Social Art Online

From: DevS.SAO

Leader: Aditya L.

December 6, 2022

Team Member Contributions

Aditya - Designed backend on flask in python. Integrated front end with back end.

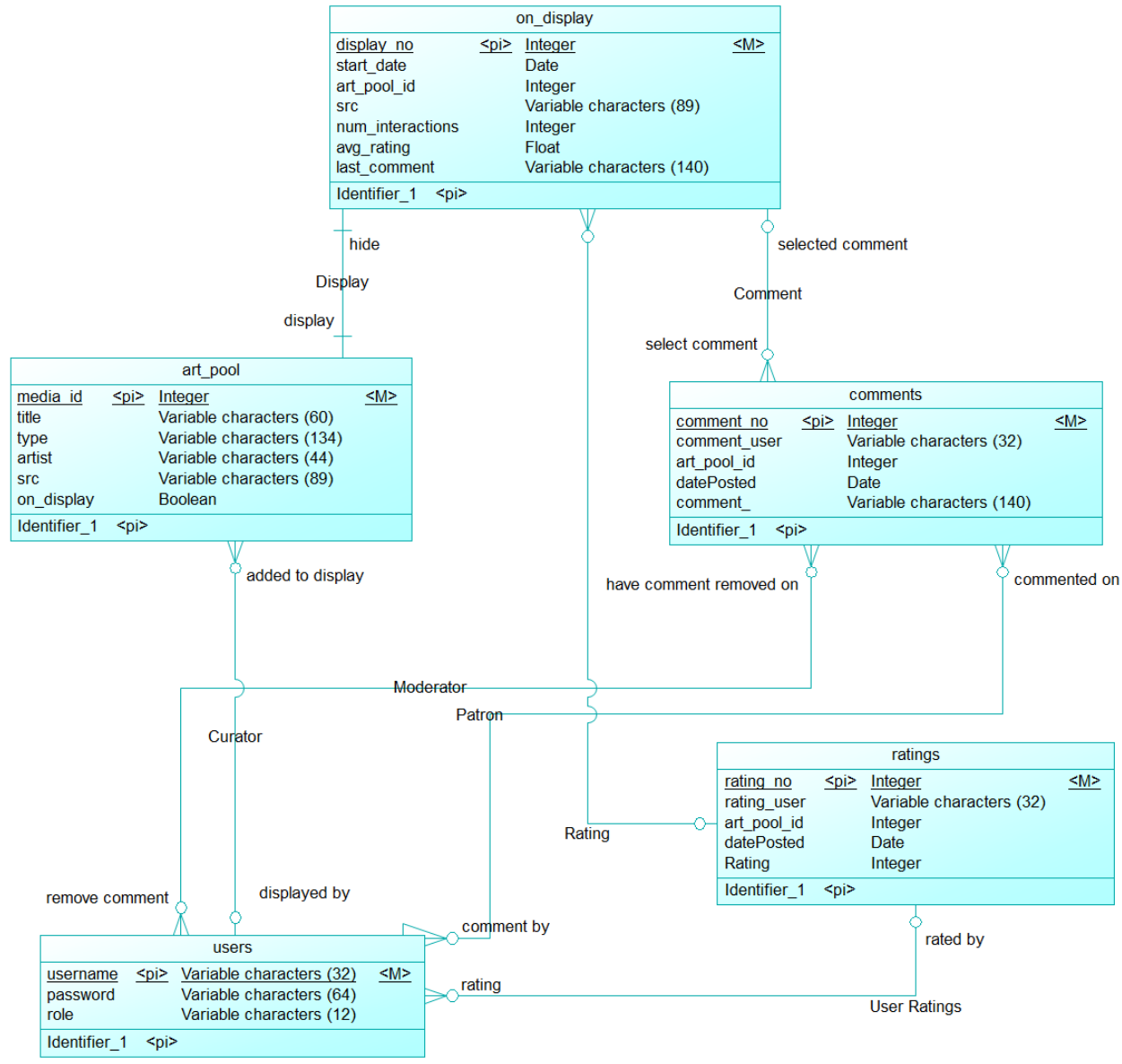
Integrated app with MySQL tables. Implemented a RESTful API for tables.(50%)

Darren - Designed GUI for application. Designed MySQL tables. Designed MySQL views.

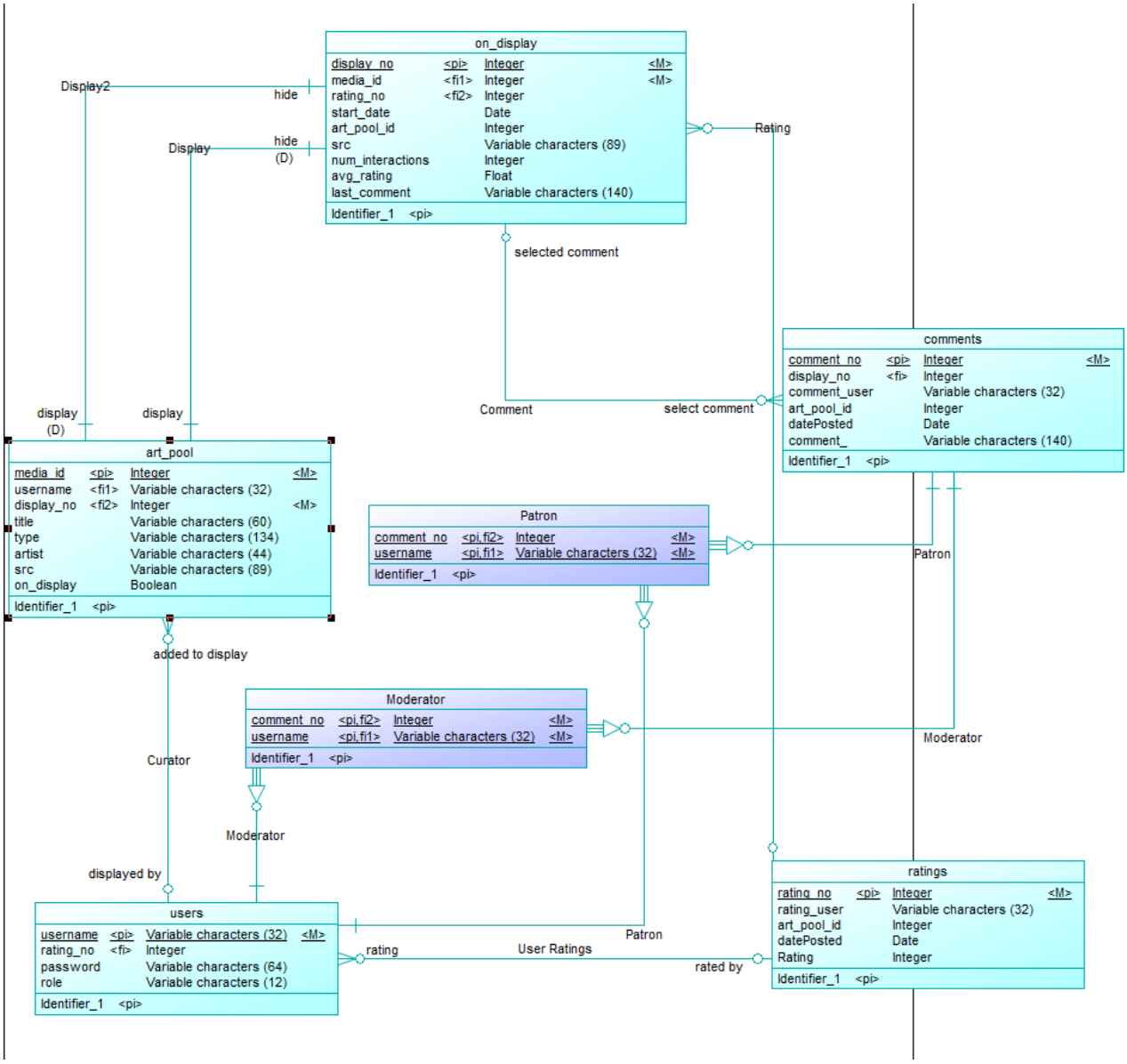
Compiled Data Models for report. (50%)

