

CAPSTONE PROJECT

ONLINE GYM MANAGEMENT SYSTEM

JAVA FULL STACK-REACT(BATCH-1)

Presented by

CHALUVADI SAI SUMANTH

Email: saisumanthchaluvadi@gmail.com

GitHub:- “ <https://github.com/SumanthChaluvadi/Sumanth-wipro-capstone-onlinegymmanagement.git> ”

ONLINEGYMMANAGEMENT:

The main aims of this project to develop a user-friendly web application that facilitates the Gym Management Software is a gym and health club membership management system.

Architecture:

Here's a detailed architecture for your gym management system, covering all classes, their attributes, relationships, and required functionalities. This architecture ensures that the system can manage users, roles, permissions, branches, gyms, trainers, members, payments, and appointments effectively.

Overview: -

The system is organized into several modules:

1. User Authentication and Registration
2. Gym Slot Scheduling Interface
3. Slot availability Management
4. Slot Booking Information Management
5. Feedback Management
6. Notification System
7. Admin Dashboard

TO create this project, I have used React and MySQL database, for crud operations post-man technologies and servers.

Backend: - (Server.js)

The server folder contains the backend code of my application. This part is responsible for handling data processing, business logic, and database interactions. It serves as the intermediary between the client and the database.

```
1 const express = require('express');
2 const mysql = require('mysql2');
3 const bodyParser = require('body-parser');
4 const cors = require('cors');
5 require('dotenv').config();
6
7 const jwt = require('jsonwebtoken'); // For token-based authentication
8
9
10
11 const app = express();
12 app.use(cors());
13 app.use(bodyParser.json());
14
15
16 const secretKey = 'asdfghjkl'; // Replace with a secure key for token signing
17
18
19
20 // Endpoint to handle admin login
21 app.post('/admin/login', (req, res) => {
22   const { username, password } = req.body;
23
24   // Replace with your actual authentication logic
25   if (username === 'admin' && password === 'password') {
26     // Generate a token
27     const token = jwt.sign({ username }, secretKey, { expiresIn: '1h' });
28     res.json({ success: true, token });
29   } else {
30     res.status(401).json({ success: false, error: 'Invalid credentials' });
31   }
32 });
33
34
35 // Middleware to verify token
36 app.use((req, res, next) => {
37   const token = req.headers.authorization?.split(' ')[1];
38 });
```

```

35 // Middleware to verify token
36 app.use((req, res, next) => {
37   const token = req.headers.authorization?.split(' ')[1];
38   if (token) {
39     jwt.verify(token, secretKey, (err, decoded) => {
40       if (err) {
41         return res.status(403).json({ error: 'Unauthorized' });
42       }
43       req.user = decoded;
44       next();
45     });
46   } else {
47     next();
48   }
49 });
50
51 // Example protected route
52 app.get('/admin/protected', (req, res) => {
53   if (req.user) {
54     res.json({ message: 'Protected data', user: req.user });
55   } else {
56     res.status(401).json({ error: 'Unauthorized' });
57   }
58 });
59
60 // MySQL connection
61 const db = mysql.createConnection({
62   host: process.env.DB_HOST,
63   user: process.env.DB_USER,
64   password: process.env.DB_PASSWORD,
65   database: process.env.DB_NAME,
66 });
67
68 db.connect((err) => {
69   if (err) throw err;
70   console.log('MySQL connected...');
71 });
72
73 // Routes
74 app.use('/api/register', require('./routes/registerRoutes'));
75 app.use('/api/login', require('./routes/loginRoutes'));
76 app.use('/api/bookings', require('./routes/slotsRoutes'));
77 app.use('/api/trainers', require('./routes/trainerRoutes'));
78 app.use('/api/user-trainers', require('./routes/userTrainerRoutes'));
79 app.use('/api/memberships', require('./routes/membershipRoutes'));
80 app.use('/api/payments', require('./routes/paymentRoute'));
81 app.use('/api/admin', require('./routes/AdminRoutes'));
82
83
84
85
86 app.use((err, req, res, next) => {
87   console.error(err.stack);
88   res.status(500).json({ message: 'Internal Server Error' });
89 });
90
91
92
93 app.listen(process.env.PORT, () => {
94   console.log(`Server running on port ${process.env.PORT}`);
95 });
96
97
98
99

```

Db.js (to connect MySQL database): -
const MySQL = require('mysql2');

```
1 DB_HOST=localhost
2 DB_USER=root
3 DB_PASSWORD=Password
4 DB_NAME=gymdb
5 PORT=5000
6 JWT_SECRET=Sumanth
```

