
Test Compliance Report

for

Online Medical Consultation System

Prepared by :-

- 1. Ashwani Kumar Kamal**
- 2. Archit Mangrulkar**
- 3. Shiladitya De**

Team:ASMR

Group: 48

30th March, 2022

Compliance Report for Backend Classes and Database Management

Testing Login () :

Case I : Giving Wrong Username

Test Case Failed

The screenshot shows the Postman interface with a failed login attempt. The request URL is `http://localhost:8000/api/auth/login`. The body contains the following JSON:

```
1 "email": "admin@1.com",
2 "password": "a"
```

The response status is 500, and the error message is:

```
/rest_framework/views.py", line 509, in dispatch
    response = self.handle_exception(exc)
    File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages
/rest_framework/views.py", line 469, in handle_exception
    self.raise_uncaught_exception(exc)
    File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages
/rest_framework/views.py", line 480, in raise_uncaught_exception
    raise exc
    File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages
/rest_framework/views.py", line 506, in dispatch
    response = handler(request, *args, **kwargs)
    File "/home/sneakypotato/junkyard/test/src/api/views.py", line 88,
in post
    user = User.objects.get(email=request.data['email'])
    File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages
/django/db/models/manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
    File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages
/django/db/models/query.py", line 496, in get
    raise self.model.DoesNotExist(
api.models.User.DoesNotExist: User matching query does not exist.
[30/Mar/2022 14:56:50] "POST /api/auth/login HTTP/1.1" 500 101265
```

Case II: Giving Correct Username and password

Test Case Passed

The screenshot shows the Postman interface with a successful login attempt. The request URL is `http://localhost:8000/api/auth/login`. The body contains the following JSON:

```
1 "email": "admin@asmr.com",
2 "password": "a"
```

The response status is 200, and the JSON data is:

```
1 {
2     "success": true,
3     "statusCode": 200,
4     "message": "User logged in successfully",
5     "access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
6     ey30b2tibl9e0B1jjoYwJ2XNfIizXmHjxNjQ0ajjuMTY4LC3pYXQ10jE2Ng;
7     osZqgD9WzvNGT_gsqpRiaQ085-x7ldc4jk",
8     "refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
9     ey30b2tibl9e0B1jjoYwJ2XNfIizXmHjxNjQ0ajjuMTY4LC3pYXQ10jE2Ng;
10    L1U_0y05XmedjYANVPT77M2dLmLRInvo-peSQuM",
11    "authenticated": 1
12    "email": "admin@asmr.com",
13    "role": "1"
14 }
```

Testing Refresh Token:

Case I : Refresh token successful
Test Case Passed

The screenshot shows the Postman application interface. The top bar displays 'Activities' and 'Alacrity' with the date 'Wed Mar 30 20:30'. The main workspace is titled 'My Workspace' and contains a collection named 'asmr_ocms / refresh'. A single POST request is selected, targeting the URL `http://localhost:8000/api/auth/token/refresh/`. The 'Body' tab is active, showing a JSON payload with a single key 'refresh': `{"refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYmVmVzaC1sImV4cCI6MTY0OTQzNDY2OSwiAiOiJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYmVmVzaC1sImV4cCI6MTY0OTQzNDY2OSwiAiM0YSLCJqdGk1o1JhMW3hNTh1MwNmMdg0ZTQ4YWU1ZjQ2MwQ2Njg2Mzc4YiInVzZXfaWQ1oJF9.eyJ079Mw9GbtaT2JEK68Hgc3sgc6cRCKtp1M6Czs"}`. The response pane shows the server's log output, which includes the generated access token and refresh token, along with other API logs. The bottom status bar indicates 'Find and Replace' and 'Console'.

Case II : Refresh token unsuccessful Test Case Failed

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, APIs, environments, and other tools. In the center, a request is being made to `http://localhost:8000/api/auth/token/refresh/`. The request method is POST, and the body contains a JSON object with a single key `"refresh"`. The response body shows a JSON error message: `{"detail": "Token is invalid or expired", "code": "token_not_valid"}`. To the right of the request, there is a terminal window showing command-line activity related to the project, including a Django development server running and a failed POST request to the refresh endpoint.

Testing Get All Users:

Case I: Get patients list using admin's expired token

Test Case Failed

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, APIs, environments, and other tools. In the center, a request is being made to `http://localhost:8000/api/auth/patients`. The request method is GET, and the authorization header is set to `Bearer Token`. A note in the request panel states: "The authorization header will be automatically generated when you send the request. Learn more about authorization >". The response body shows a JSON error message: `{"detail": "Given token not valid for any token type.", "code": "token_not_valid", "messages": [{"token_class": "AccessToken", "token_type": "access", "message": "Token is invalid or expired"}]}`. To the right of the request, there is a terminal window showing command-line activity related to the project, including a Django development server running and a failed GET request to the patients endpoint.

Case II : Get patients list using admin's correct token

Test Case Passed

The screenshot shows the Postman application interface. The top bar displays 'Activities Alacrity' and the date 'Wed Mar 30 20:28'. The main workspace is titled 'asmr_ocms / Get all Users'. A GET request is selected with the URL 'http://localhost:8000/api/auth/patients'. The 'Authorization' tab is active, showing a Bearer Token input field with the value 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.e...'.

The response body is displayed in a code block:

```
File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages/rest_framework/views.py", line 480, in raise_uncaught_exception
    raise exc
File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages/rest_framework/views.py", line 506, in dispatch
    response = handler(request, *args, **kwargs)
File "/home/sneakypotato/junkyard/test/src/api/views.py", line 88,
in post
    user = User.objects.get(email=request.data['email'])
File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages/django/db/models/manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
File "/home/sneakypotato/junkyard/test/lib/python3.10/site-packages/django/db/models/query.py", line 496, in get
    raise self.model.DoesNotExist()
api.models.User.DoesNotExist: User matching query does not exist.
[30/Mar/2022 14:56:50] "POST /api/auth/login HTTP/1.1" 500 101265
[30/Mar/2022 14:57:48] "POST /api/auth/login HTTP/1.1" 200 613
^[[2~Unauthorized: /api/auth/patients
[30/Mar/2022 14:58:17] "GET /api/auth/patients HTTP/1.1" 401 183
[30/Mar/2022 14:58:44] "GET /api/auth/patients HTTP/1.1" 200 653
]
```

Case III : Get Doctor's list using admin's expired token

Test Case Failed

The screenshot shows the Postman application interface. The top bar displays "Activities Alacrity" and the date "Wed Mar 30 20:53". The main workspace shows a collection named "asmr_ocms" with a "Get all Users" endpoint selected. The request method is "GET" and the URL is "http://localhost:8000/api/auth/doctors". The "Authorization" tab is active, showing "Bearer Token" selected and a token input field containing "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1Ni...". Below the request, the "Body" tab is open, displaying a JSON response:

```
1  "detail": "Given token not valid for any token type",
2  "code": "token_not_valid",
3  "messages": [
4    {
5      "token_class": "AccessToken",
6      "token_type": "access",
7      "message": "Token is invalid or expired"
8    }
9  ]
```

The response body is highlighted in red. The status bar at the bottom right indicates "System check identified no issues (0 silenced)." and shows log entries for Django development server and a failed API call to "/api/auth/doctors".