Data Models

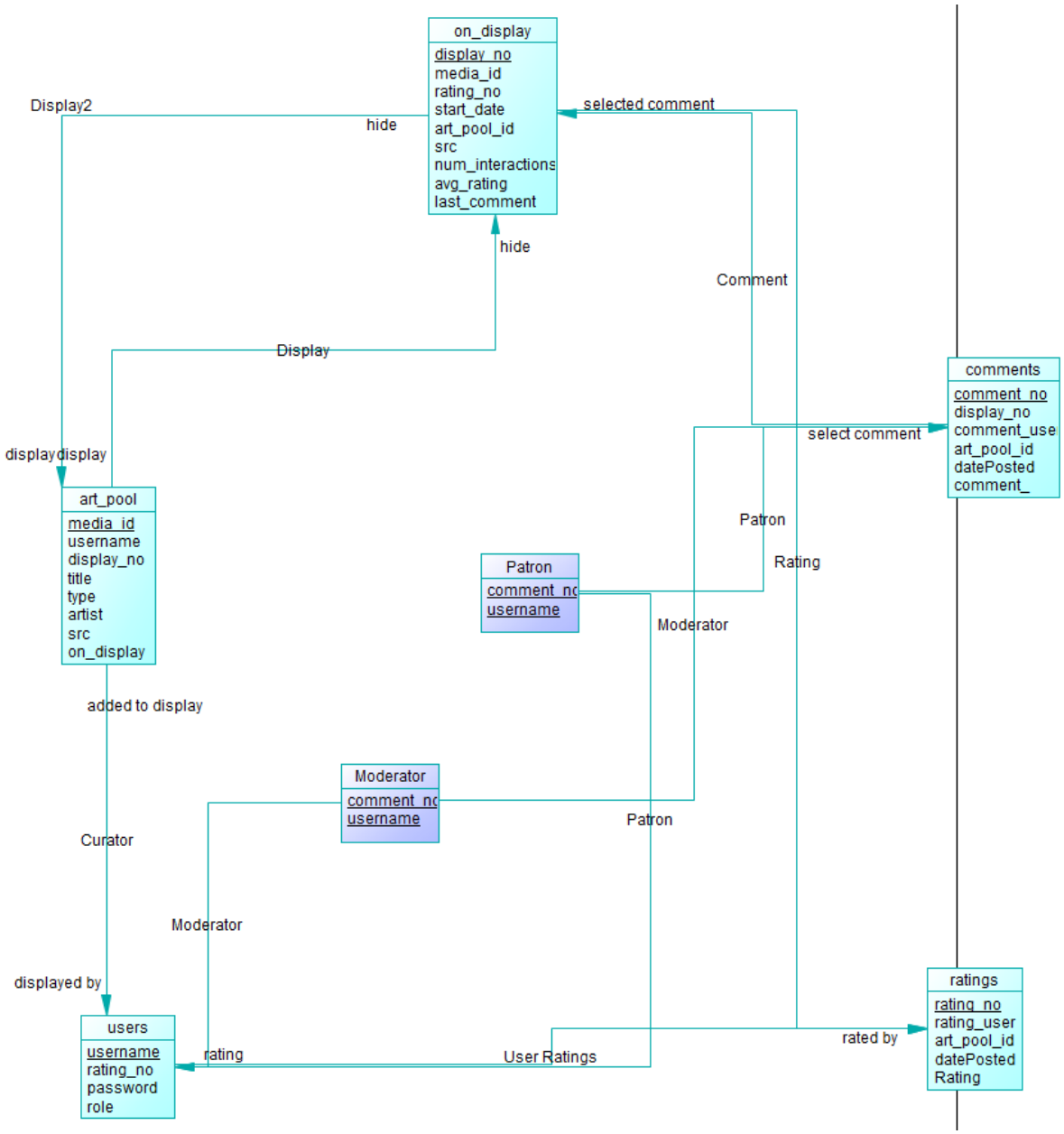
Conceptual



Logical


on_display
display_no <pi> Integer <M>
media_id <fi1> Integer <M>
rating_no <fi2> Integer
start_date Date
art_pool_id Integer
src Variable characters (89)
num_interactions Integer
avg_rating Float
last_comment Variable characters (140)
Identifier_1 <pi>
Display2
hide
Display
hide (D)
Rating
selected comment
comments
comment_no <pi> Integer <M>
display_no <fi> Integer
comment_user Variable characters (32)
art_pool_id Integer
datePosted Date
comment_ Variable characters (140)
Identifier_1 <pi>
Comment
select comment
display (D)
display
art_pool
media_id <pi> Integer <M>
username <fi1> Variable characters (32)
display_no <fi2> Integer <M>
title Variable characters (60)
type Variable characters (134)
artist Variable characters (44)
src Variable characters (89)
on_display Boolean
Identifier_1 <pi>
Patron
comment_no <pi,fi2> Integer <M>
username <pi,fi1> Variable characters (32) <M>
Identifier_1 <pi>
Patron
added to display
Moderator
comment_no <pi,fi2> Integer <M>
username <pi,fi1> Variable characters (32) <M>
Identifier_1 <pi>
Moderator
Curator
Moderator
displayed by
ratings
rating_no <pi> Integer <M>
rating_user Variable characters (32)
art_pool_id Integer
datePosted Date
Rating Integer
Identifier_1 <pi>
users
username <pi> Variable characters (32) <M>
rating_no <fi> Integer
password Variable characters (64)
role Variable characters (12)
Identifier_1 <pi>
Patron
rating
User Ratings
rated by