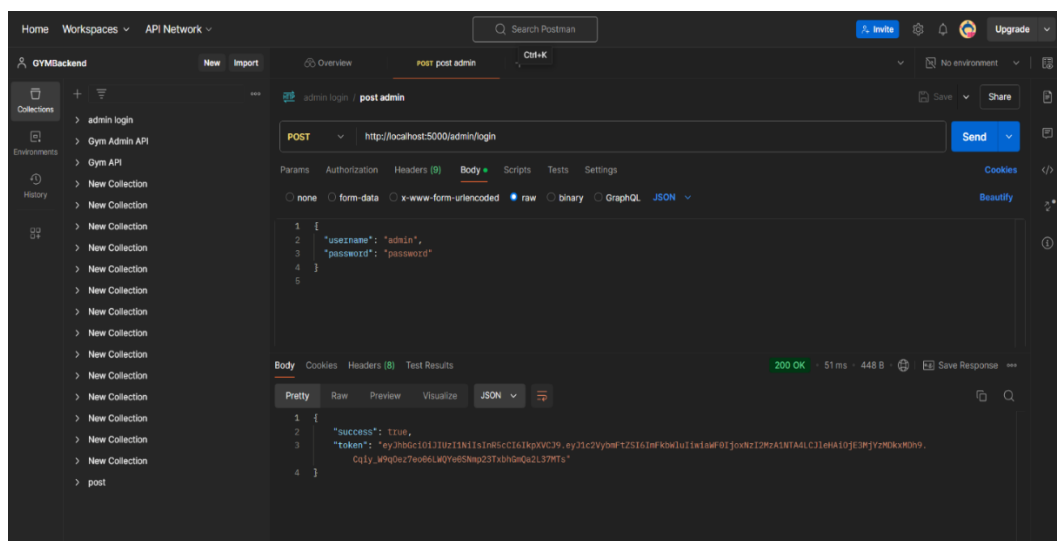
Routes used: -

1. Registerroutes.js
2. Loginroutes.js
3. Adminroutes.js
4. Trainerroutes.js
5. Slotroutes.js
6. Membershiproutes.js

Backend: - (Using postman)

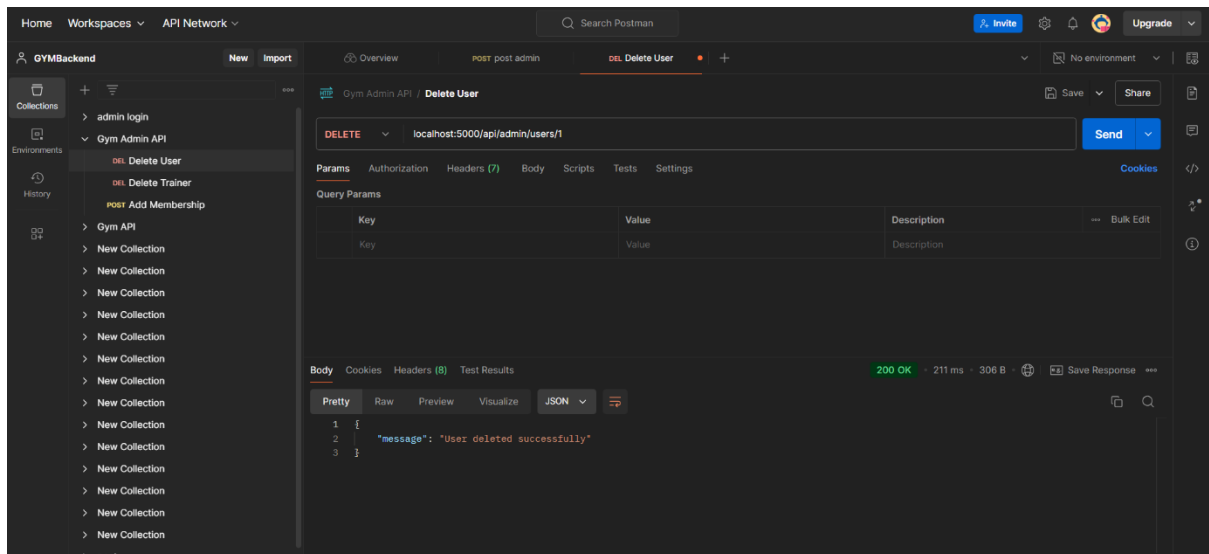
1.1 ADMIN LOGIN:

1.1.1 Method: - **POST** (<http://localhost:5000/admin/login>)



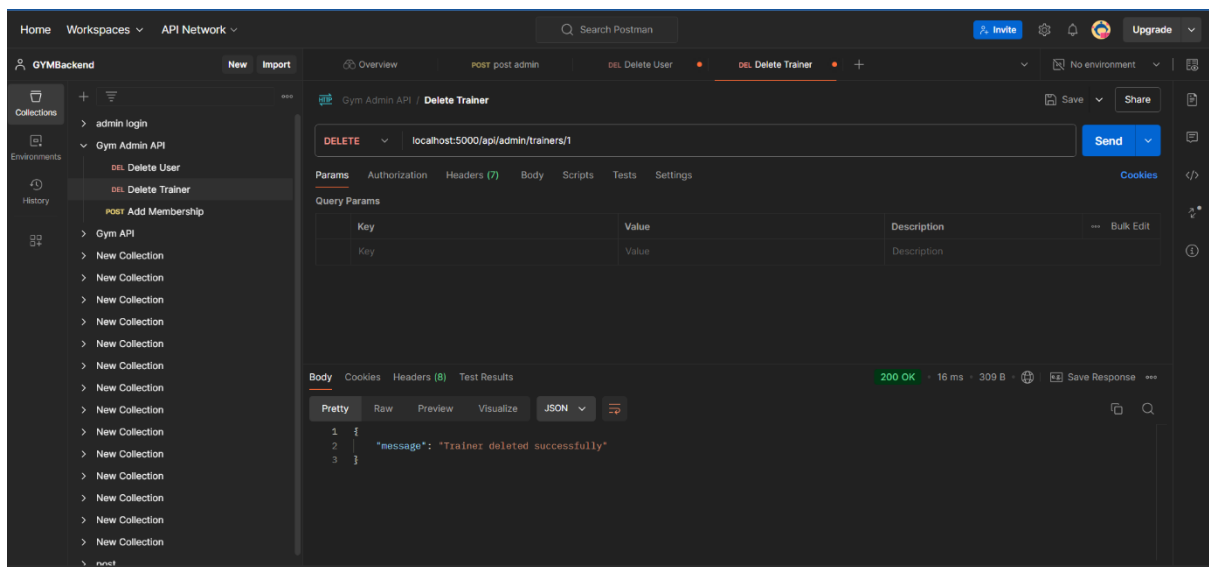
1.1.2 Delete user

Method: - **DELETE** (localhost:5000/api/admin/users/1)



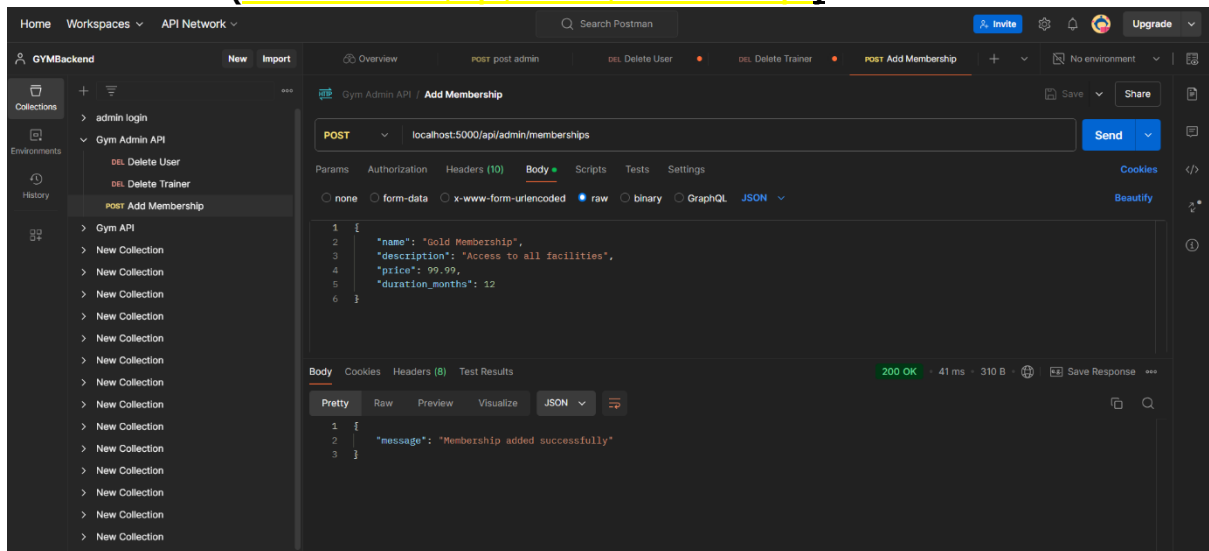
1.1.3 Delete user

Method: - **Delete** (localhost:5000/api/admin/trainers/1)