Case IV: Get doctors list using admin's correct token

Test Case Passed

The screenshot shows the Postman application interface. At the top, it displays "Wed Mar 30 20:53". The left sidebar shows a workspace named "asmr_ocms" containing several API collections: "Register User", "Login User", "Get all Users", "Edit User", "Register Appointment", "Get appointments", "Edit appointment", "Register Hospital", "Get Hospitals", "Ping", "Refresh", "Me", and "Junkyard". The main area shows a GET request to "http://localhost:8000/api/auth/doctors". The "Authorization" tab is selected, showing a Bearer Token input field with a placeholder "Token" and a value "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ...". Below the request, the "Body" tab is selected, showing a JSON response with a success status and two doctor objects. On the right side, there is a terminal window showing command-line activity related to the test.

```
source bin/activate
~/junkyard/test via v3.10.2 (test)
cd src
src on develop [!?] via v3.10.2 (test)
python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
March 30, 2022 - 15:21:50
Django version 4.0.3, using settings 'core.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Unauthorized: /api/auth/patients
[30/Mar/2022 15:22:04] "GET /api/auth/patients HTTP/1.1" 401 183
[30/Mar/2022 15:22:15] "POST /api/auth/login HTTP/1.1" 200 613
[30/Mar/2022 15:22:35] "POST /api/auth/token/refresh/ HTTP/1.1" 200 4
83
[30/Mar/2022 15:22:46] "GET /api/auth/patients HTTP/1.1" 200 653
[30/Mar/2022 15:23:26] "GET /api/auth/doctors HTTP/1.1" 200 2065
[]
```

Request Details:

- Type: GET
- URL: http://localhost:8000/api/auth/doctors
- Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ...

Response Body (Pretty JSON):

```
1  {
2      "success": true,
3      "status_code": 200,
4      "message": "Successfully fetched all doctors",
5      "doctors": [
6          {
7              "id": 7,
8              "email": "doctor@1.com",
9              "password": "pbkdf2_sha256$320000$RGBJ8ej0oP9F3qg1jjB3py$1es",
10             "first_name": "John",
11             "last_name": "Doe",
12             "address": "Azad",
13             "contact": "9876543210",
14             "hospital": 1,
15             "speciality": "Dental",
16             "pincode": "202001",
17             "pending": 0
18         },
19         {
20             "id": 8,
21             "email": "doctor@2.com",
22         }
23     ]
24 }
```

Test get user details (Class name “me”)

Test Case Passed

The screenshot shows the Postman application interface. The top bar displays the date and time as "Wed Mar 30 20:30" and the title "Activities • Alacrity". The main workspace shows a collection named "asmr_ocms" with several API endpoints listed under it. One endpoint, "GET Me", is selected and highlighted in orange. The request URL is set to "http://localhost:8000/api/auth/me". The "Authorization" tab is active, showing a Bearer Token with a long token value. The "Body" tab contains a JSON response object:

```
1 {  
2     "success": true,  
3     "status_code": 200,  
4     "message": "Successfully fetched user",  
5     "user": {  
6         "email": "admin@djangocom",  
7         "password": "pbkdf2_sha256$32000$1a8LwxuR3AXM4VIKA0KJwt$32MOEHkv",  
8         "role": 1  
9     }  
10 }
```

The "Headers" tab shows 10 headers. The "Test Results" tab is visible at the bottom. The right side of the screen displays a terminal window showing the command-line history of the API interactions.

Testing Edit Appointment :

Case I: Editing appointment using Patient's token

The screenshot shows the Postman application interface. On the left, the sidebar displays collections, APIs, environments, mock servers, monitors, flows, and history. The main workspace shows a collection named 'asmr_ocms' with several API endpoints listed under it. A specific endpoint, 'PUT /api/auth/appointments/2/' (Edit appointment), is selected and highlighted in blue. The request body is set to 'JSON' and contains the following data:

```
1  "doctor_id": 1,
2  "patient_id": 1,
3  "description": "appointment",
4  "status": 1
```

The 'Body' tab of the response panel shows the error message: "You do not have the permission to edit an appointment". The 'Test Results' tab shows a system check output:

```
Performing system checks...
System check identified no issues (0 silenced).
March 30, 2022 - 15:04:58
Django version 4.0.3, using settings 'core.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
/home/sneakypotato/junkyard/test/src/api/views.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...
```

Below this, another system check output is shown:

```
System check identified no issues (0 silenced).
March 30, 2022 - 15:05:42
Django version 4.0.3, using settings 'core.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Unauthorized: /api/auth/users/2/
[30/Mar/2022 15:05:44] "PUT /api/auth/users/2/ HTTP/1.1" 401 55
Entering views put request
Unauthorized: /api/auth/appointments/2/
[30/Mar/2022 15:06:01] "PUT /api/auth/appointments/2/ HTTP/1.1" 401 5
5
```

Test Case Failed

Case II : Editing appointment using Doctor's / Admin's token

Test Case Passed

Compliance Report for the Frontend GUI Interface

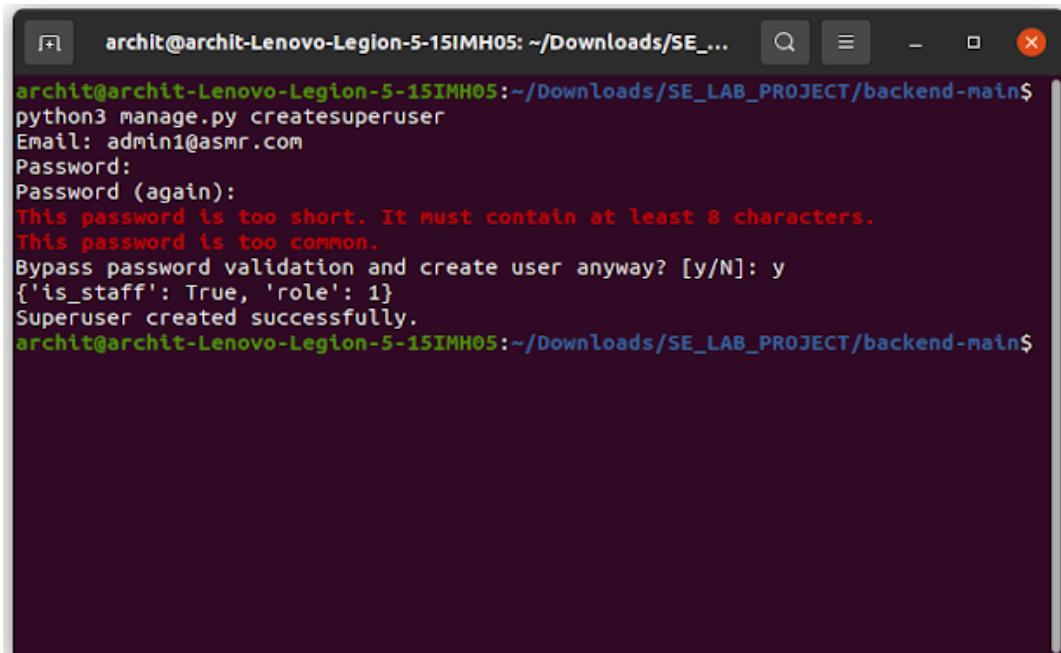
Testing Report for Admin Register():

Input: (<email>,<password>)

Case I : Mail ID is used for the first time

Correct : (email : admin1@asmr.com, password :a)

