 'Public' option is described as 'Publicly visible and accessible for image pulls.' At the bottom of the 'General settings' section, there is a 'Repository name' input field with placeholder text 'Provide a name for your new repository'. The overall interface is dark-themed.

Creating ECR to store our future containers

```

Changes to Outputs:
+ database_endpoint = (known after apply)
+ database_port      = (known after apply)
+ web_public_dns     = (known after apply)
+ web_public_ip       = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_key_pair.EC2_kp: Creating...
aws_vpc.qa_cinema_vpc: Creating...
aws_key_pair.EC2_kp: Creation complete after 0s [id=EC2_kp]
aws_vpc.qa_cinema_vpc: Creation complete after 1s [id=vpc-0380270532a5d7177]
aws_subnet.qa_cinema_private_subnet[1]: Creating...
aws_internet_gateway.qa_cinema_igw: Creating...
aws_subnet.qa_cinema_private_subnet[0]: Creating...
aws_subnet.qa_cinema_public_subnet[0]: Creating...
aws_route_table.qa_cinema_private_rt: Creating...
aws_security_group.qa_cinema_web_sg: Creating...
aws_route_table.qa_cinema_private_rt: Creation complete after 1s [id=rtb-0c3fa4189990325ba]
aws_internet_gateway.qa_cinema_igw: Creation complete after 1s [id=igw-088068557611a3ec8]
aws_route_table.qa_cinema_public_rt: Creating...
aws_subnet.qa_cinema_private_subnet[1]: Creation complete after 1s [id=subnet-0731ae09ad63272f6]
aws_subnet.qa_cinema_private_subnet[0]: Creation complete after 1s [id=subnet-0fdb174964d98966]
aws_subnet.qa_cinema_public_subnet[0]: Creation complete after 1s [id=subnet-0c0fb32d26e9cf252]
aws_db_subnet_group.qa_cinema_db_subnet_group: Creating...
aws_route_table_association.private[0]: Creating...
aws_route_table_association.private[1]: Creating...
aws_route_table_association.private[0]: Creation complete after 0s [id=rtbassoc-07aa9f3e49f72c6d3]
aws_route_table_association.private[1]: Creation complete after 0s [id=rtbassoc-08bdbdedb41b19ef0]
aws_route_table.qa_cinema_public_rt: Creation complete after 0s [id=rtb-0bd3c70c989d057b4]
aws_route_table_association.public[0]: Creating...
aws_route_table_association.public[0]: Creation complete after 0s [id=rtbassoc-0b33859a438b59544]
aws_security_group.qa_cinema_web_sg: Creation complete after 2s [id=sg-048d79ff4cb692d54]
aws_security_group.qa_cinema_db_sg: Creating...
aws_instance.web[0]: Creating...
aws_db_subnet_group.qa_cinema_db_subnet_group: Creation complete after 1s [id=qa_cinema_db_subnet_group]
aws_security_group.qa_cinema_db_sg: Creation complete after 1s [id=sg-0d1bd1af81b36ebb3]
aws_db_instance.qa_cinema_db: Creating...
aws_instance.web[0]: Still creating... [10s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [10s elapsed]
aws_instance.web[0]: Still creating... [20s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [20s elapsed]
aws_instance.web[0]: Still creating... [30s elapsed]
aws_instance.web[0]: Creation complete after 31s [id=i-01e02758f773ac1ac]
aws_eip.qa_cinema_web_eip[0]: Creating...
aws_db_instance.qa_cinema_db: Still creating... [30s elapsed]
aws_eip.qa_cinema_web_eip[0]: Creation complete after 1s [id=eipalloc-0145eaa67cf75f2d1]
aws_db_instance.qa_cinema_db: Still creating... [40s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [50s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m0s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m10s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m20s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m30s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m40s elapsed]
aws_db_instance.qa_cinema_db: Still creating... [1m50s elapsed]

```

Running terraform configuration to create EC2 instances. This is done due to the limited time, else it would have been set up using Jenkins Plugins. We will create our EC2 instances, SSH into them, clone our project repository, and modify the database connection URI to work with our private RDS

## TODO:

- Jenkins:
  - used only to build the application and run the tests
  - due to the limited time, we were not able to actually use any of the plugins we initially wanted to use (i.e., Terraform, Ansible, Docker)
    - Terraform configuration has been created and used
      - it uses a parameter to create a N number of EC2 instances (where our application is going to be placed)
      - it creates a private RDS which is accessible only from the newly created EC2 instances

→ Ansible

→ not used at all, it should be used to correctly set up our EC2 instances (i.e., update, install docker, and everything else that is needed for the app to work on the EC2 instances)

→ Docker

→ used in the following way:

1. we SSH into the EC2 instances created using the Terraform configuration
2. we set everything up manually for now (no Ansible used)
  - a. clone project repo
  - b. install things needed and dependencies
  - c. use Docker to create image/container
  - d. Run container

→ everything is still very fragmented, and the automation part lacks a lot

→ future iterations: work on the missing things and complete the CI/CD side of the project

→ different EC2 instances are there, however, there's no load balancer, yet.