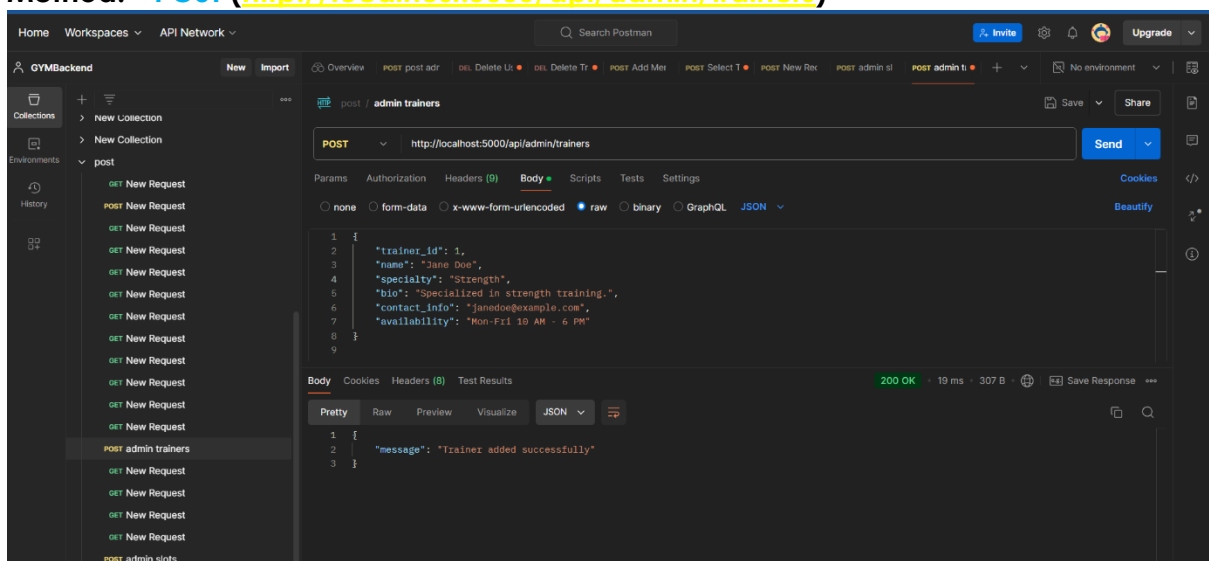
1.1.4 Post Member

Method: - **POST** (localhost:5000/api/admin/memberships)



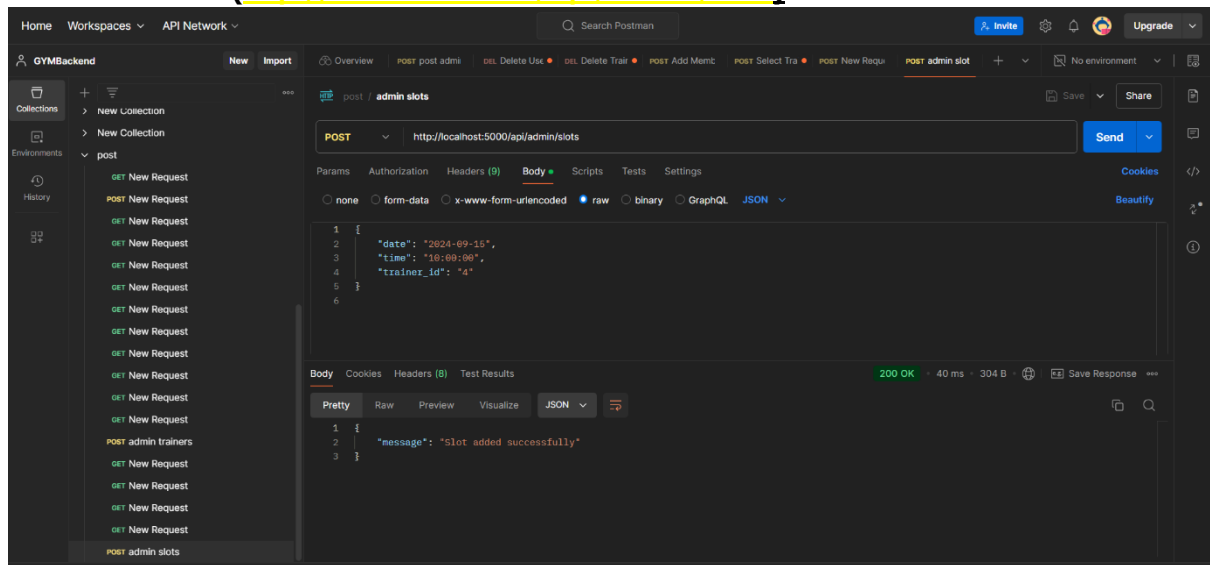
1.1.5 Post trainer

Method: - **POST** (<http://localhost:5000/api/admin/trainers>)



1.1.6 Post Slot

Method: - **POST** (<http://localhost:5000/api/admin/slots>)



MySQL

1.ADMIN TABLE: -

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

memberships membershipsplans bookings admins admins admins trainersnew trainersnew slots admins admins

1 • SELECT * FROM gymdb.admins;