Test Case Passed

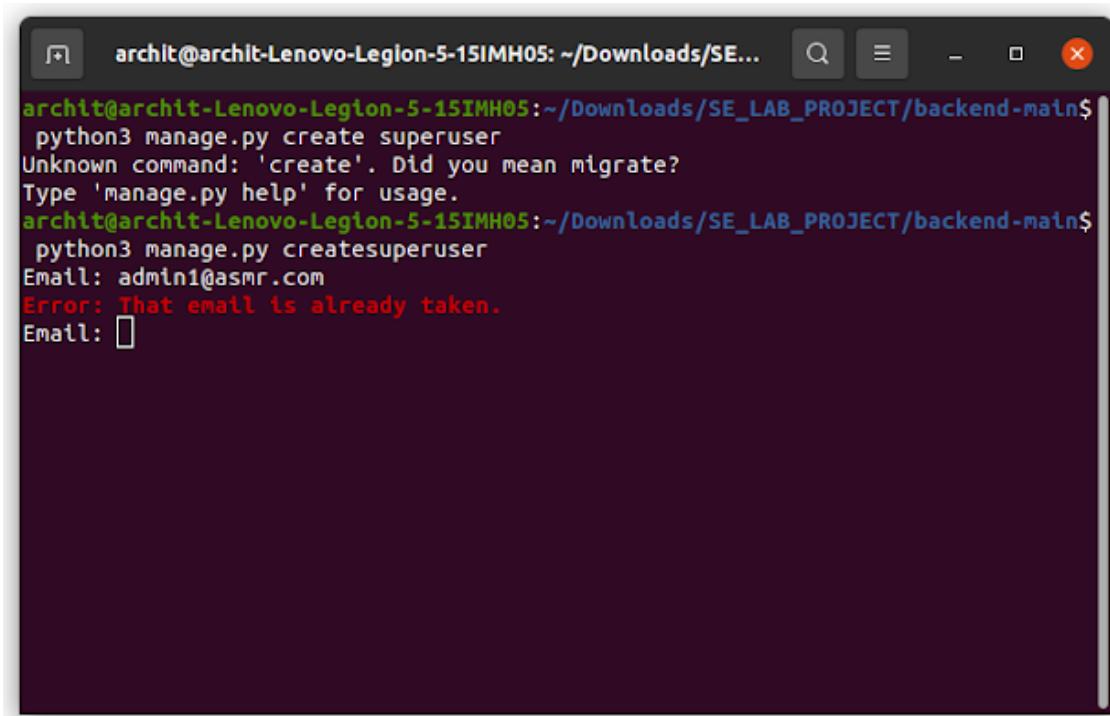


```
archit@archit-Lenovo-Legion-5-15IMH05:~/Downloads/SE_LAB_PROJECT/backend-main$ python3 manage.py createsuperuser
Email: admin1@asmr.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
{'is_staff': True, 'role': 1}
Superuser created successfully.
archit@archit-Lenovo-Legion-5-15IMH05:~/Downloads/SE_LAB_PROJECT/backend-main$
```

Case II: Same Mail ID is used again which shows error

Incorrect : (email : admin1@asmr.com, password : NA)

Test Case Failed



```
archit@archit-Lenovo-Legion-5-15IMH05:~/Downloads/SE_LAB_PROJECT/backend-main$ python3 manage.py create superuser
Unknown command: 'create'. Did you mean migrate?
Type 'manage.py help' for usage.
archit@archit-Lenovo-Legion-5-15IMH05:~/Downloads/SE_LAB_PROJECT/backend-main$ python3 manage.py createsuperuser
Email: admin1@asmr.com
Error: That email is already taken.
Email: 
```

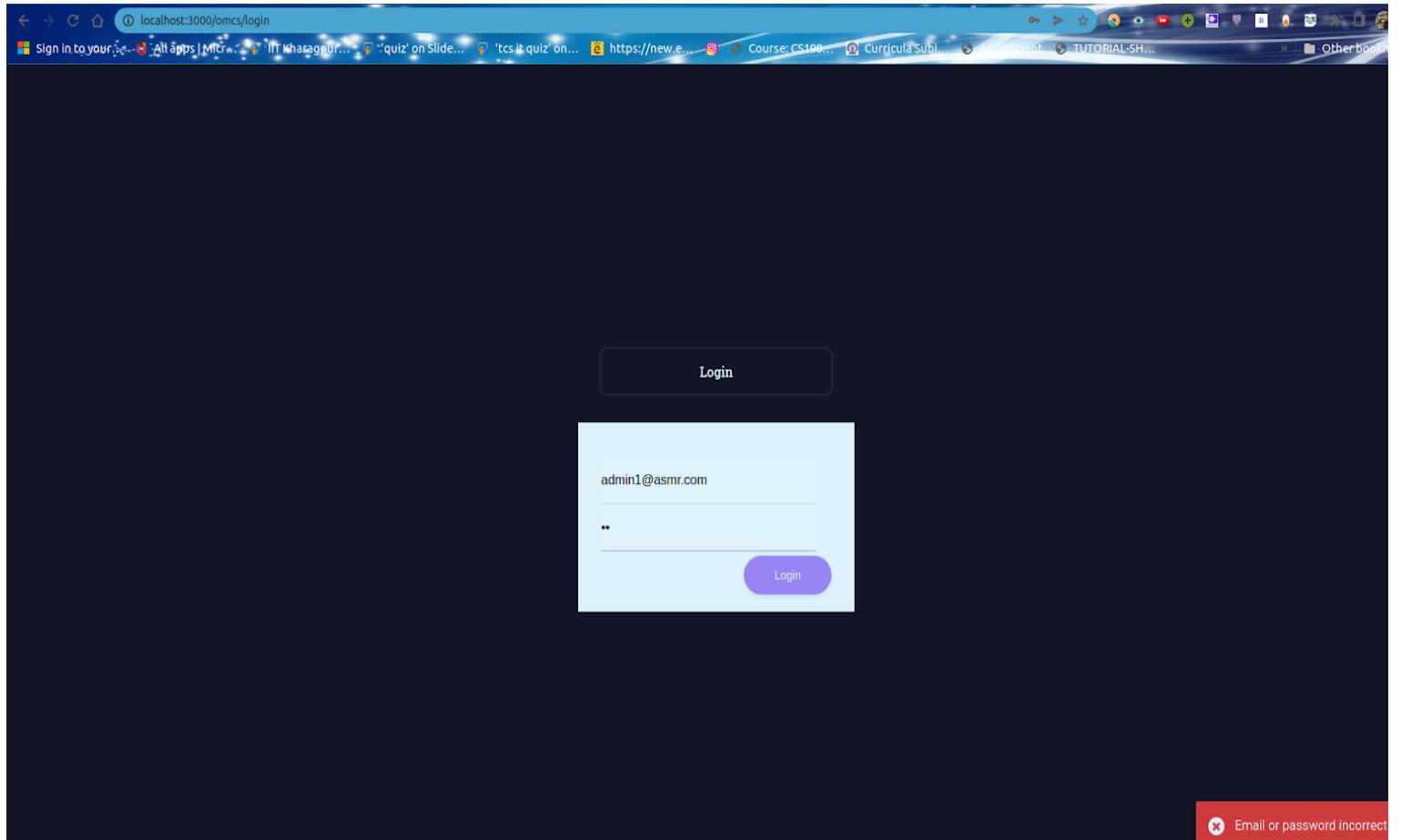
Testing Report for User Login () : (*The mail ID used is of Admin*)

Input: (<email>, <password>)

Case I : Incorrect password but correct email ID

Incorrect : (email: admin1@asmr.com, password: ab)