A screenshot of a terminal window on an Ubuntu system. The terminal shows the output of several commands:

```
ubuntu@ip-10-0-1-83:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following packages will be upgraded:
  curl libcurl4
2 upgraded, 0 newly installed, 0 to remove and 117 not upgraded.
Need to get 484 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.13 [194 kB]
Get:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4 amd64 7.81.0-1ubuntu1.13 [290 kB]
Fetched 484 kB in 0s (19.5 MB/s)
(Reading database ... 64295 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.13_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.13) over (7.81.0-1ubuntu1.10) ...
Preparing to unpack .../libcurl4_7.81.0-1ubuntu1.13_amd64.deb ...
Unpacking libcurl4:amd64 (7.81.0-1ubuntu1.13) over (7.81.0-1ubuntu1.10) ...
Setting up libcurl4:amd64 (7.81.0-1ubuntu1.13) ...
Setting up curl (7.81.0-1ubuntu1.13) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-1-83:~$ curl https://get.docker.com | sudo bash
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total   Spent    Left Speed
100  21927  100  21927    0      0  212k      0 --:--:-- --:--:-- 214k
# Executing docker install script, commit: e5543d473431b782227f8908005543bb4389b8de
+ sh -c 'apt-get update -qq >/dev/null'
+ sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null'
```

Installed Docker on one of the EC2 instances created using the Terraform configuration

After setting everything up (for now, only venv, dependencies, etc), we want to test if everything works as expected with the cloud database before creating the docker image/container

```
DB_PASSWORD="BxV7N22dEqB337Vq2B.n"
DB_HOST="terraform-20230914113523960800000001.cxwrrn7oq8ws.eu-west-2.rds.amazonaws.com"
DB_NAME="qa_cinema"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

### Expected problem:

- when trying to run our create.py using the cloud based db, there is no qa\_cinema database yet.

```
sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (1049, "Unknown database 'qa_cinema'")
(Background on this error at: https://sqlalche.me/e/20/e3q8)
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$
```

### → fix:

- we log into the RDS instance, and manually create the database for now
- this needs to be done only ONCE

```
ubuntu@ip-10-0-1-83: ~/Codebusters
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ mysql -h terraform-2023091411352396080000001.cxwwrn7oq8ws.eu-west-2.rds.amazonaws.com -P 3
306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

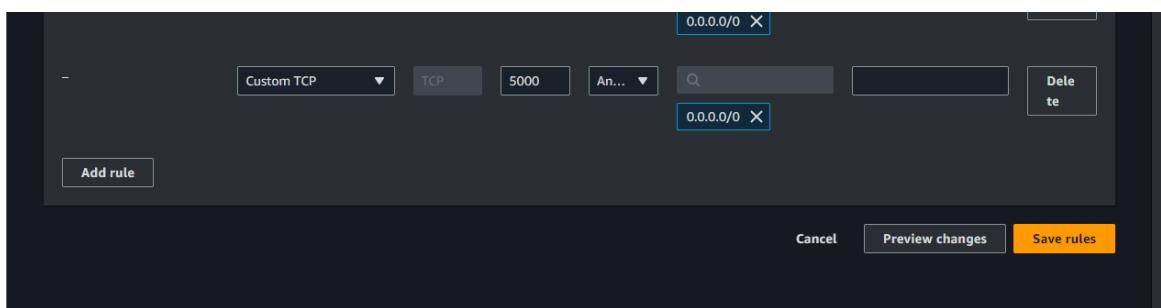
mysql> CREATE DATABASE qa_cinema;
Query OK, 1 row affected (0.01 sec)

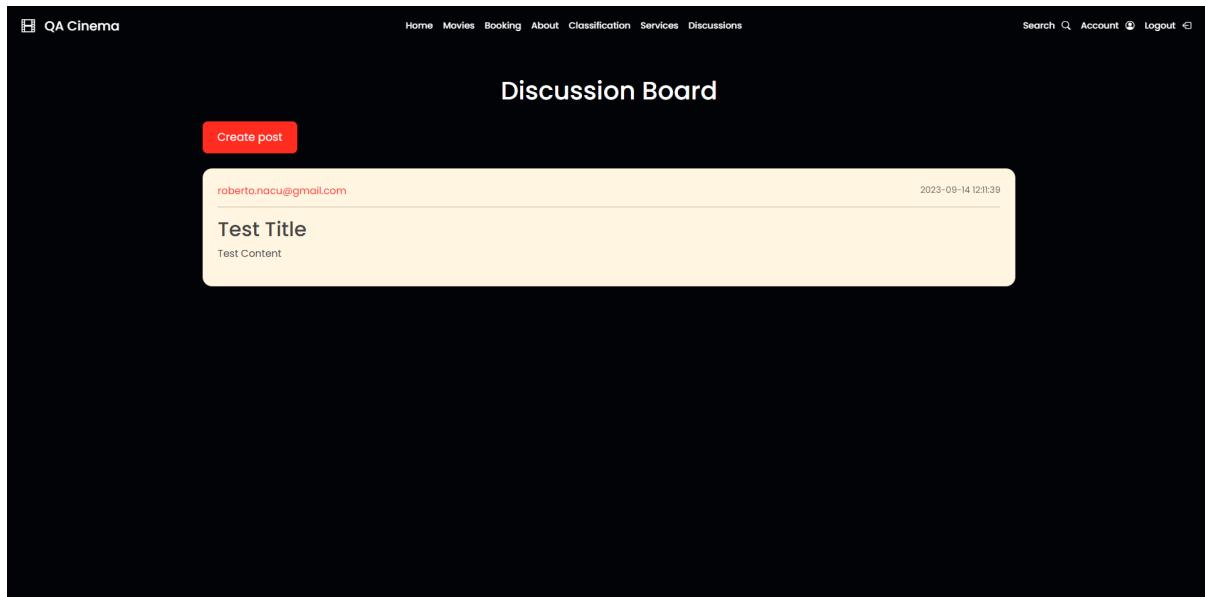
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| qa_cinema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

## PROBLEM: Terraform config file needs to be fixed

- we allow access on port 80 for the EC2 instances that are being created, not good?
- manually edited the web app instances security group for now to allow access on the flask port