Result Grid

	id	username	password
1	admin	\$2b\$10\$au6MnL89kUuQhsz9ob3UvUkiekWDUw...	

admins 1 x

Output

#	Time	Action	Message	Duration / Fetch
1	15:33:36	SELECT * FROM gymdb.admins LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
2	15:33:44	SELECT * FROM gymdb.admins LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

No object selected

2.MEMBERSHIP TABLE: -

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

trainersnew trainersnew slots admins admins bookings classes classsschedules login membershipsplans memberships

1 • SELECT * FROM gymdb.memberships;

Result Grid

	id	name	description	price	duration_months	created_at	updated_at
43		Gold Membership	Access to ...	50.00	12	2024-09-10 20:25:56	2024-09-10 20:25:56
44		Gold Membership	Access to ...	99.99	12	2024-09-10 22:18:11	2024-09-10 22:18:11
45		silver Members...	Access to ...	99.99	12	2024-09-10 22:18:24	2024-09-10 22:18:24
46		my Membership	Access to ...	99.99	12	2024-09-10 22:18:31	2024-09-10 22:18:31
47		yours Member...	Access to ...	99.99	12	2024-09-11 16:54:38	2024-09-11 16:54:38
48		Pranav Kulkarni	frv	500.00	4	2024-09-11 16:54:39	2024-09-11 16:54:39
49		Pranav Kulkarni	frv	500.00	4	2024-09-11 16:54:40	2024-09-11 16:54:40
50		Pranav Kulkarni	frv	500.00	4	2024-09-11 16:54:40	2024-09-11 16:54:40
51		Pranav Kulkarni	frv	500.00	4	2024-09-12 21:54:11	2024-09-12 21:54:11
52		Pranav Kulkarni	bnn	500.00	4	2024-09-12 22:20:40	2024-09-12 22:20:40
53		Pranav Kulkarni	ngn	500.00	4	2024-09-12 23:07:25	2024-09-12 23:07:25
54		Gold Membership	Access to ...	99.99	12	2024-09-13 00:40:22	2024-09-13 00:40:22
55		my	ffin	400.00	3	2024-09-13 10:25:29	2024-09-13 10:25:29
56		Gold Membership	Access to ...	99.99	12	2024-09-14 14:50:49	2024-09-14 14:50:49
57		Gold Membership	Access to ...	99.99	12		

memberships 1 x

Output

No object selected

3.POST TRAINERSSLOT TABLE: -

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

gymdb

Tables

admins

bookings

classes

classsschedules

login

membershipplans

memberships

payment_methods

payments

posttrainers

roles

slots

trainers

trainersnew

user_trainers

users

Administration Schemas

Information

No object selected

slots admins admins bookings classes classsschedules login membershipplans memberships posttrainers posttrainers

Limit to 1000 rows

1 • SELECT * FROM gymdb.posttrainers;

Result Grid

Filter Rows:

Export/Import:

Wrap Cell Content:

	id	name	image_url	skills
1	John Doe	http://example.com/image1.jpg	Weightlifting, Cardio	
2	Jane Smith	http://example.com/image2.jpg	Yoga, Pilates	
3	John Doe	http://example.com/image1.jpg	Weightlifting, Cardio	
4	Jane Smith	http://example.com/image2.jpg	Yoga, Pilates	

posttrainers 1 x

Apply Revert

Output

Action Output

4.SLOT TABLE: -

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

gymdb

Tables

admins

bookings

classes

classsschedules

login

membershipplans

memberships

payment_methods

payments

posttrainers

roles

slots

trainers

trainersnew

user_trainers

users

Administration Schemas

Information

No object selected

admins bookings classes classsschedules login membershipplans memberships posttrainers posttrainers slots

1

SELECT * FROM gymdb.slots;

Result Grid

Filter Rows:

Edit

Export/Import:

Wrap Cell Content:

	id	date	time	trainer_id
26	2024-09-15	10:00:00	4	
27	2024-09-15	10:00:00	4	
28	2024-09-13	00:00:01	1	
29	2024-09-14	00:00:02	2	
30	2024-09-15	00:00:03	3	
31	2024-09-15	10:00:00	4	
32	2024-09-13	00:00:01	1	
33	2024-09-15	10:00:00	4	
34	2024-09-15	10:00:00	4	
NULL	NULL	NULL	NULL	