Test Case Failed



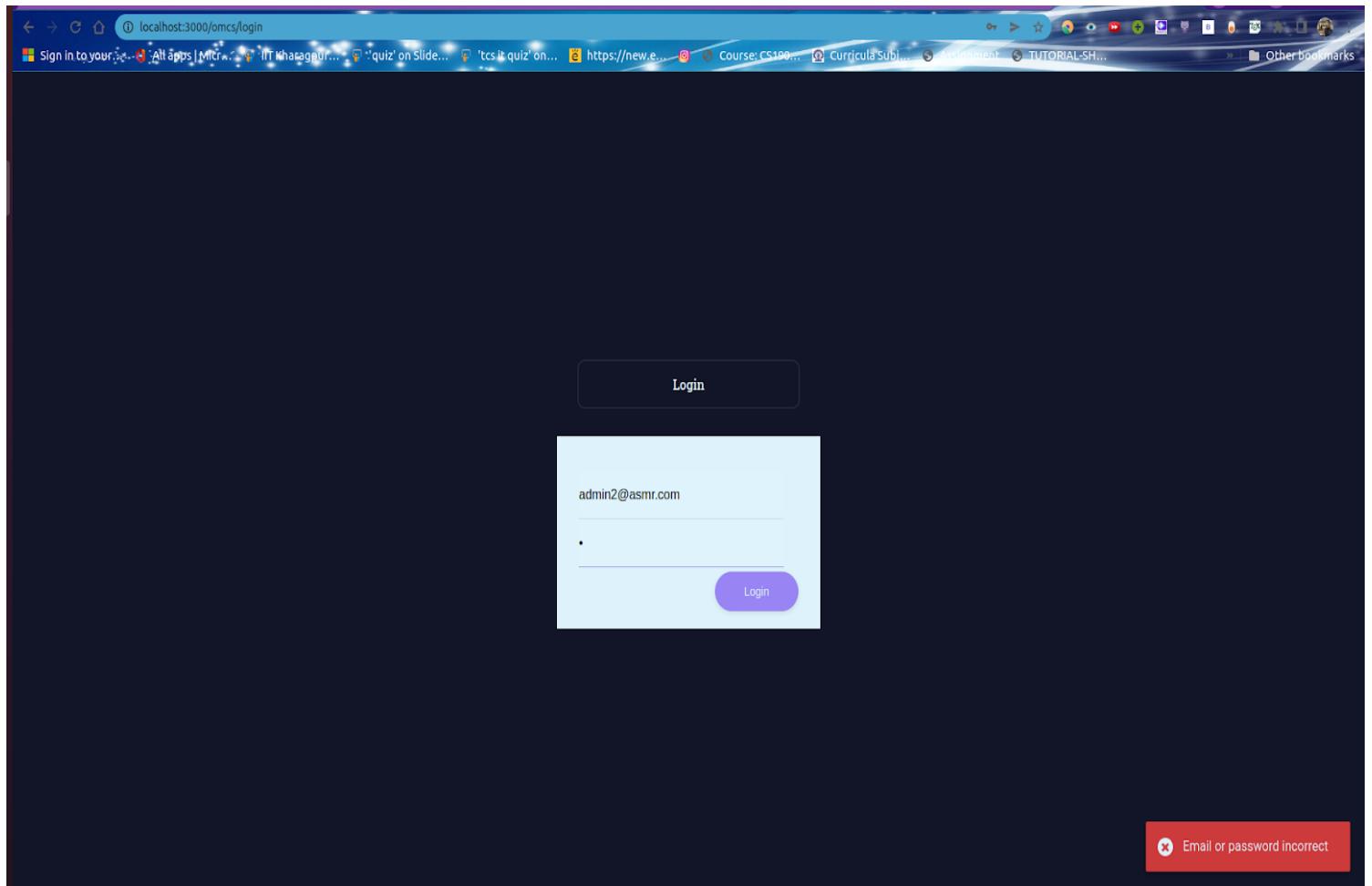
Case II : Correct email ID and Password
Correct : (email: admin1@asmr.com, password: a)

Test Case Passed

A screenshot of a web browser window showing a successful login to a profile page. The URL in the address bar is `localhost:3000/omcs/me`. The browser's toolbar shows various pinned and open tabs. The main content area has a dark header with navigation links: "Profile", "List of Doctors", "List of Hospitals", and a yellow "Log Out" button. Below this is a white rectangular box titled "PROFILE". Inside the box, the text "You are currently logged in as" is followed by the email address "admin1@asmr.com". In the bottom right corner of the main content area, there is a green success message box containing a checkmark icon and the text "Successfully logged in".

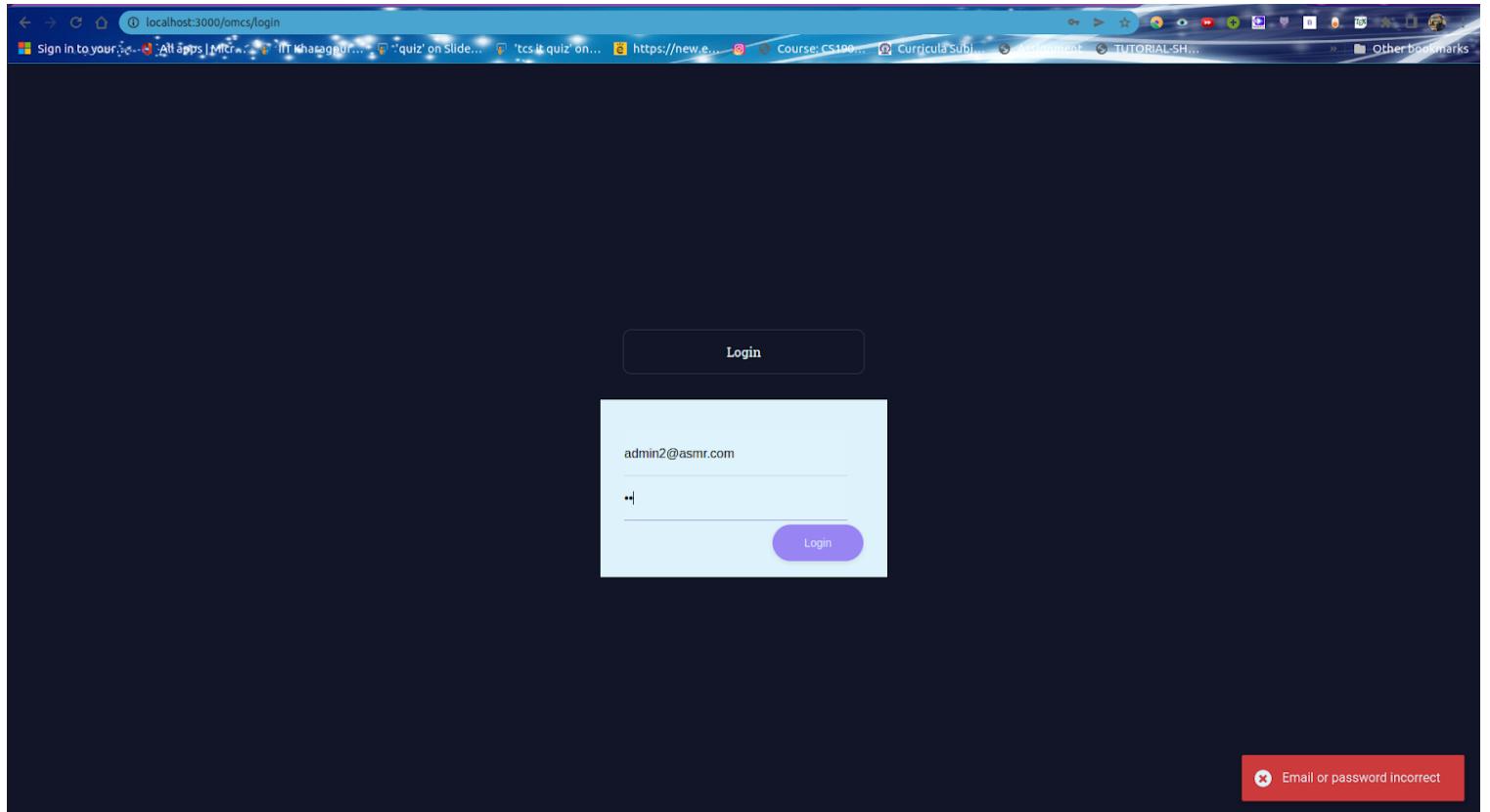
Case III : Incorrect email ID but correct password
Incorrect : (email: admin1@asmr.com, password: a)