Cloud database working. The database is fully private and accessible only from the EC2 instances that are created using the Terraform configuration files

```
ubuntu@ip-10-0-1-83: ~/Codebusters
# Python image
FROM python:3.11-alpine

# set working dir
WORKDIR /app

# copy current dir content into the container at /app
COPY . /app

# install required dependencies
RUN pip install -r requirements.txt

# make port 80 available to the world outside this container
EXPOSE 5000

# we set some environment variables, only the non-sensitive ones
# the sensitive ones will be set at container run
ENV DB_TYPE=mysql+pymysql://
ENV DB_USER=admin
# TODO: in future iterations, this will not be set like this, but passed to the container using the outputs from the terraform config
ENV DB_HOST=terraform-20230914113523960800000001.cxwrrn7qq8ws.eu-west-2.rds.amazonaws.com
ENV DB_NAME=qa_cinema

# run app.py when the container launches
ENTRYPOINT python app.py
~
```

A screenshot of a terminal window on an Ubuntu system. The window title is "ubuntu@ip-10-0-1-83: ~/Codebusters". The terminal displays a Dockerfile. The Dockerfile starts with a Python image, sets the working directory to /app, copies the current directory content into the container, installs required dependencies using pip, exposes port 5000, sets environment variables (DB\_TYPE, DB\_USER), and specifies the entrypoint as "python app.py". There are approximately 20 tilde (~) characters at the end of the file.

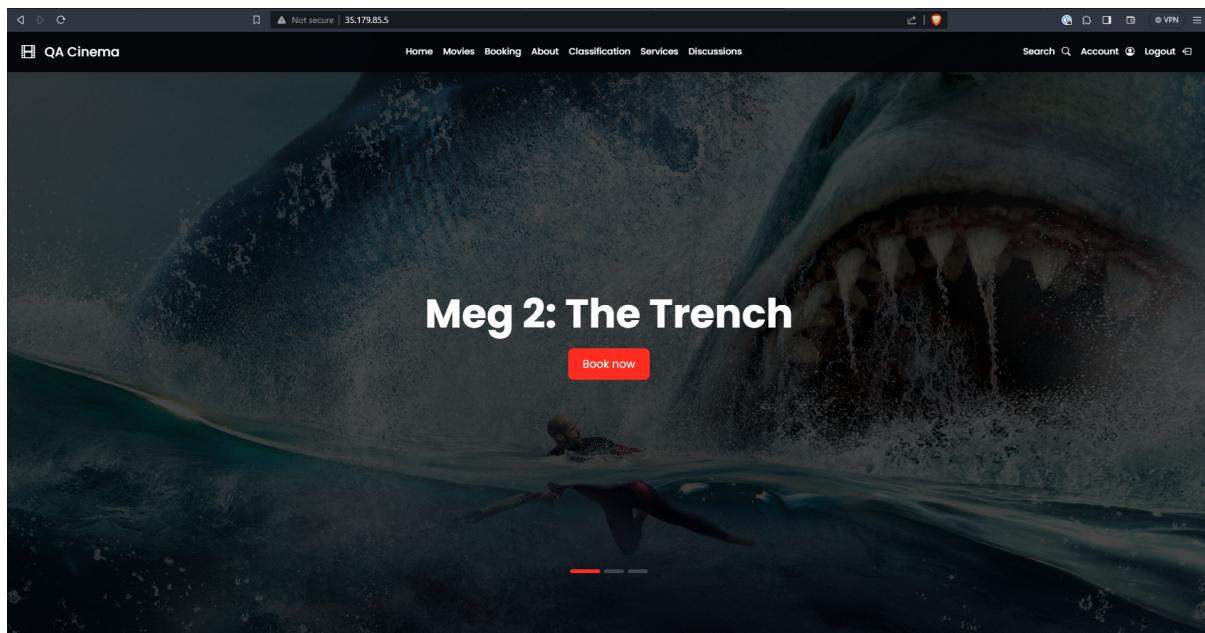
Now we containerise the application. In the Dockerfile we set only the non-sensitive environment variables. The remaining ones (db password and secret key) are set at container run time

```
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ ls -l
total 80
-rw-rw-r-- 1 ubuntu ubuntu 348 Sep 14 11:49 Dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 712 Sep 14 11:49 README.md
-rw-rw-r-- 1 ubuntu ubuntu 108 Sep 14 11:49 app.py
drwxrwxr-x 7 ubuntu ubuntu 4096 Sep 14 11:56 application
-rw-rw-r-- 1 ubuntu ubuntu 50217 Sep 14 11:49 create.py
-rw-rw-r-- 1 ubuntu ubuntu 446 Sep 14 11:49 requirements.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 14 11:49 terraform
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 14 11:53 venv
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ vim Dockerfile
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ vim Dockerfile
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ vim Dockerfile
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ docker build -t qa-cinema-flask:v1 .
[+] Building 16.5s (9/9) FINISHED                                            docker:default
  => [internal] load build definition from Dockerfile                      0.1s
  => transferring dockerfile: 7968                                         0.0s
  => [internal] load .dockerignore                                       0.1s
  => transferring context: 110B                                         0.0s
  => [internal] load metadata for docker.io/library/python:3.11-alpine   1.4s
  => [1/4] FROM docker.io/library/python:3.11-alpine@sha256:5d769f990397afbb2aca24b0655e404c0f2806d268f454b052e81e39d87abf42 2.5s
  =>  resolve docker.io/library/python:3.11-alpine@sha256:5d769f990397afbb2aca24b0655e404c0f2806d268f454b052e81e39d87abf42 0.0s
  =>=> sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 3.40MB / 3.40MB 0.3s
  =>=> sha256:66e1d5e70e420aa86a23bd8b4ebf2a6eb60b4aff9ee8a6ca52e27f51f57b1be 622.31kB / 622.31kB 0.3s
  =>=> sha256:0448660c92fc1b6557cf48081d99028eeaba7809a1c2f23018e2331f3cf4b29 12.46MB / 12.46MB 0.6s
  =>=> sha256:5d769f90397afbb2aca24b0655e404c0f2806d268f454b052e81e39d87abf42 1.65kB / 1.65kB 0.0s
  =>=> sha256:e5d92c422d6e527cb946ae6abb1886c511a5e163d3543865f5a5b9b61c01584 1.37kB / 1.37kB 0.0s
  =>=> sha256:b9b301ab01c2af0b5f2069b97e1d89885433dc55b6623ad6aa4a43b2e017fb3 6.26kB / 6.26kB 0.0s
  =>=> extracting sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 0.5s
  =>=> sha256:3ce23f846e315e35618c4604547bbe6aa2b48a0601c335c76b8fe02729bb5c49 241B / 241B 0.5s
  =>=> sha256:efebc2e683d297ae71acae9230d17af8b95684d94af9b7f601a8e1e9b103bda3 3.11MB / 3.11MB 0.6s
  =>=> extracting sha256:0448660c92fc1b6557cf48081d99028eeaba7809a1c2f23018e2331f3cf4b29 0.7s
  =>=> extracting sha256:3ce23f846e315e35618c4604547bbe6aa2b48a0601c335c76b8fe02729bb5c49 0.0s
  =>=> extracting sha256:efebc2e683d297ae71acae9230d17af8b95684d94af9b7f601a8e1e9b103bda3 0.3s
  => [internal] load build context                                         0.8s
  => transferring context: 27.15MB                                         0.2s
  => [2/4] WORKDIR /app                                                 0.0s
  => [3/4] COPY . /app                                                 0.2s
  => [4/4] RUN pip install -r requirements.txt                           10.9s
  => exporting to image                                                 1.0s
  =>=> exporting layers                                              1.0s
  =>=> writing image sha256:3df7dc91ccbf598cb4231dd74cd4acd92fe5880a314ecd0a075359726111b1c3 0.0s
  =>=> naming to docker.io/library/qa-cinema-flask:v1                 0.0s
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ docker images
REPOSITORY      TAG           IMAGE ID      CREATED       SIZE
qa-cinema-flask   v1          3df7dc91ccbf  8 seconds ago  142MB
(venv) ubuntu@ip-10-0-1-83:~/Codebusters$ _
```