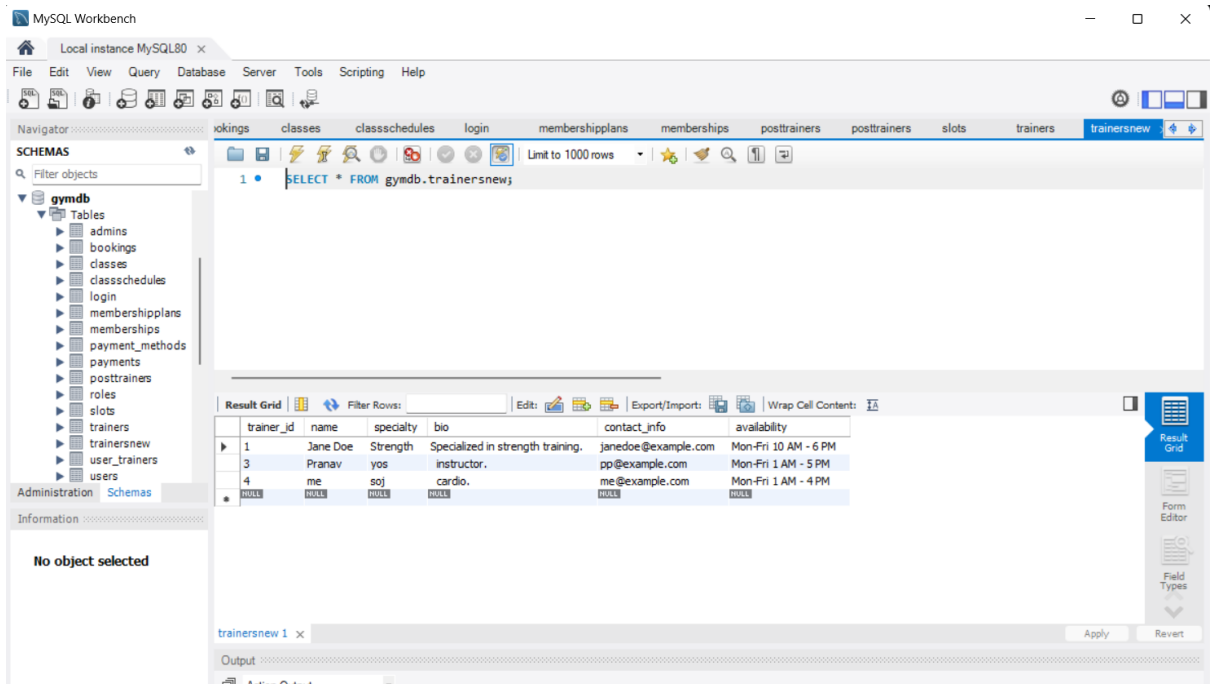
slots 1

Apply Revert

Output

Action Output

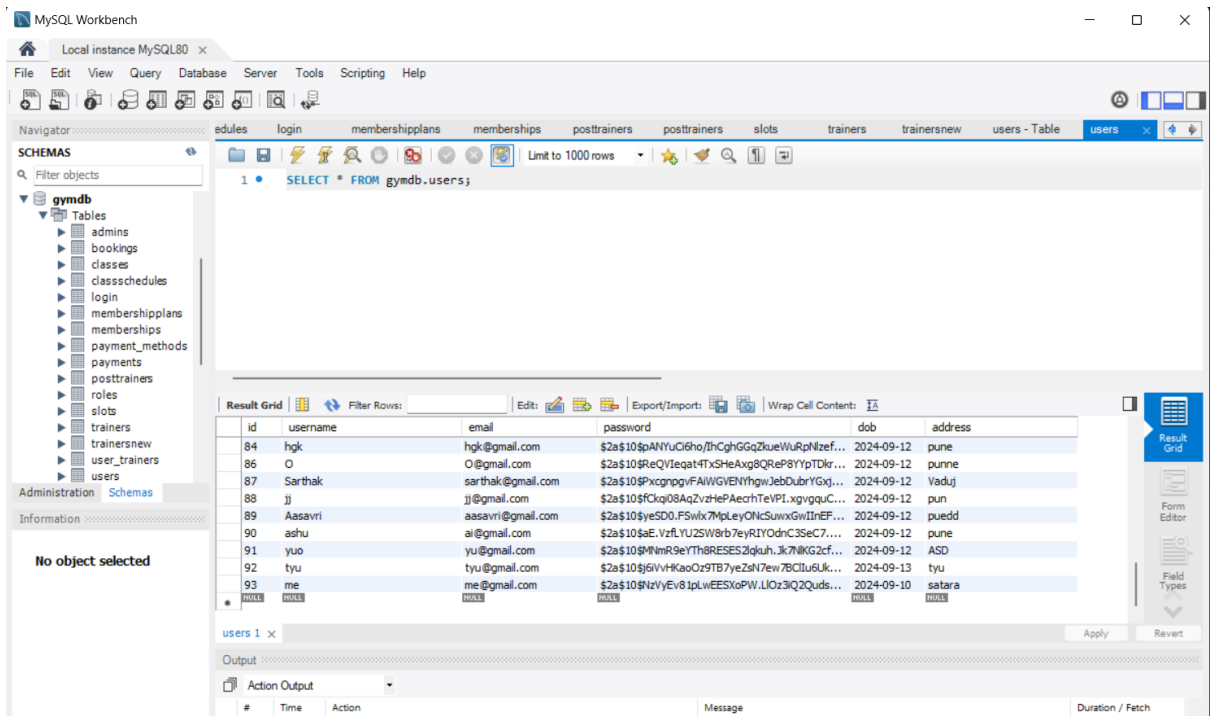
5. TRAINERS TABLE: -



MySQL Workbench interface showing the 'trainersnew' table in the 'gymdb' database. The table contains 4 rows of data. The query executed is `SELECT * FROM gymdb.trainersnew;`