Test Case Failed



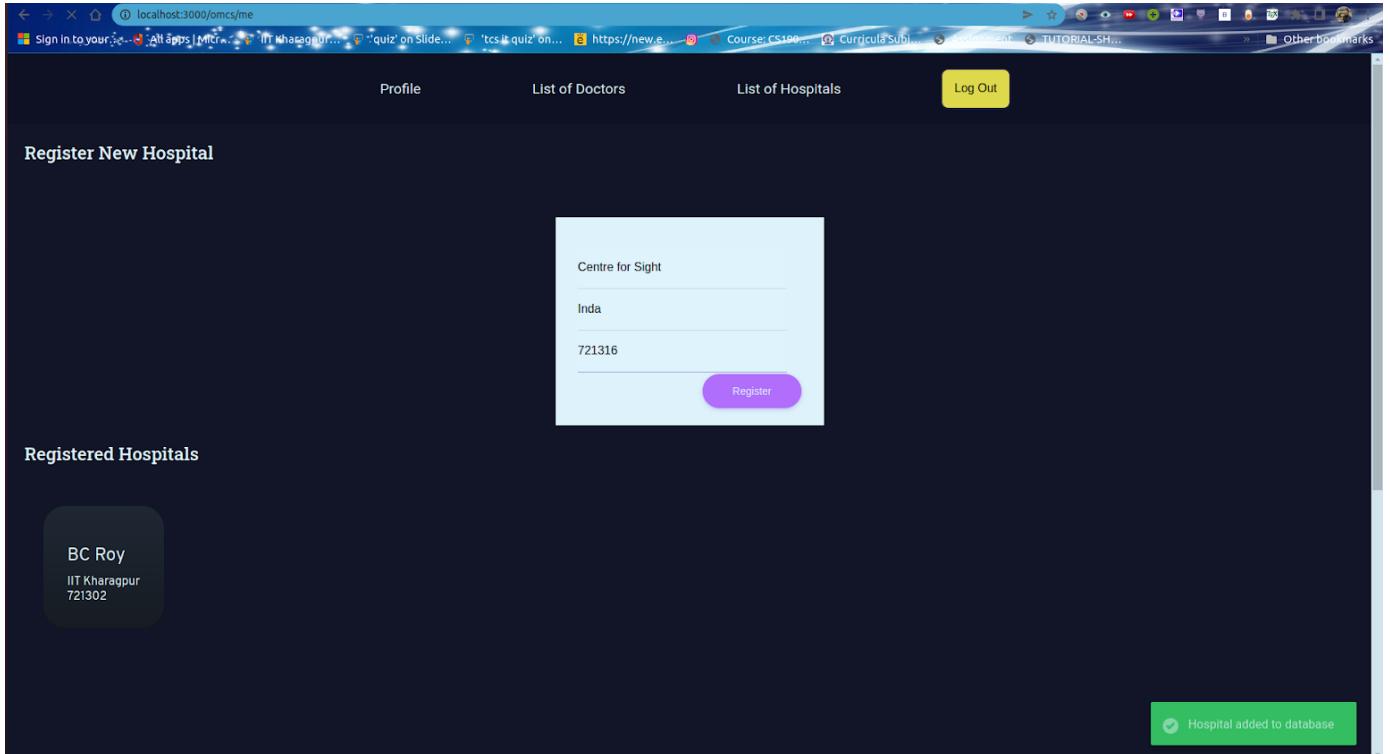
Case IV : Incorrect email ID but correct password
Incorrect : (email: admin2@asmr.com, password: ab)

Test Case Failed



Testing Hospital Registration

Input : (<Name>, <Address>, <Pin-code>)
Test Case Passed



localhost:3000/omcs/me

Profile List of Doctors List of Hospitals Log Out

Register New Hospital

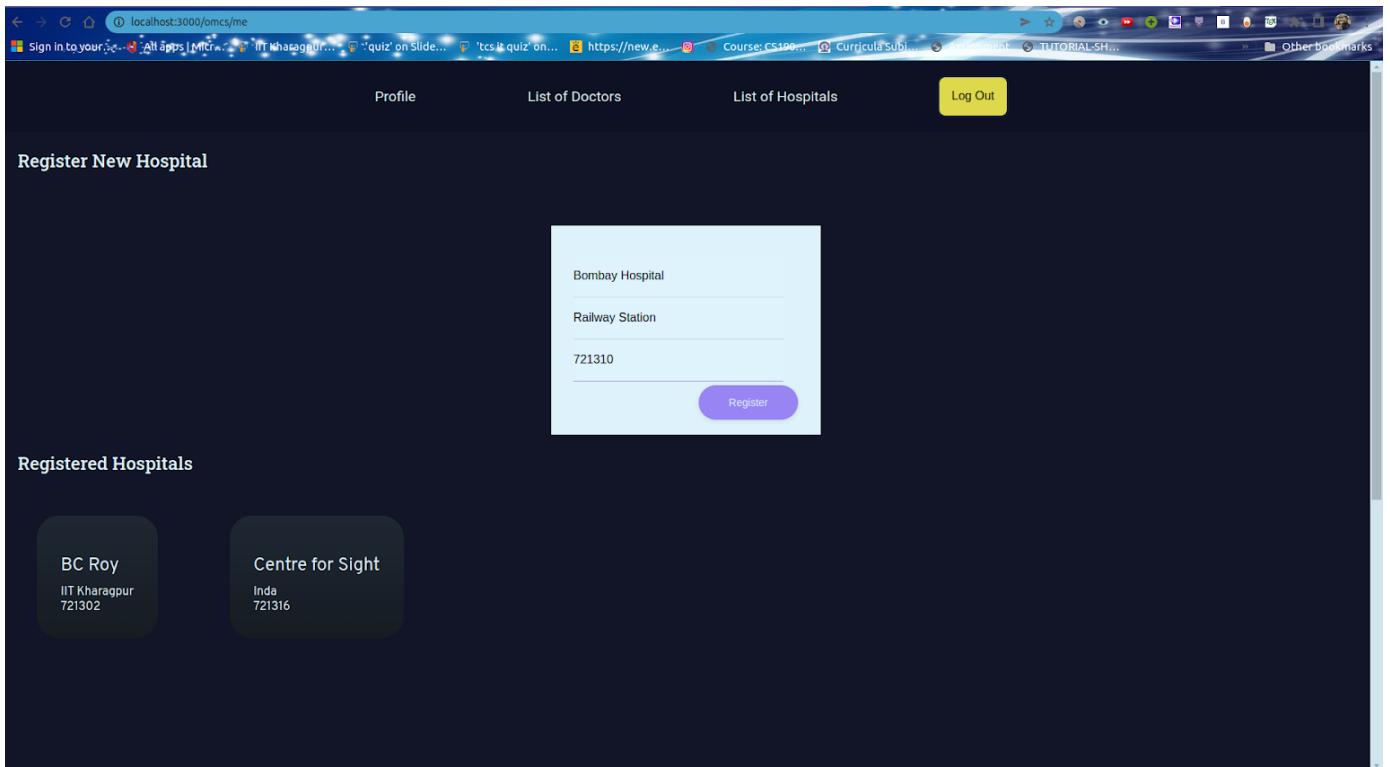
Centre for Sight
India
721316

Register

Registered Hospitals

BC Roy
IIT Kharagpur
721302

Hospital added to database



localhost:3000/omcs/me

Profile List of Doctors List of Hospitals Log Out

Register New Hospital

Bombay Hospital
Railway Station
721310

Register

Registered Hospitals

BC Roy
IIT Kharagpur
721302

Centre for Sight
India
721316

Testing Doctor and Patient Register : Test Case Passed

A screenshot of a web browser showing the 'Register' page for doctors. The URL in the address bar is `localhost:3000/omcs/register/doctor`. The page contains fields for email, password, name, address, phone number, hospital selection, and gender. A dropdown menu for 'Hospital' shows 'Centre for Sight' and 'BC Roy' as options, with 'Centre for Sight' currently selected. The 'Gynacologist' field is checked. The 'Gender' field shows '721302'. A purple 'Register' button is at the bottom right.