Physical


on_display
display_no
media_id
rating_no
start_date
art_pool_id
src
num_interactions
avg_rating
last_comment
Display2
hide
selected comment
hide
Comment
Display
comments
comment_no
display_no
comment_use
art_pool_id
datePosted
comment_
select comment
display
display
art_pool
media_id
username
display_no
title
type
artist
src
on_display
Patron
Rating
Patron
comment_nc
username
Moderator
added to display
Moderator
comment_nc
username
Patron
Curator
Moderator
displayed by
ratings
rating_no
rating_user
art_pool_id
datePosted
Rating
users
username
rating_no
password
role
rating
User Ratings
rated by

Application Description

Social Art Online (SAO) is a multi-page web app integrating MySQL database tables with a dynamic React website. The project will utilize the National Gallery of Art database as well as some custom tables to display art, allow users to comment and rate different pieces, and allow for comment moderation. Users will log on to the website and their username and password will be verified through the user table. Each user will have a role (C, P, M) that will allow them to access different functionalities of the app. The GUI will display 7 highlighted artworks from a pool of art selected by the curator. The curator will be able to change the artworks on the page whenever they wish.

Instruction Manual

Front-end

The front end of the application SAO is built using React.js and CSS for styling. To run the front end, the git repo needs to be cloned. This is done by Git Bash or the GUI. After cloning, the dependencies can be installed by using the node package manager (NPM) `npm install` downloads all the dependencies required by the app for running the react app. After the installation is completed, the react app can be started by `npm start`. Any changes made are reflected in real time. The different pages and roles are in the pages directory and the login screen is the first page that greets the user.

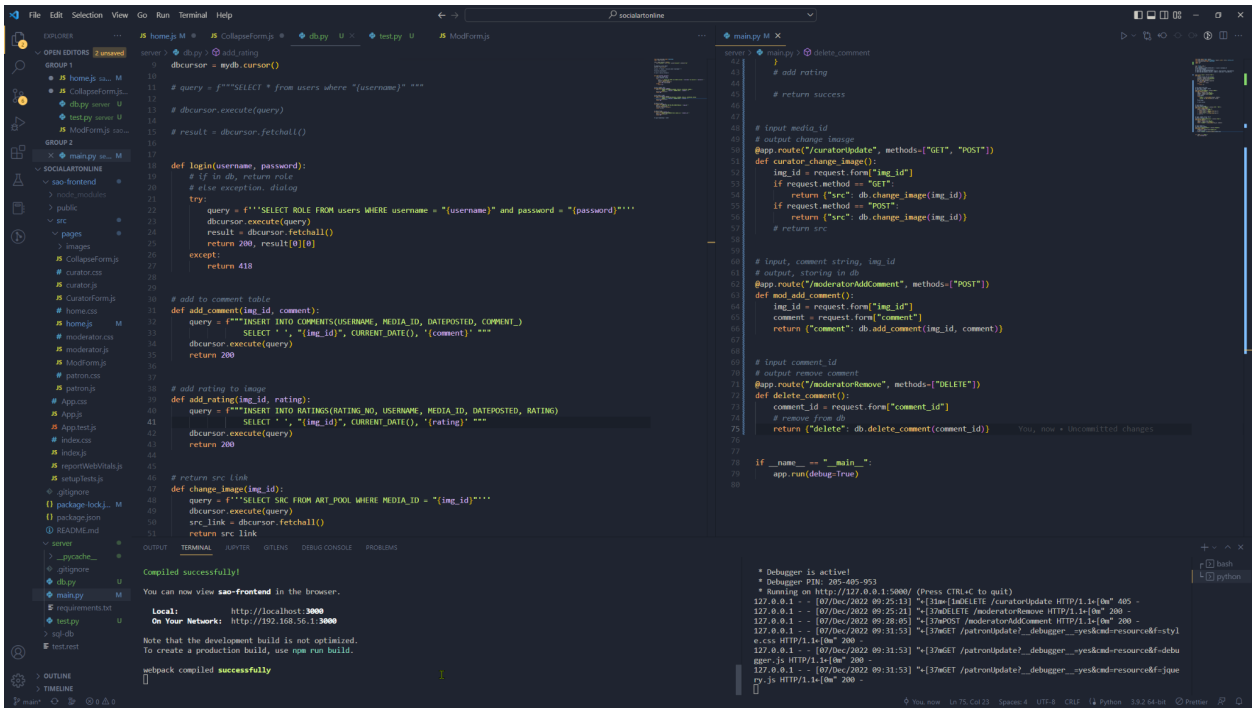
Back-end

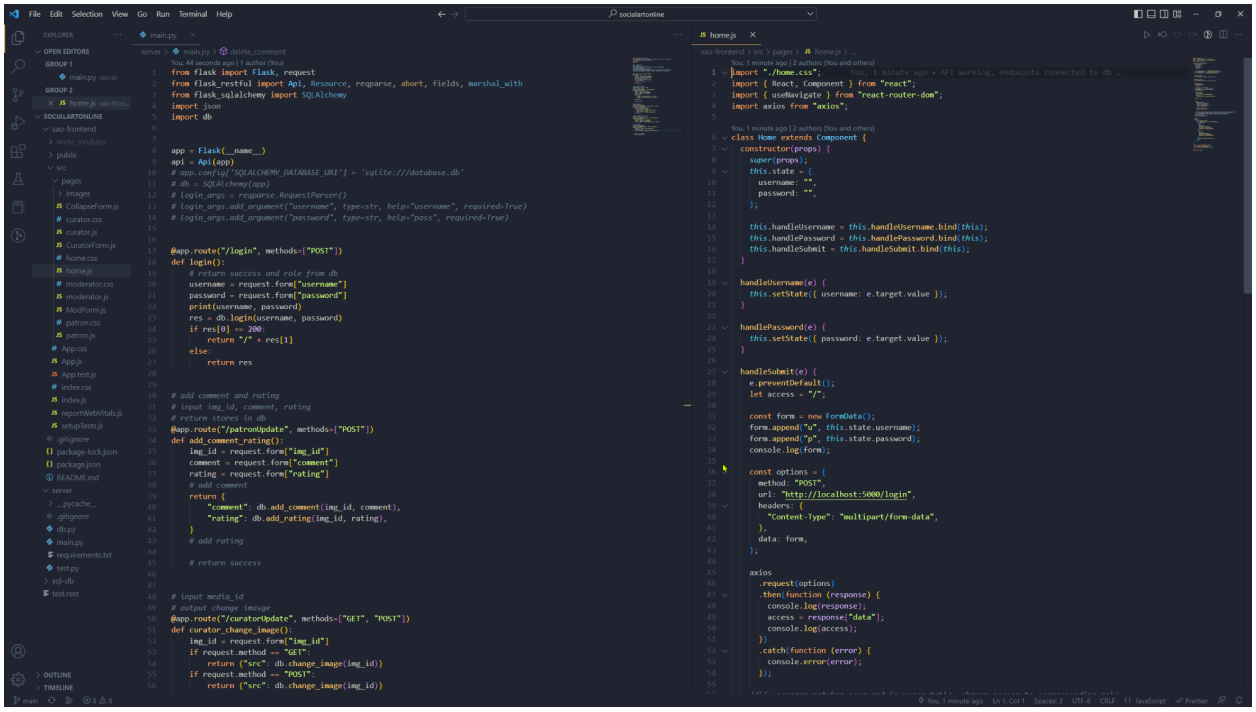
The back end of SAO is made with Python using the Flask framework. It is a lightweight framework capable of running and implementing RESTful methods. Once the git repo is cloned, the requirements as stated in the requirements.txt can be installed by using pip, pip installation program. `Pip install requirements.txt. D`asdf After installing, the server can be started by running main.py. Db.py contains all the queries and the connector for the server to connect to the database. The database needs to be opened or be in the same machine and the details need to be put into db.py in the connector object.