trainer_id	name	specialty	bio	contact_info	availability
1	Jane Doe	Strength	Specialized in strength training.	janedoe@example.com	Mon-Fri 10 AM - 6 PM
3	Pranav	yos	instructor.	pp@example.com	Mon-Fri 1 AM - 5 PM
4	me	soj	cardio.	me@example.com	Mon-Fri 1 AM - 4 PM
NULL	NULL	NULL	NULL	NULL	NULL

6. USER TABLE: -



MySQL Workbench interface showing the 'users' table in the 'gymdb' database. The table contains 10 rows of data. The query executed is `SELECT * FROM gymdb.users;`

id	username	email	password	dob	address
84	hgk	hgk@gmail.com	\$2s\$10\$paNYuCi6no/ThCghGgZueWuRplNzef...	2024-09-12	pune
86	O	O@gmail.com	\$2s\$10\$ReQVieqat4TxSHeAxp8QReP8YpTDkr...	2024-09-12	punne
87	Sarthak	sarthak@gmail.com	\$2s\$10\$PxcgngpvFAIWGVENYhgWJebDubrYGxj...	2024-09-12	Vaduj
88	jj	jj@gmail.com	\$2s\$10\$Fckq08AqZvzHePaeChTeVPLxgvqqC...	2024-09-12	pun
89	Aasavri	aasavri@gmail.com	\$2s\$10\$yeSD0.FSwlx7MpleYQNCsuwxGwIInEF...	2024-09-12	puedd
90	ashu	ai@gmail.com	\$2s\$10\$eE.VzflYU2SW8rb7eyRIYDnCSec7...	2024-09-12	pune
91	yuo	yu@gmail.com	\$2s\$10\$MnmR9eYTh8RESES2qluh.Jk7NKG2cf...	2024-09-12	ASD
92	tyu	tyu@gmail.com	\$2s\$10\$6VvHkaoOz9TB7yeZsh7ew7BClu6Uk...	2024-09-13	tyu
93	me	me@gmail.com	\$2s\$10\$NzVvEv8IplwEESx0PW.LI0z3IQ2Quds...	2024-09-10	satara
NULL	NULL	NULL	NULL	NULL	NULL

OUTPUT FOR BACKEND: -

The output is the persistent storage of the data in the MySQL database. After I submit data through Swagger UI, it is successfully processed by My API and stored in MySQL, where it can be accessed, modified, or queried as needed.

FRONTEND (React.js)

- I Run the **npm start-react-app client** command to create a new React project
- **cd my-project** is used to navigate to project directory.

Src(components):

- 1.register.js
- 2.login.js
- 3.Admin.js
- 4.AdminLogin.js
- 5.BookSlot.js
- 6.Contact.js
- 7.gallery.js
- 8.ManageSlot.js
- 9.MembershipForm.js
- 10.TrainerList.js [TL.JS]

Main files: -

- 1.App.js
- 2.index.js
- 3.axiosConfig.js

```

1 import React, { useState } from 'react'; //returns the values
2
3 import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
4 import './App.css';
5
6 // Import images
7 import facebookIcon from './1384005.png';
8 import instagramIcon from './1384015.png';
9 import twitterIcon from './1384017.png';
10 import youtubeIcon from './1384012.png';
11 import linkedinIcon from './1384007.png';
12 import cp from './corporate-membership.jpg';
13 import pt from './personal-training.jpg';
14 import gp from './group-programs.jpg';
15
16
17
18 // Import components
19 import Register from './Register';
20 import Login from './Login';
21 import ManageSlots from './ManageSlots';
22 import BookSlot from './BookSlot';
23 import UserDashboard from './UserDashboard';
24 import MembershipForm from './MembershipForm';
25 import TL from './TL';
26 import PaymentForm from './PaymentForm';
27 import Gallery from './gallery';
28 import Contact from './Contact';
29 import Admin from './Admin';
30 import AdminLogin from './AdminLogin';
31
32
33
34
35 function App() {
36   const [clientSecret] = useState('');
37   const [membershipId] = useState('');