manish@gmail.com

Manish
De
Prem Bazaar
8305414601
Hospital: Centre for Sight
Gynacologist
721302

Register

Doctor Register

A screenshot of a web browser showing the 'Register' page for patients. The URL in the address bar is `localhost:3000/omcs/register/patient`. The page contains fields for email, password, name, address, phone number, and gender. The 'Gender' field shows '721307'. A purple 'Register' button is at the bottom right.

arjun@gmail.com

Arjun
Kamal
Gole Bazaar
9064888515
721307

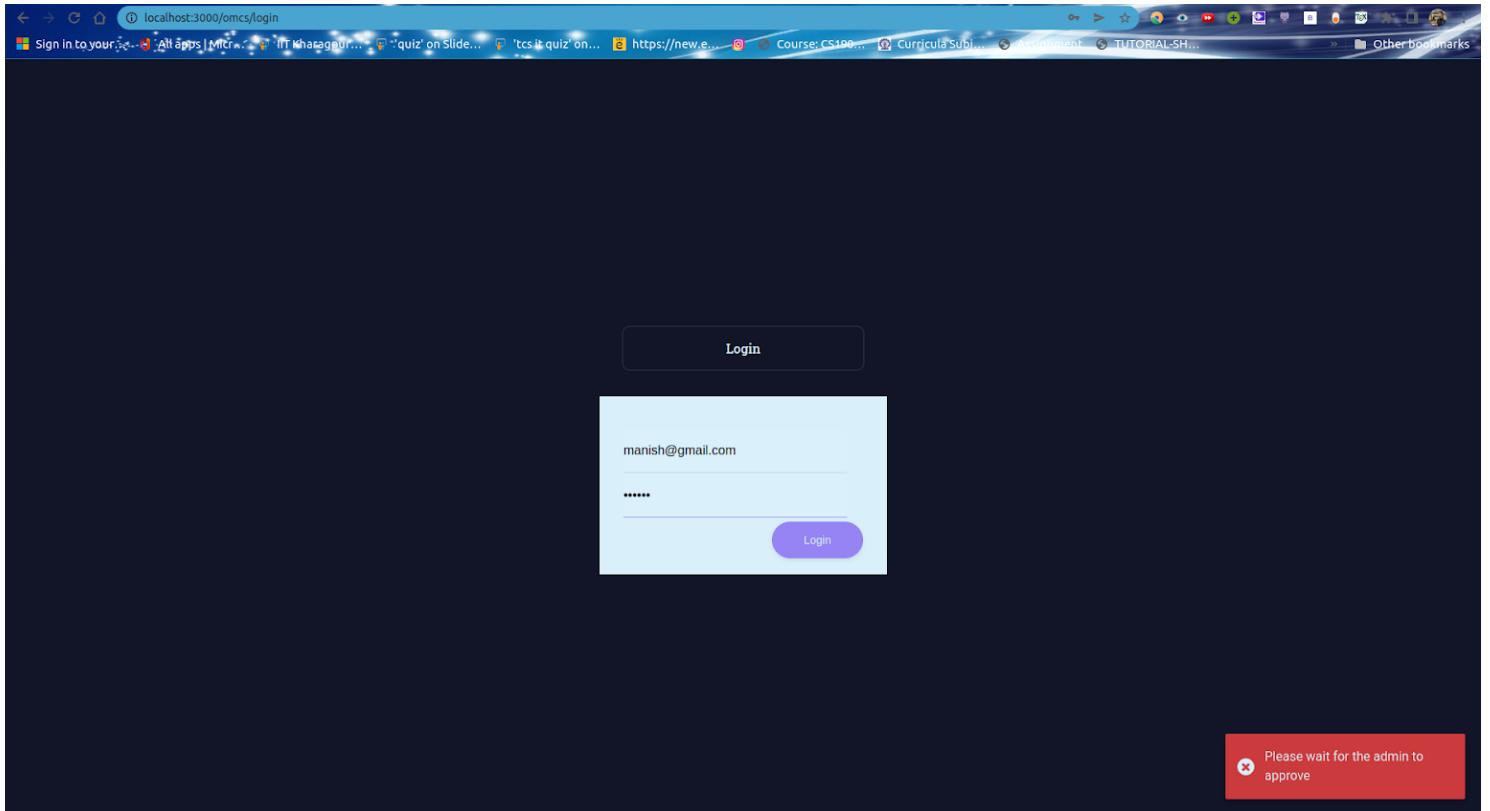
Register

Patient Register

Testing Doctor's Login:

Case I : After registering if the doctor tries to login without admin approval then it shows error.

Test Case Failed



Case II : After admin approves the Doctor, He/She can Login

Test Case Passed

(In the Admin's Dashboard)

The screenshot shows a web browser window with the URL `localhost:3000/omcs/me`. The page has a dark blue header with navigation links: Profile, List of Doctors, List of Hospitals, and Log Out. Below the header, there are two sections: "Pending Doctors" and "Enrolled Doctors". The "Pending Doctors" section contains a card for "Dr. Manish De" with the following details:
Hospital: BC Roy, IIT Kharagpur
Specialty: Gynaecologist
Prem Bazaar
721302
8305414601
manish@gmail.com
A pink "ACCEPT" button is at the bottom of the card.