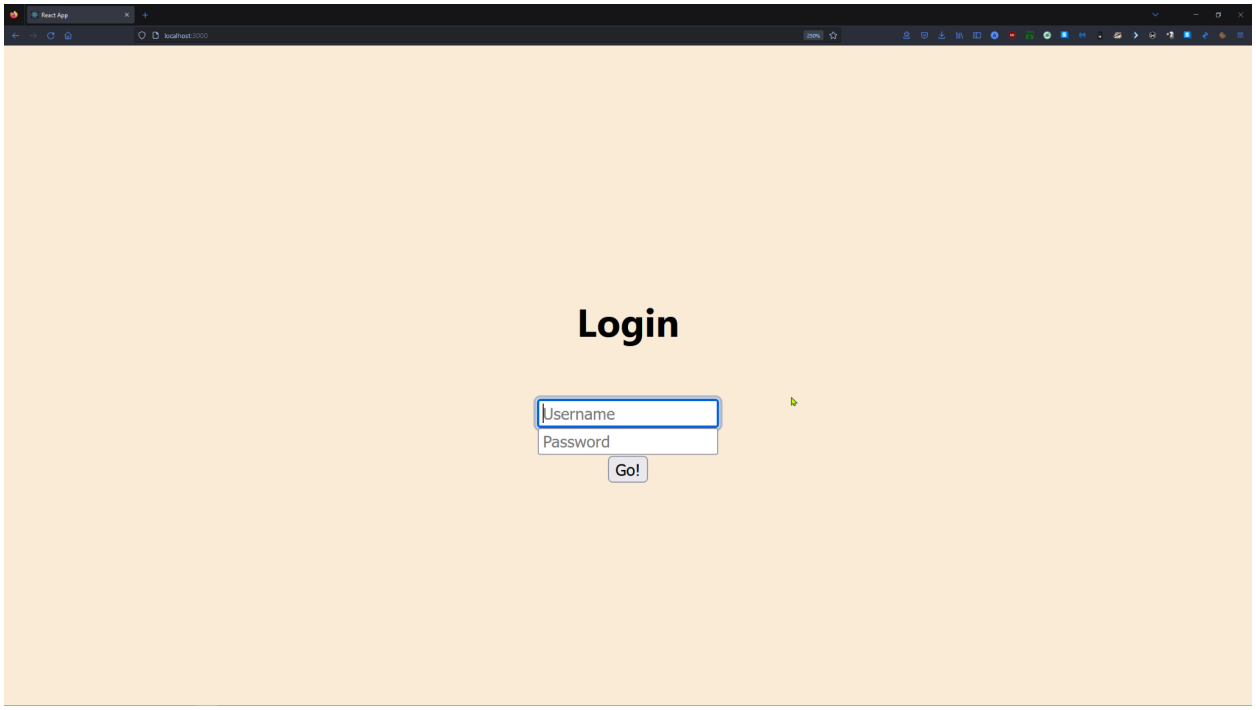
Repository Link

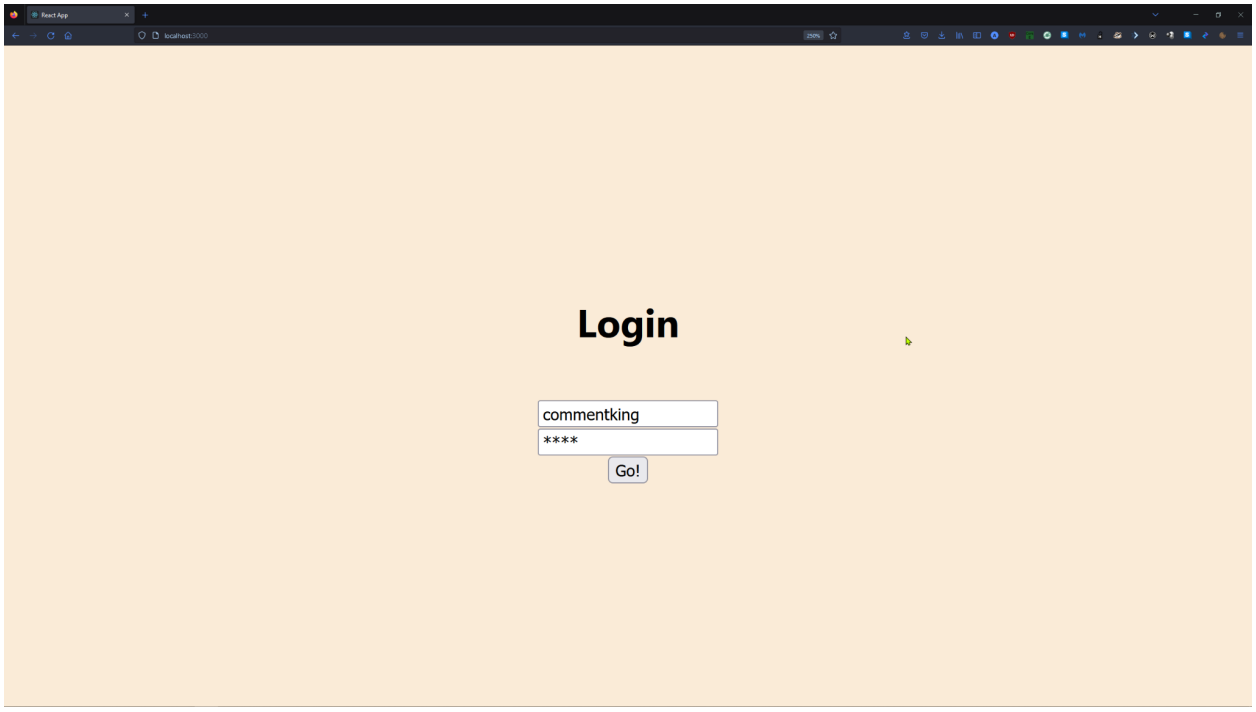
https://github.com/hello-adi/socialartonline