We build the Docker image

→ we set the missing environment variables using `export`

```
docker run -d -p 80:5000 -e DB_PASSWORD=${DB_PASSWORD} -e SECRET_KEY=${SECRET_KEY} --name qa_cinema qa-cinema-flask:v2
```



Site working!

```
# Python image
FROM python:3.11-alpine

# set working dir
WORKDIR /app

# copy current dir content into the container at /app
COPY . /app

# install required dependencies
RUN pip install -r requirements.txt

# make port 80 available to the world outside this container
EXPOSE 5000

# we set some environment variables, only the non-sensitive ones
# the sensitive ones will be set at container run |DB_PASSWORD, DB_HOST, and SECRET_KEY|
ENV DB_TYPE=mysql+pymysql://
ENV DB_USER=admin
ENV DB_NAME=qa_cinema

# run app.py when the container launches
ENTRYPOINT python app.py
```

Updated Dockerfile

```

ubuntu@ip-10-0-1-83:~/Codebusters$ vim Dockerfile
ubuntu@ip-10-0-1-83:~/Codebusters$ docker build -t qa-cinema-flask:v1 .
[+] Building 1.0s (9/9) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 61B
--> [internal] load .dockignore
--> => transferring context: 110B
--> [internal] load metadata for docker.io/library/python:3.11-alpine
--> [1/4] FROM docker.io/library/python:3.11-alpine@sha256:5d769f990397afbb2aca24b0655e404c0f2806d268f454b052e81e39d87abf42
--> => 5.66kB
--> => transferring context: 5.66kB
--> CACHED [2/4] WORKDIR /app
--> CACHED [3/4] COPY ./app
--> CACHED [4/4] RUN pip install -r requirements.txt
--> => exporting to image
--> => exporting layers
--> => writing image sha256:5292f94666e400454fdef9a5743551ad282063b364Fc555108e02ace0f063d7
--> => naming to docker.io/library/qa-cinema-flask:v1
ubuntu@ip-10-0-1-83:~/Codebusters$ docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
qa-cinema-flask   v1            5292f94666e4        2 hours ago        142MB
ubuntu@ip-10-0-1-83:~/Codebusters$ export DB_HOST=$(terraform output -raw database_endpoint).cxwrrn7oq8ws.eu-west-2.rds.amazonaws.com"

```

Rebuilding image

This now allows us to run the container using the following command:

- we set the missing environment variables using `export`
1. `DB_PASSWORD` → db password associated with admin user
  2. `SECRET_KEY`
  3. `DB_HOST` → cloud database endpoint
    - a. `export DB_HOST=$(terraform output -raw database_endpoint)`

```

docker run -d -p 80:5000 -e DB_PASSWORD=${DB_PASSWORD} -e SECRET_KEY=${SECRET_KEY} -e
DB_HOST=${DB_HOST} --name qa_cinema qa-cinema-flask:v2