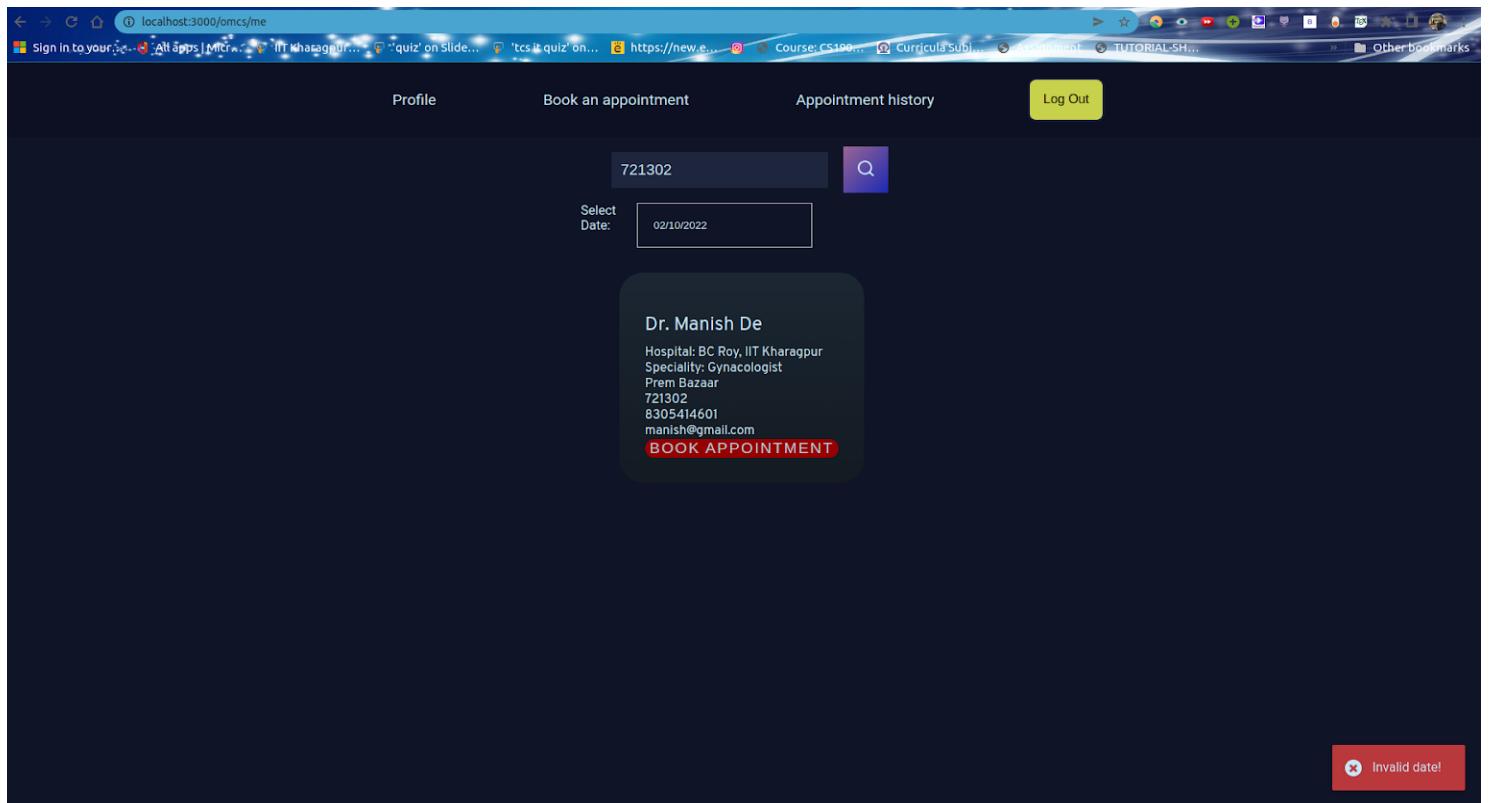
The screenshot shows the same web browser window after the doctor has been accepted. The "Pending Doctors" section is now empty. The "Enrolled Doctors" section contains a card for "Dr. Manish De" with the same details as before, but the "ACCEPT" button has been replaced by a pink "REMOVE" button.

(In the Doctor's Dashboard)

A screenshot of a web browser window displaying a doctor's dashboard. The URL in the address bar is `localhost:3000/omcs/me`. The page has a dark blue header with several tabs and links visible in the background. Below the header, there is a navigation bar with four items: "Profile", "Pending Appointments", "Appointment history", and a yellow "Log Out" button. A large, semi-transparent modal window is centered on the screen, titled "PROFILE". Inside the modal, it says "You are currently logged in as Dr. Manish De".

Testing Book Appointment :

Case I : Selecting a past date for appointment will cause error
Test Case Failed



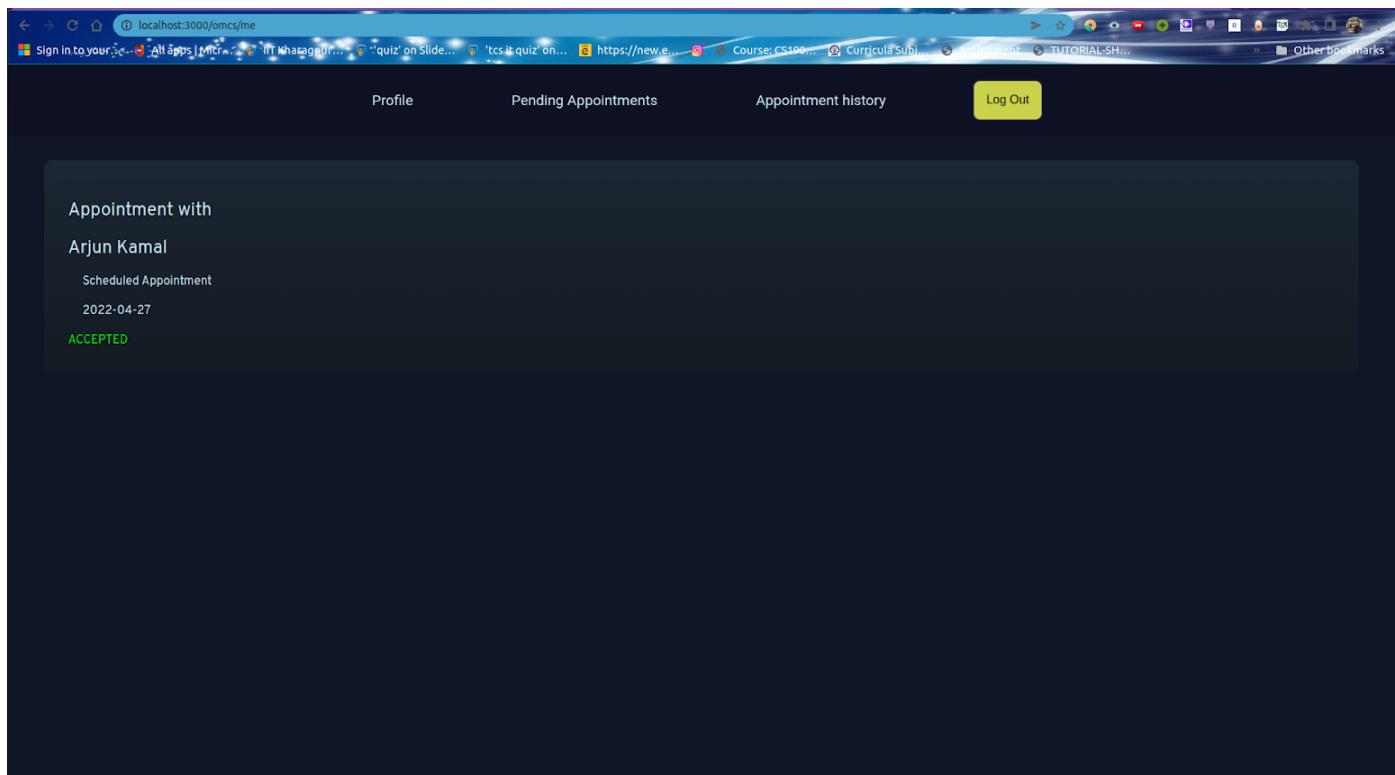
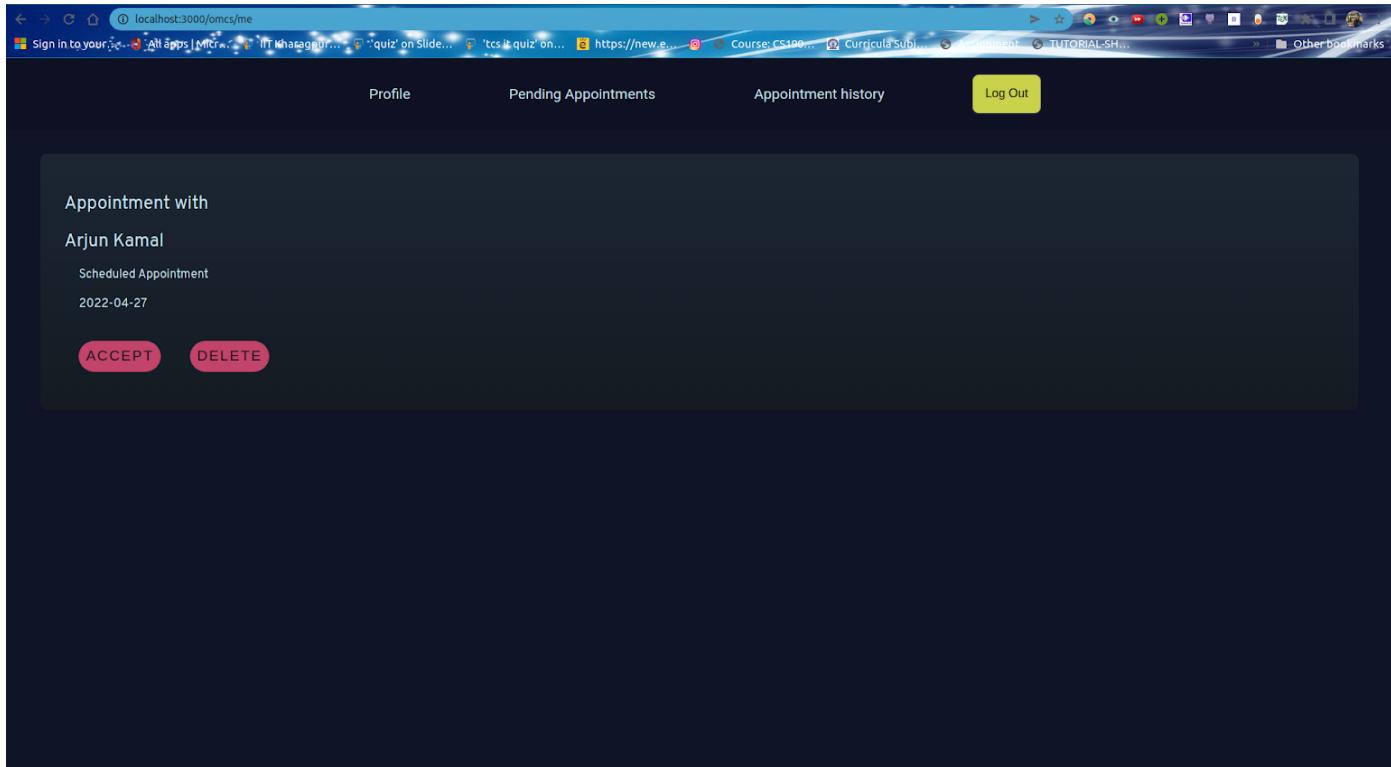
Case II : Selecting Proper date for Appointment Test Case Passed