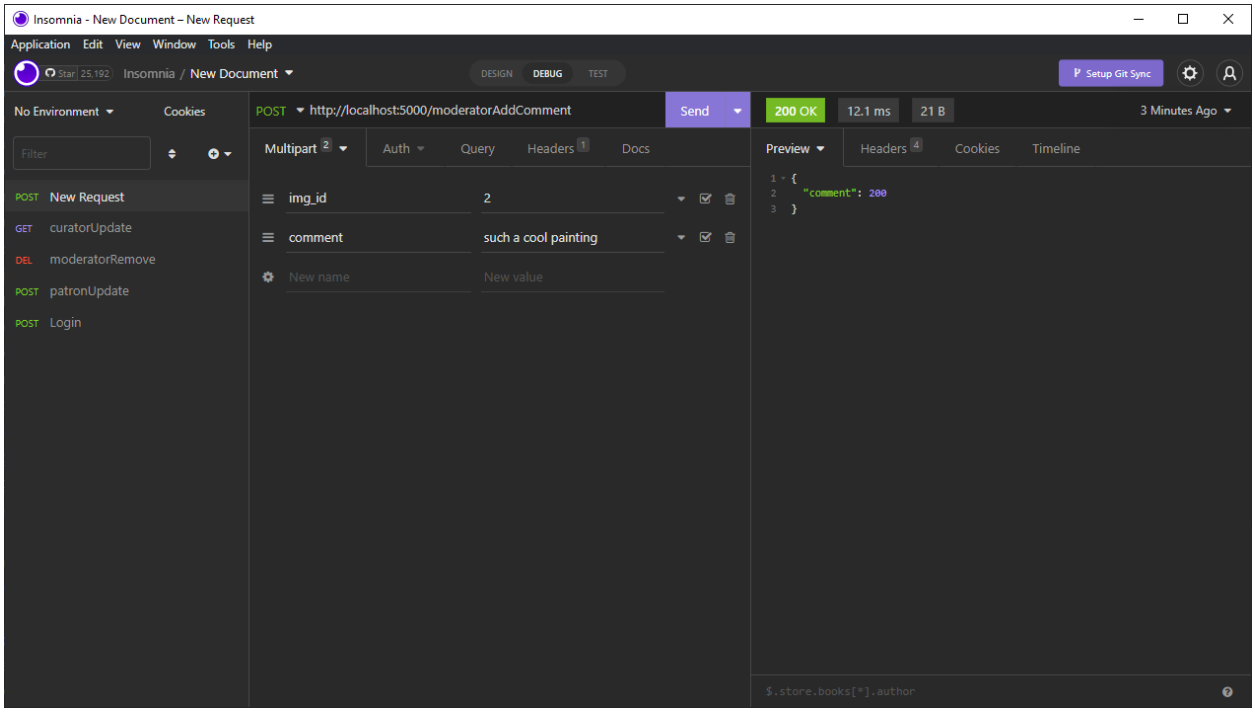
Screenshots

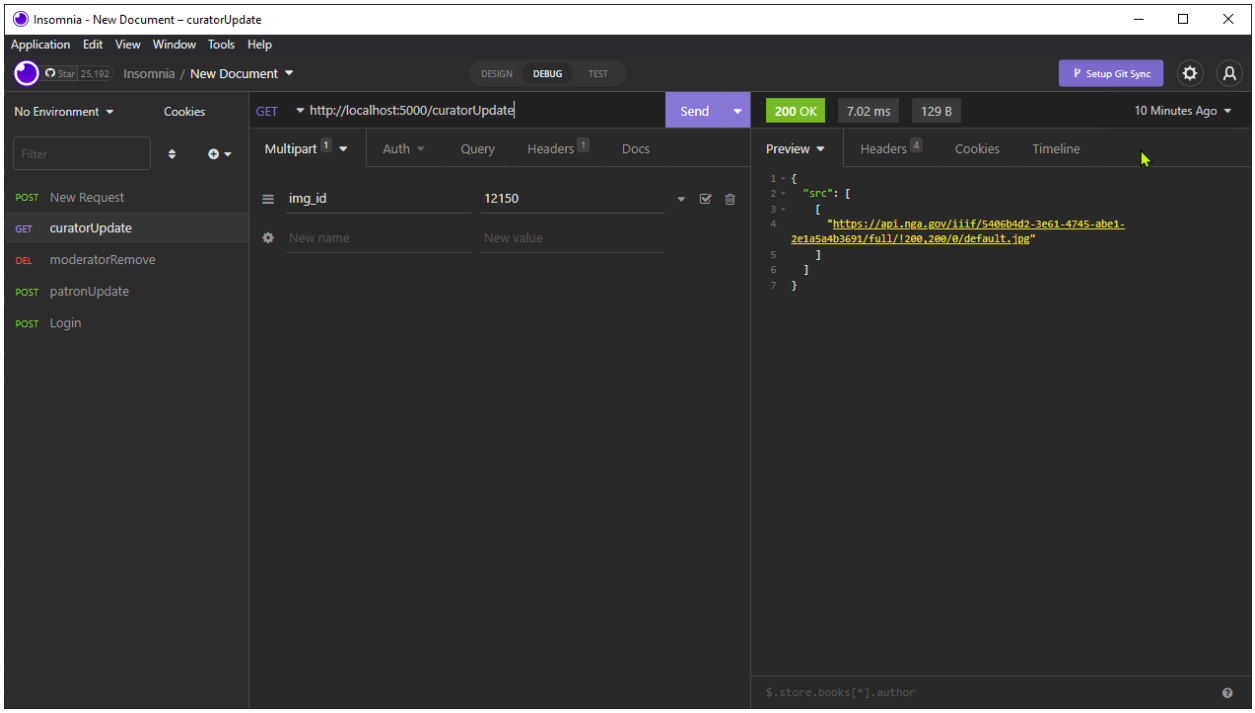


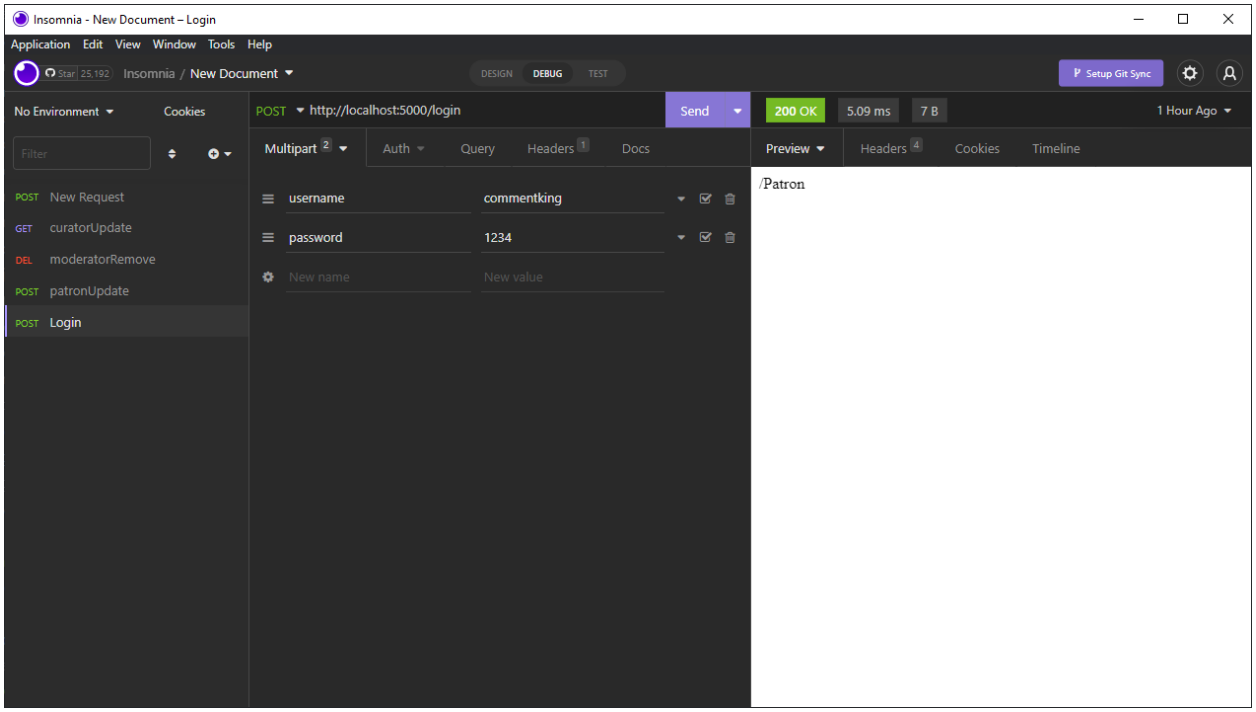