```

```
ubuntu@ip-10-0-1-83:~/Codebusters$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: 1391819
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-10-0-1-83:~/Codebusters$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
qa-cinema-flask    v1      5292f94666e4  3 hours ago   142MB
ubuntu@ip-10-0-1-83:~/Codebusters$ docker tag qa-cinema-flask:v1 1391819/qa-cinema-flask:v1
ubuntu@ip-10-0-1-83:~/Codebusters$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
1391819/qa-cinema-flask  v1      5292f94666e4  3 hours ago   142MB
qa-cinema-flask        v1      5292f94666e4  3 hours ago   142MB
ubuntu@ip-10-0-1-83:~/Codebusters$ docker push 1391819/qa-cinema-flask:v1
The push refers to repository [docker.io/1391819/qa-cinema-flask]
a8195d611993: Pushed
cf2abb8e4a98: Pushed
ba624229fc95: Pushed
08928985481f: Mounted from library/python
7acf52b2a13c: Mounted from library/python
ce0f4c80e9b7: Mounted from library/python
9ad60cb84bfbe: Mounted from library/python
4693057ce236: Mounted from library/python
v1: digest: sha256:01ee5c49a2b3394de6e4f11a3a999e02af86ead1fc5725bd452e30ff8ed64e2d size: 1998
ubuntu@ip-10-0-1-83:~/Codebusters$ docker logout
Removing login credentials for https://index.docker.io/v1/
ubuntu@ip-10-0-1-83:~/Codebusters$ _
```

Pushed our repository to Dockerhub

1391819 Repositories qa-cinema-flask General Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Collaborators Webhooks Settings

[Add a short description for this repository](#) [Update](#)  
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

1391819 / qa-cinema-flask

Description  
This repository does not have a description [Edit](#)

Last pushed: a minute ago

Docker commands  
To push a new tag to this repository:  
`docker push 1391819/qa-cinema-flask:tagname`

[Public View](#)

Tags  
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	--	a minute ago

[See all](#) [Go to Advanced Image Management](#)

Automated Builds  
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.  
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

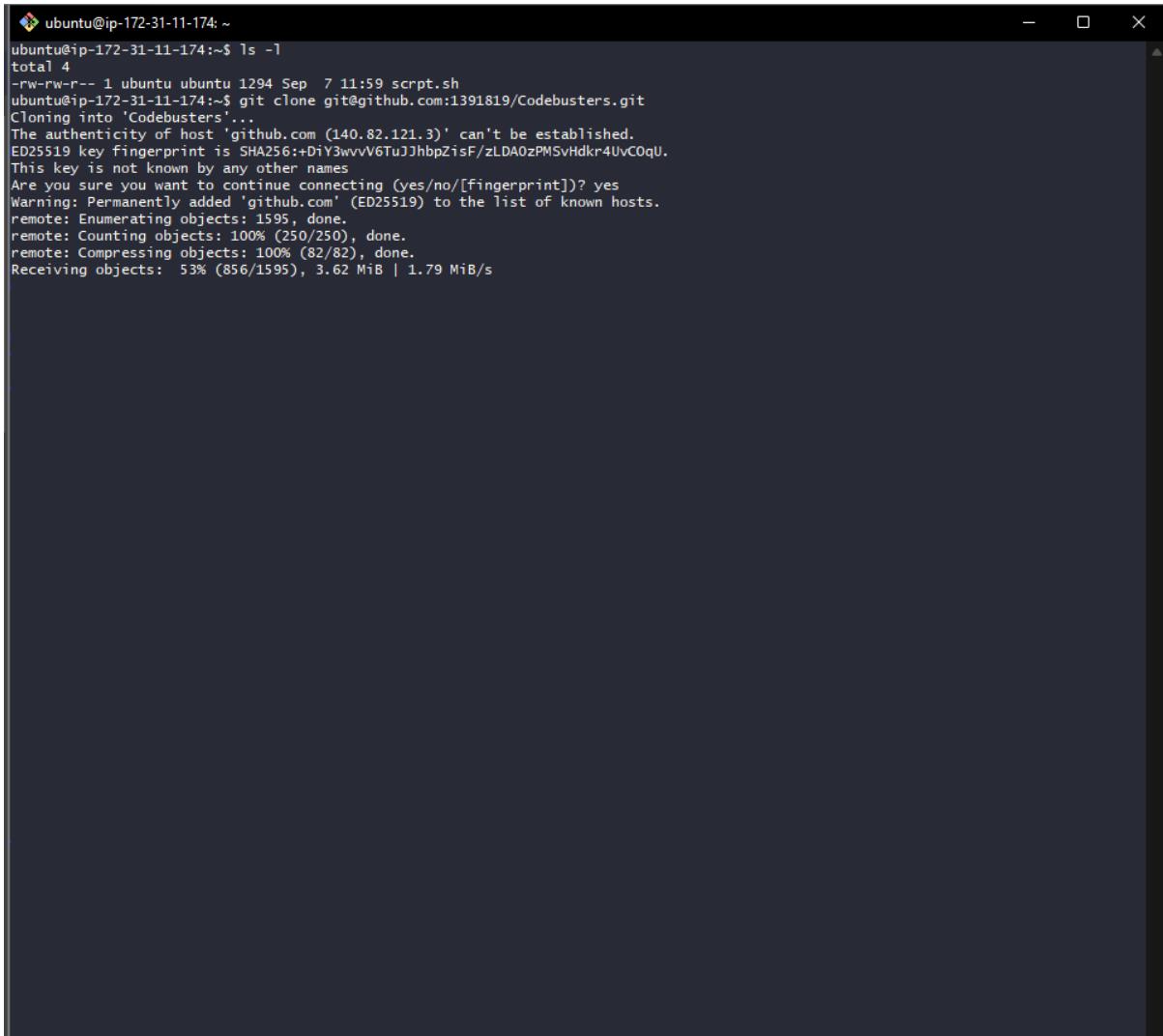
Repository overview [Edit](#)  
An overview describes what your image does and how to run it. It displays in [the public view of your repository](#).