(In Patient's DashBoard)

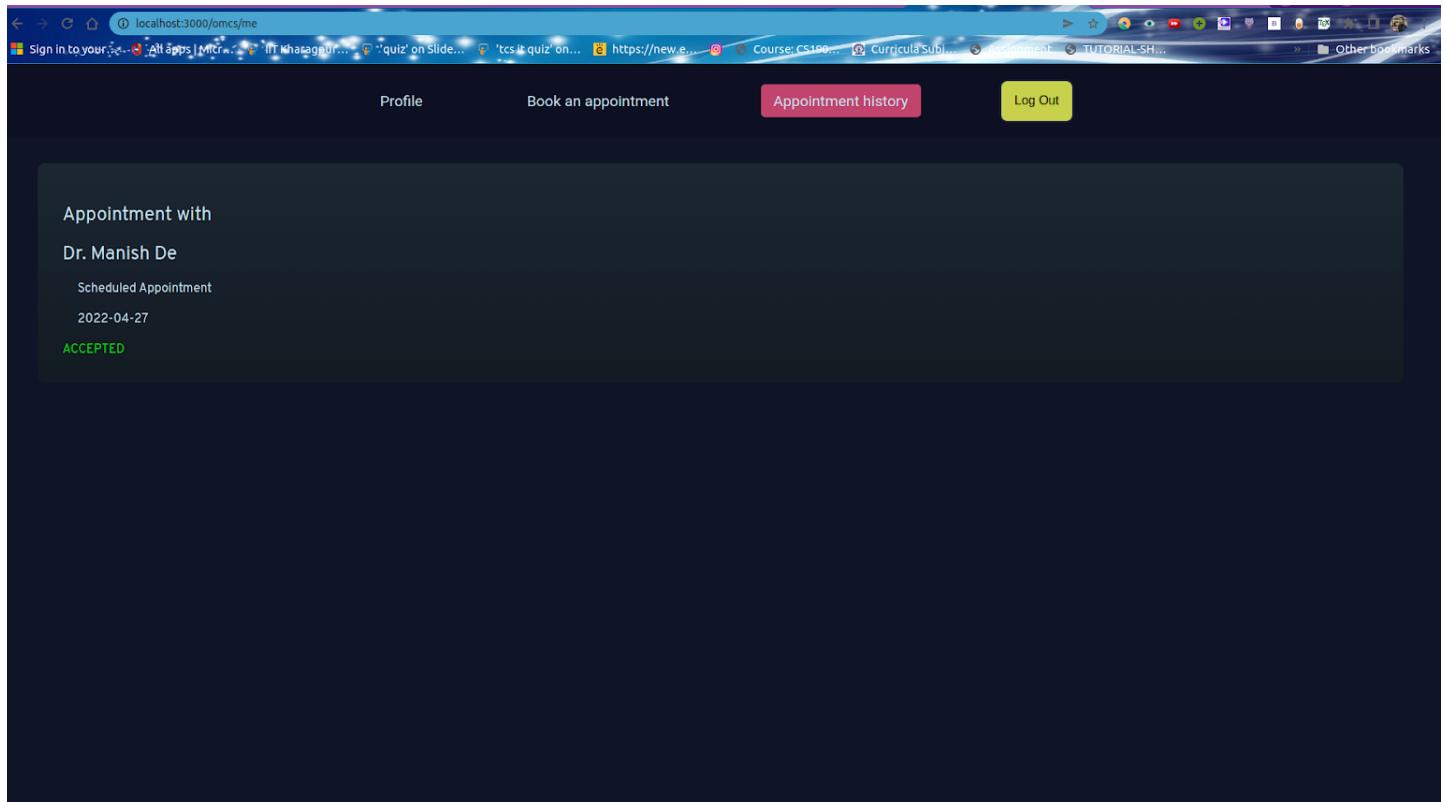
The screenshot shows a dark-themed web application interface for booking an appointment. At the top, there are navigation links: 'Profile', 'Book an appointment' (which is highlighted in blue), 'Appointment history', and 'Log Out'. Below these, a search bar contains the number '721302' and a magnifying glass icon. Underneath the search bar, there is a 'Select Date:' field containing the date '03/30/2022'. A large, semi-transparent callout box is centered on the page, displaying the following information about 'Dr. Manish De':
Hospital: BC Roy, IIT Kharagpur
Speciality: Gynaecologist
Prem Bazaar
721302
8305414601
manish@gmail.com
A pink 'BOOK APPOINTMENT' button is at the bottom of this callout.

The screenshot shows the same dark-themed web application interface, but now the 'Appointment history' link is highlighted in blue. The main content area displays a summary of a scheduled appointment:
Appointment with
Dr. Manish De
Scheduled Appointment
2022-04-27
PENDING

(In Doctor's Dashboard)



(In Patient's Dashboard upon acceptance)



We tried to add an email feature to the system which sends mail to the user's email ID from the Admin's email ID upon registering , booking appointment etc. but due to some error we couldn't implement it. Below attached is an image of the mail it sends to a doctor upon registering.

Welcome to OMCS! Inbox ×



sneakypotato.hello@gmail.com

to me ▾

7:52 PM (2 hours ago)

Hi Shiladitya De,

You've registered in the ASMR Online Medical COnsulation Service.

Since you've registered as a doctor, you will need to wait for your account to be activated by the admin before being allowed to use ther service to the fullest
Hope you have a nice time using our service

Thanks and regards

ASMR Team