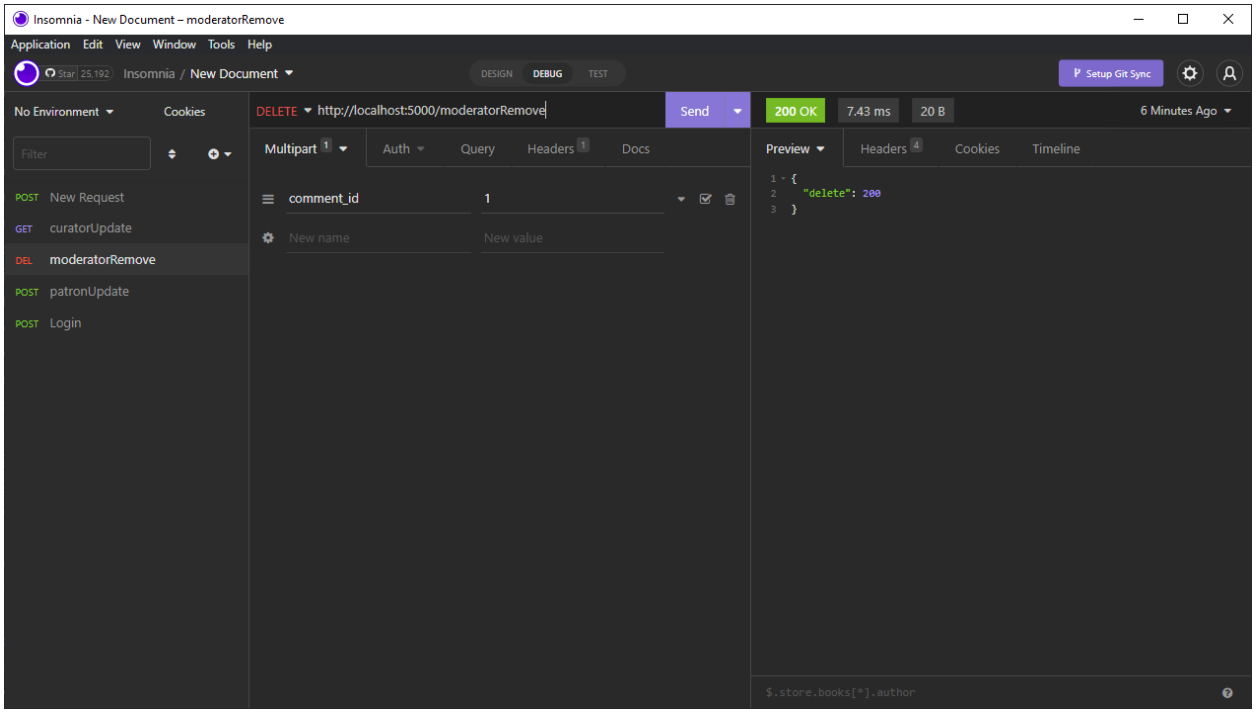

React App
localhost:3000
Login
Username
Password
Go!