[Add overview](#)

```
ubuntu@ip-172-31-11-174:~$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 1294 Sep  7 11:59 scrpt.sh
ubuntu@ip-172-31-11-174:~$ vim scrpt.sh
ubuntu@ip-172-31-11-174:~$ sudo visudo
visudo: /etc/sudoers.tmp unchanged
ubuntu@ip-172-31-11-174:~$ sudo visudo
visudo: /etc/sudoers.tmp unchanged
ubuntu@ip-172-31-11-174:~$ sudo visudo
ubuntu@ip-172-31-11-174:~$
```

```
ubuntu@ip-172-31-11-174:~$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 1294 Sep  7 11:59 scrpt.sh
ubuntu@ip-172-31-11-174:~$ vim scrpt.sh
ubuntu@ip-172-31-11-174:~$ sudo visudo
visudo: /etc/sudoers.tmp unchanged
ubuntu@ip-172-31-11-174:~$ sudo visudo
visudo: /etc/sudoers.tmp unchanged
ubuntu@ip-172-31-11-174:~$ sudo visudo
ubuntu@ip-172-31-11-174:~$
```

Setting sudoers for jenkins pipeline

A screenshot of a terminal window titled "ubuntu@ip-172-31-11-174: ~". The window shows the command "git clone git@github.com:1391819/Codebusters.git" being run. The output includes file listing, GitHub key fingerprint verification, and progress of cloning objects from the remote repository.

```
ubuntu@ip-172-31-11-174:~$ ls -l
total 4
-rw-r--r-- 1 ubuntu ubuntu 1294 Sep  7 11:59 scrpt.sh
ubuntu@ip-172-31-11-174:~$ git clone git@github.com:1391819/Codebusters.git
Cloning into 'Codebusters'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vvvV6TuJhbpZisF/zLDA0zPMSvHdkr4UvCoqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 1595, done.
remote: Counting objects: 100% (250/250), done.
remote: Compressing objects: 100% (82/82), done.
Receiving objects: 53% (856/1595), 3.62 MiB | 1.79 MiB/s
```

Cloning finished application on jenkins server so we can work on Jenkinsfile

The screenshot shows the Jenkins Global credentials (unrestricted) page. The page header includes the Jenkins logo, a search bar, and various status indicators. The breadcrumb navigation shows: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The main title is "Global credentials (unrestricted)". A blue button at the top right says "+ Add Credentials". Below the title, a sub-instruction reads: "Credentials that should be available irrespective of domain specification to requirements matching." A table lists six credentials:

ID	Name	Kind	Description
DOCKER_LOGIN	1391819/*****	Username with password	
SECRET_KEY	SECRET_KEY	Secret text	
DB_PASSWORD	DB_PASSWORD	Secret text	
DB_HOST	DB_HOST	Secret text	
DB_USER	DB_USER	Secret text	
DB_NAME	DB_NAME	Secret text	

Below the table, there is a note about icon sizes: "Icon: S M L".

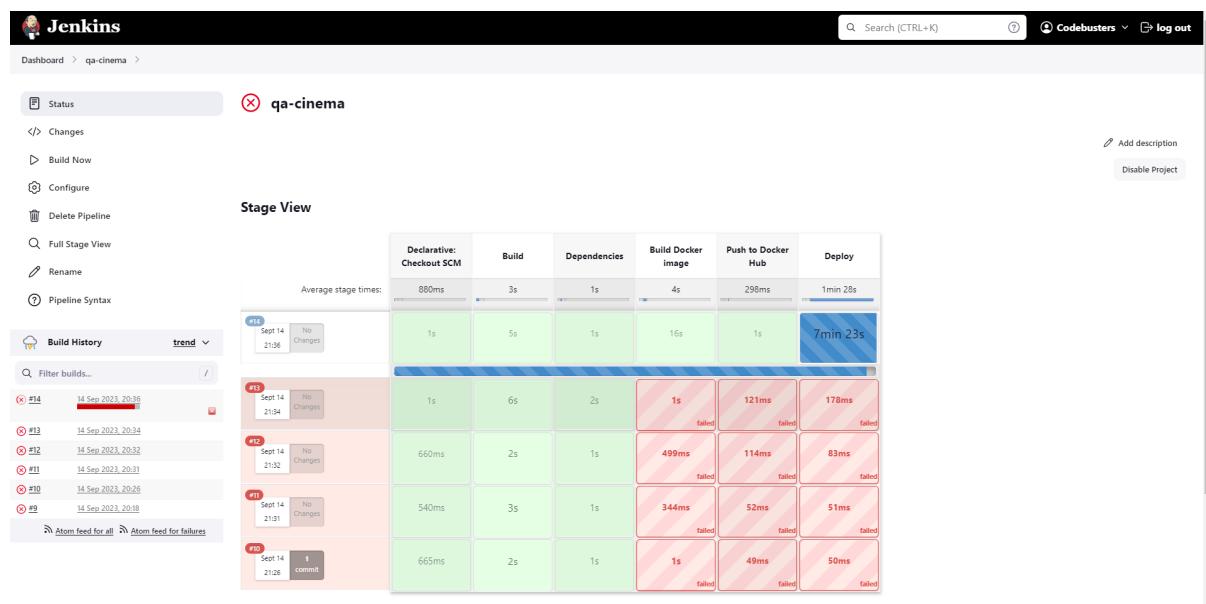
Setting up environment variables in Jenkins so that we can run the Pipeline job containing our application

```

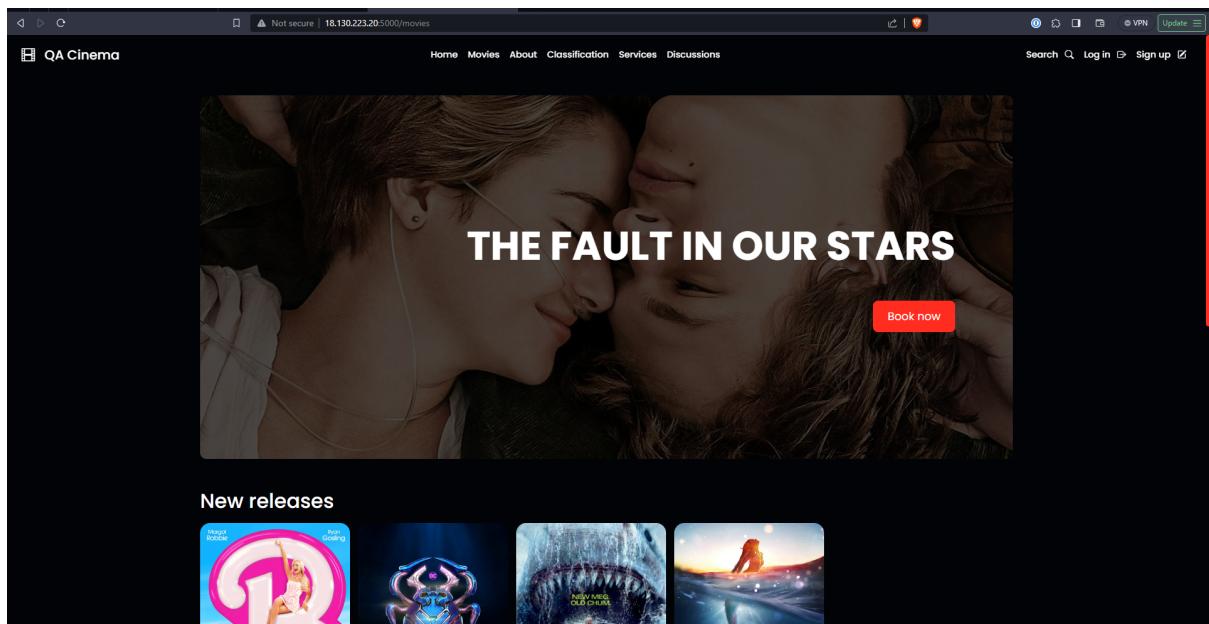
ubuntu@ip-172-31-11-174: ~
ubuntu@ip-172-31-11-174:~$ ls -l
total 8
drwxrwxr-x 6 ubuntu ubuntu 4096 Sep 14 20:30 Codebusters
-rw-rw-r-- 1 ubuntu ubuntu 1294 Sep 7 11:59 scrpt.sh
ubuntu@ip-172-31-11-174:~$ groups jenkins
jenkins : jenkins
ubuntu@ip-172-31-11-174:~$ sudo usermod -aG docker jenkins
ubuntu@ip-172-31-11-174:~$ sudo service jenkins restart
ubuntu@ip-172-31-11-174:~$ _

```

Adding jenkins to the docker group so that we can do docker



Ended up creating a Jenkins pipeline which successfully builds our application. This uses the credentials set in the credentials manager + a local mysql server (set on the jenkins AWS instance, not the cloud database one). This was simply done to showcase how Jenkins can be used to build and continuously run our application



Enabling access on port 5000 for the Jenkins instance so that we can test our application.

Everything working

Instead of building the application like that, we modify the Jenkinsfile to simply create and push the docker image of our application to docker hub. For now, the retrieving of this image will be done manually on the EC2 instances created using the Terraform configuration. However, in future iterations, we will use Ansible to fully complete our CI/CD pipeline

ID	Name	Kind	Description
DOCKER_LOGIN	1391819/*****	Username with password	
SECRET_KEY	SECRET_KEY	Secret text	
DB_PASSWORD	DB_PASSWORD	Secret text	
DB_HOST	DB_HOST	Secret text	

Icon: S M L

Updating the credentials (all the non-sensitive ones are already set in the Dockerfile)

**qa-cinema**

**Stage View**

Declarative: Checkout SCM	Build Docker image	Push to Docker Hub
3s	13s	11s

Average stage times:  
(Average full run time: ~30s)

**Build History**

- #15 Sept 14 22:12 1 commit
- #14
- #13
- #12
- #11
- #10
- #9

Filter builds... /

**Permalinks**

- Last build (#14), 35 min ago
- Last failed build (#13), 37 min ago
- Last unsuccessful build (#13), 37 min ago
- Last completed build (#13), 37 min ago

Atom feed for all Atom feed for failures

Once the Jenkinsfile has been modified accordingly, the Docker image is created and pushed to Docker Hub

1391819

Search by repository name

1391819 / qa-cinema  
Contains: Image | Last pushed: a few seconds ago

```
ubuntu@ip-10-0-1-83: ~
ubuntu@ip-172-31-9-139:~/Codebusters/terraform$ terraform output
database_endpoint = "terraform-2023091411352396080000001.cxwwrn7oq8ws.eu-west-2.rds.amazonaws.com"
database_port = 3306
web_public_dns = "ec2-35-179-85-5.eu-west-2.compute.amazonaws.com"
web_public_ip = "35.179.85.5"
ubuntu@ip-172-31-9-139:~/Codebusters/terraform$ ssh -i ~/.ssh/EC2_kp ubuntu@35.179.85.5
ssh: Could not resolve hostname ubuntu35.179.85.5: Name or service not known
ubuntu@ip-172-31-9-139:~/Codebusters/terraform$ ssh -i ~/.ssh/EC2_kp ubuntu@35.179.85.5
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Thu Sep 14 21:16:32 UTC 2023

 System load: 0.0          Processes:           105
 Usage of /:   38.9% of 7.57GB  Users logged in:      0
 Memory usage: 42%          IPv4 address for docker0: 172.17.0.1
 Swap usage:   0%            IPv4 address for eth0:   10.0.1.83

 * Ubuntu Pro delivers the most comprehensive open source security and
 compliance features.

 https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

111 updates can be applied immediately.
59 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Thu Sep 14 18:41:55 2023 from 13.40.136.11
ubuntu@ip-10-0-1-83:~$ -
```