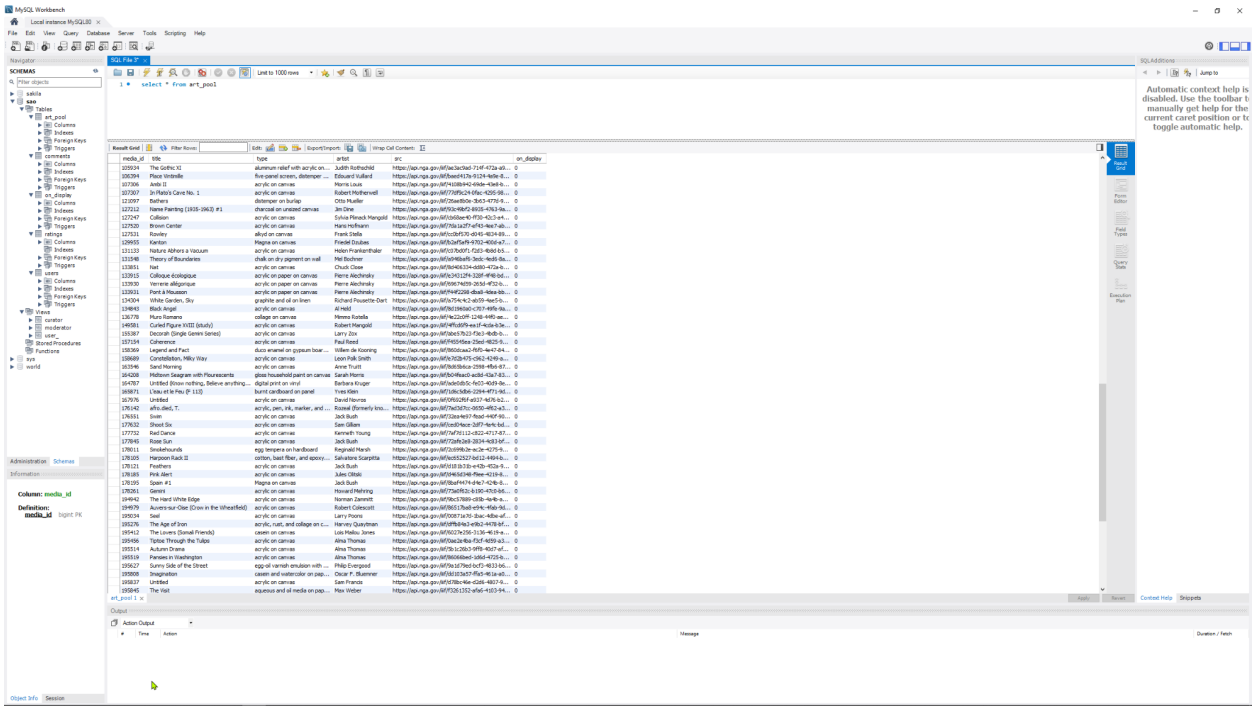

React App
localhost:3000
Login
commentking
****
Go!

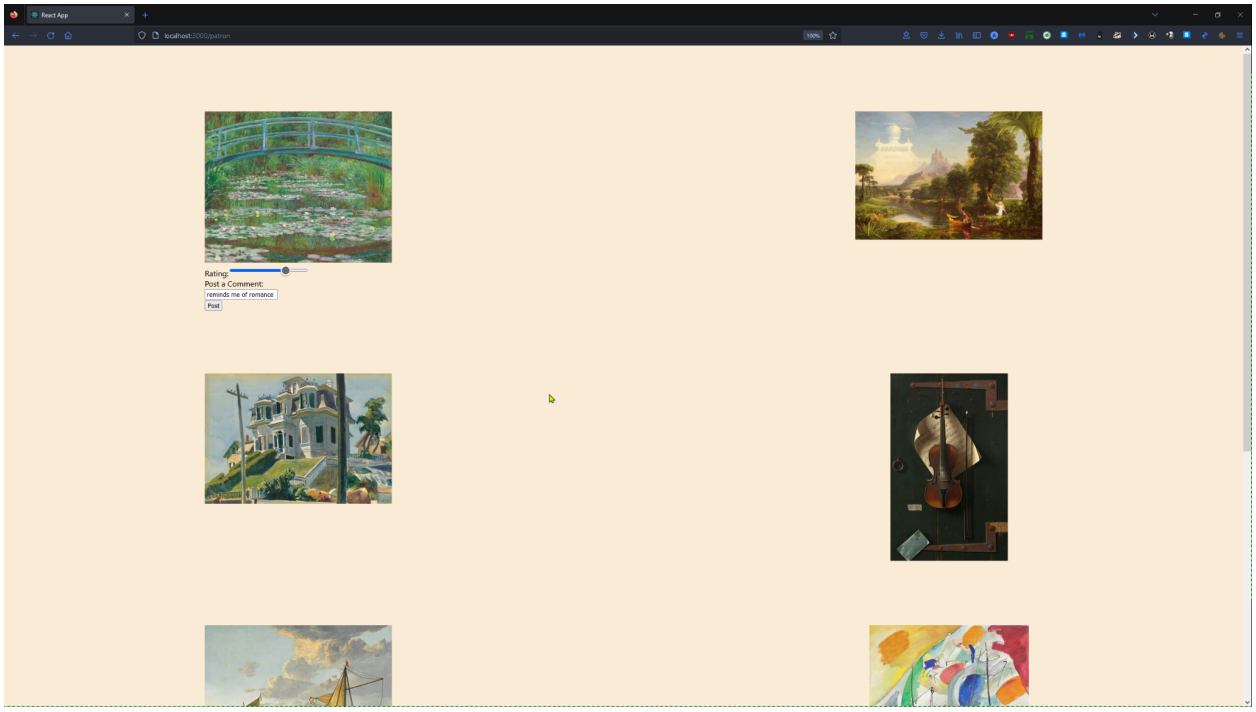

Insomnia - New Document – New Request
Application Edit View Window Tools Help
Insomnia / New Document
DESIGN DEBUG TEST
Setup Git Sync
No Environment
Cookies
POST http://localhost:5000/moderatorAddComment
Send
200 OK
12.1 ms
21 B
3 Minutes Ago
Multipart 2
Auth
Query
Headers 1
Docs
Preview
Headers 4
Cookies
Timeline
POST New Request
GET curatorUpdate
DEL moderatorRemove
POST patronUpdate
POST Login
img_id
2
comment
such a cool painting
New name
New value
{
"comment": 200
}
$.store.books[*].author

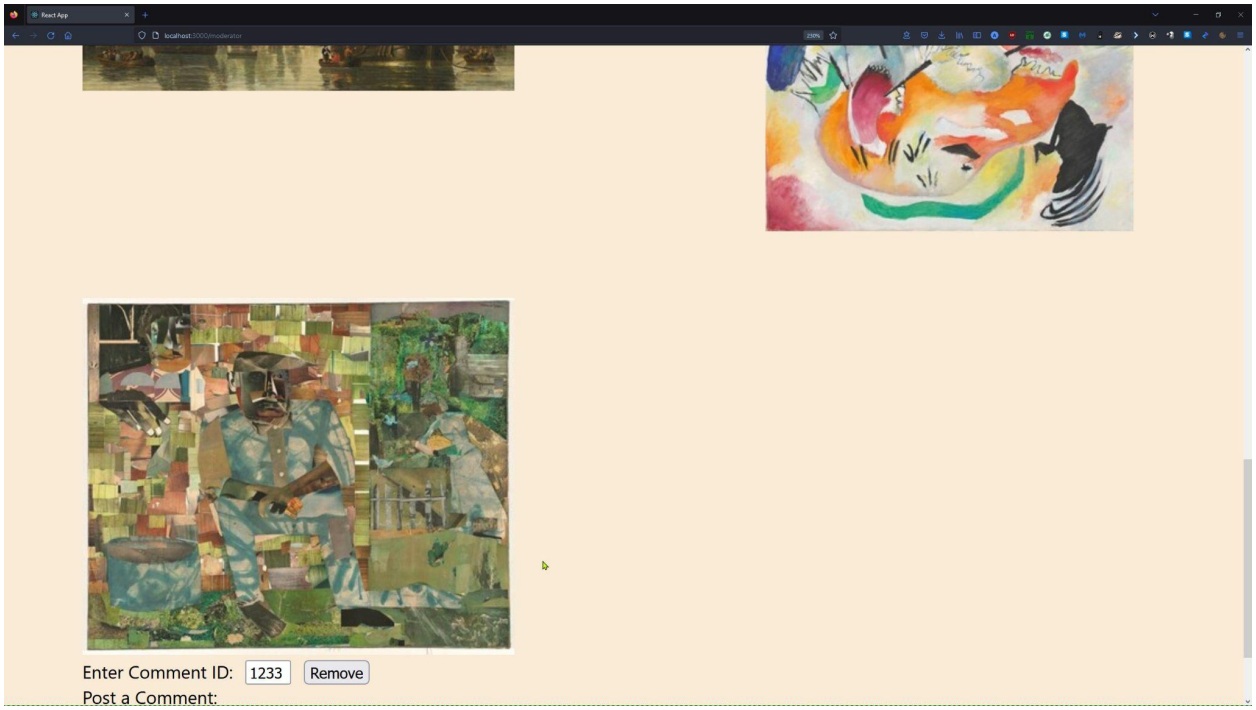

Insomnia - New Document – curatorUpdate
Application Edit View Window Tools Help
Insomnia / New Document
DESIGN DEBUG TEST
Setup Git Sync
No Environment
Cookies
GET http://localhost:5000/curatorUpdate
Send
200 OK
7.02 ms
129 B
10 Minutes Ago
Multipart 1
Auth
Query
Headers 1
Docs
Preview
Headers 4
Cookies
Timeline
POST New Request
GET curatorUpdate
DEL moderatorRemove
POST patronUpdate
POST Login
img_id
12150
New name
New value
{
"src": [
[
"https://api.nga.gov/iiif/5406b4d2-3e61-4745-abe1-2e1a5a4b3691/full/!200,200/0/default.jpg"
]
]
}
$.store.books[*].author

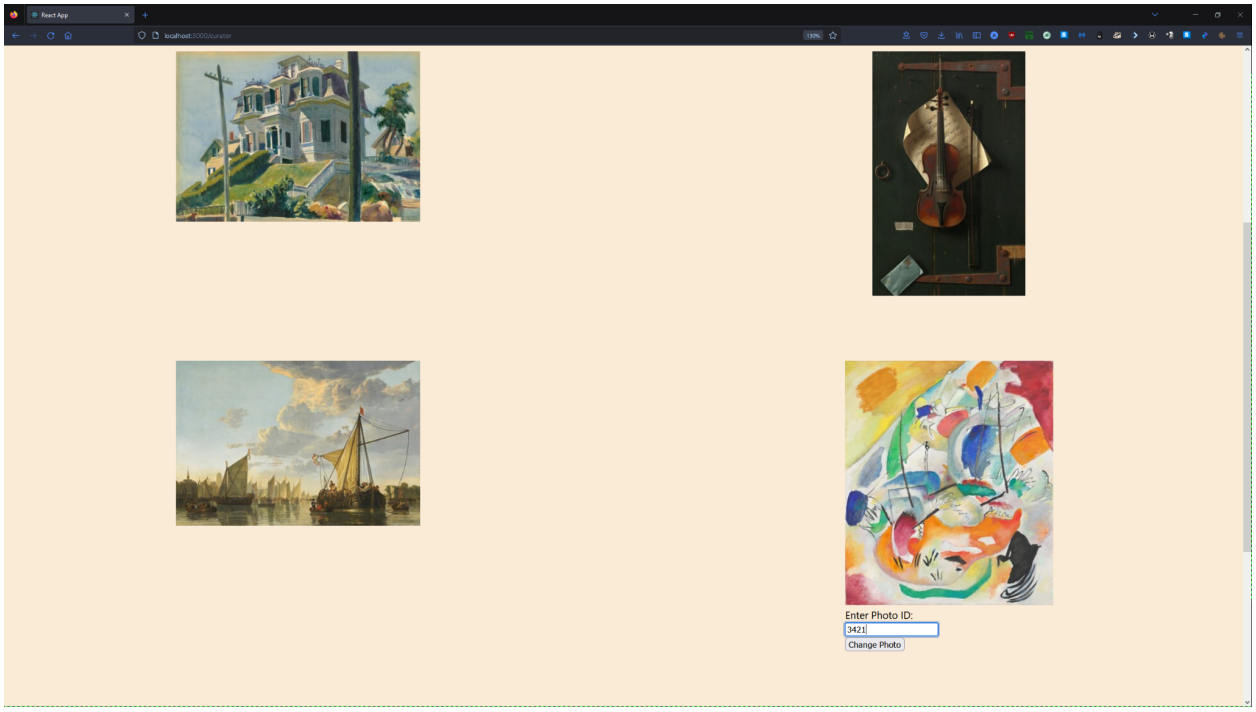

Insomnia - New Document – Login
Application Edit View Window Tools Help
Insomnia / New Document
DESIGN DEBUG TEST
Setup Git Sync
No Environment
Cookies
POST http://localhost:5000/login
Send
200 OK
5.09 ms
7 B
1 Hour Ago
Multipart
Auth
Query
Headers
Docs
Preview
Headers
Cookies
Timeline
username
commentking
password
1234
New name
New value
/Patron
POST New Request
GET curatorUpdate
DEL moderatorRemove
POST patronUpdate
POST Login


Insomnia - New Document – moderatorRemove
Application Edit View Window Tools Help
Insomnia / New Document
DESIGN DEBUG TEST
Setup Git Sync
No Environment
Cookies
DELETE http://localhost:5000/moderatorRemove
Send
200 OK
7.43 ms
20 B
6 Minutes Ago
Multipart
Auth
Query
Headers
Docs
Preview
Headers
Cookies
Timeline
comment_id
1
New name
New value
{
"delete": 200
}
$.store.books[*].author
POST New Request
GET curatorUpdate
DEL moderatorRemove
POST patronUpdate
POST Login


MySQL Workbench
select * from art_pool


React App
localhost:3000/patron
Rating:
Post a Comment:
reminds me of romance
Post


Enter Comment ID:
1233
Remove
Post a Comment:


Enter Photo ID:
3421
Change Photo

Lessons Learned

One of the most important lessons learnt was the fact that good planning is always necessary for a good outcome. We were pressed for time and we had to divide the work within the team in an efficient and effective manner. Further, it requires a great sense of trust to ask for help from others and strive towards a greater goal. Another important lesson we learned was that during the conceptual stages, it is essential to iron out any kinks, otherwise they might be too expensive to fix later. Agile methodologies must be followed to keep up with the changing use cases.Also, testing must take priority as every piece of code before being released into production must be checked for any unseen bugs. Therefore, this experience was a great learning effect for our future endeavors.