We SSH into one of the instances created using the Terraform configuration

```

ubuntu@ip-10-0-1-83:~/Codebusters$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu@ip-10-0-1-83:~/Codebusters$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-10-0-1-83:~/Codebusters$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-10-0-1-83:~/Codebusters$ docker pull 1391819/qa-cinema
Using default tag: latest
latest: Pulling from 1391819/qa-cinema
7264a8db6415: Already exists
66e1d5e70e42: Already exists
0448660c92fc: Already exists
3ce23f846e31: Already exists
efebc2e683d2: Already exists
8547b932f922: Pull complete
6407b6d48248: Pull complete
a48e67435660: Pull complete
Digest: sha256:7a6caffd9697b26b046cc4a07a00211ac10758b1d46e9041a070b804a7f2328
Status: Downloaded newer image for 1391819/qa-cinema:latest
docker.io/1391819/qa-cinema:latest
ubuntu@ip-10-0-1-83:~/Codebusters$
```

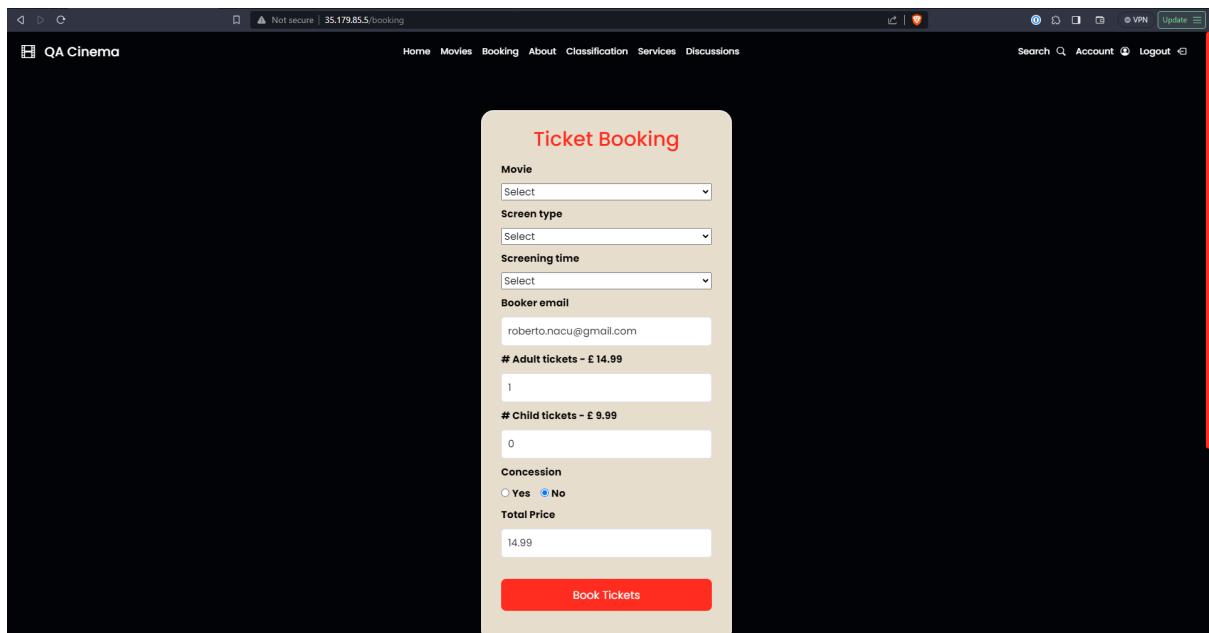
Pulling image created using Jenkins pipeline

```
docker run -d -p 80:5000 -e DB_PASSWORD=${DB_PASSWORD} -e SECRET_KEY=${SECRET_KEY} -e DB_HOST=${DB_HOST} --name qa_cinema qa-cinema
```

```

ubuntu@ip-10-0-1-83:~/Codebusters$ docker run -d -p 80:5000 -e DB_PASSWORD=${DB_PASSWORD} -e SECRET_KEY=${SECRET_KEY} -e DB_HOST=${DB_HOST} --name qa_cinema qa-cinema
Unable to find image 'qa-cinema:latest' locally
docker: Error response from daemon: pull access denied for qa-cinema, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
ubuntu@ip-10-0-1-83:~/Codebusters$ docker run -d -p 80:5000 -e DB_PASSWORD=${DB_PASSWORD} -e SECRET_KEY=${SECRET_KEY} -e DB_HOST=${DB_HOST} --name qa_cinema 1391819/qa-cinema
b7b0cba5ade1dbc20e5966d3795d7cbc544ec52065b5bf85b25865c3984b625
ubuntu@ip-10-0-1-83:~/Codebusters$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b7b0cba5ade1 1391819/qa-cinema "/bin/sh -c 'python ..." 3 seconds ago Up 3 seconds 0.0.0.0:80->5000/tcp, :::80->5000/tcp qa_cinema
ubuntu@ip-10-0-1-83:~/Codebusters$ _
```

Run a container whilst passing all the needed env variables



Site is working!