```

return (

<Router>

<div className="App">

<header className="App-header">

<div className="header-top">

<div className="contact-info">

Email: Titans@gmail.com ::

Phone: +919876543211

</div>

<div className="gym-name">

<h1>PEAK PERFORMANCE GYM</h1>

</div>

<div className="social-media">


```
<img src={facebookIcon} alt="Facebook" />
</a>
<a href="https://instagram.com" target="_blank" rel="noopener noreferrer">
  <img src={instagramIcon} alt="Instagram" />
</a>
<a href="https://twitter.com" target="_blank" rel="noopener noreferrer">
  <img src={twitterIcon} alt="Twitter" />
</a>
<a href="https://youtube.com" target="_blank" rel="noopener noreferrer">
  <img src={youtubeIcon} alt="YouTube" />
</a>
<a href="https://linkedin.com" target="_blank" rel="noopener noreferrer">
  <img src={linkedinIcon} alt="LinkedIn" />
</a>
</div>
</div>
<nav>
  <ul className="nav-list">
    <li><Link className="nav-link" to="/">Home</Link></li>
    <li><Link className="nav-link" to="/register">Register</Link></li>
    <li><Link className="nav-link" to="/login">Login</Link></li>
    <li><Link className="nav-link" to="/book-slot">Book Slot</Link></li>
    <li><Link className="nav-link" to="/TL">Trainer List</Link></li>
    <li><Link className="nav-link" to="/gallery">Gallery</Link></li>
    <li><Link className="nav-link" to="/contact">Contact Us</Link></li>
    <li><Link className="nav-link" to="/MembershipForm">MembershipForm</Link></li>
    <li><Link className="nav-link" to="/AdminLogin">AdminLogin</Link></li>
  </ul>
</nav>
</header>
<main>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/register" element={<Register />} />
    <Route path="/login" element={<Login />} />
```

```

    <Route path="/manage-slots" element={<ManageSlots />} />
    <Route path="/book-slot" element={<BookSlot />} />
    <Route path="/UD" element={<UserDashboard />} />
    <Route path="/MembershipForm" element={<MembershipForm />} />
    <Route path="/TL" element={<TL/>} />
    <Route path="/gallery" element={<Gallery/>} />
    <Route path="/Contact" element={<Contact/>} />
    <Route path="/Admin" element={<Admin/>} />
    <Route path="/AdminLogin" element={<AdminLogin />} />
  </Routes>

  {clientSecret && membershipId && (
    <PaymentForm clientSecret={clientSecret} membershipId={membershipId} />
  )}
</main>
</div>
</Router>
);
}

function Home() {
  return (
    <div className="Home">
      <div className="background-image">
        <div className="content-overlay">
          <h1>Welcome to Our Gym</h1>
          <p>Your journey to fitness starts here!</p>
        </div>
      </div>
      <div className="tabs">
      </div>
      <section className="unique-features">
        <h2>What Makes Us Different</h2>
        <p> State-of-the-Art Equipment

```

At Peak Performance Gym, we pride ourselves on offering the latest in fitness technology and equipment. Our state-of-the-art machines, free weights, and cardio equipment are meticulously maintained and regularly updated to ensure you have access to the best tools for achieving your fitness goals. Whether you're a seasoned athlete or just starting your fitness journey, our cutting-edge equipment provides the versatility and support you need to succeed.

Personalized Training Programs

We understand that every individual has unique fitness goals and needs. That's why our certified personal trainers work closely with you to design personalized training programs tailored specifically to your objectives. From weight loss and muscle building to sports-specific training and injury rehabilitation, our trainers use their expertise to create customized plans that maximize your results and keep you motivated.

Expert Guidance and Support

Our team of fitness professionals includes certified trainers, nutritionists, and wellness coaches who are dedicated to your success. We provide ongoing guidance and support to help you navigate your fitness journey with confidence. Our trainers offer one-on-one coaching, group classes, and workshops that cover various aspects of health and wellness, ensuring you have access to expert advice every step of the way.

Community and Motivation

At Peak Performance Gym, we believe in the power of community to drive motivation and success. Our supportive environment fosters a sense of camaraderie among members, creating an encouraging space where you can connect with like-minded individuals. We regularly host fitness challenges, group classes, and social events that bring our members together, making your fitness journey not only effective but also enjoyable.

Comprehensive Wellness Services

We go beyond traditional gym services to offer a holistic approach to wellness. Our facility includes amenities such as a wellness center, massage therapy, and nutrition counseling. We provide a full spectrum of services designed to enhance your overall well-being, helping you achieve a balanced and healthy lifestyle both inside and outside the gym.

Flexible Membership Options

We recognize that everyone's lifestyle and fitness needs are different. That's why we offer a variety of membership options to suit your individual preferences and schedule. Whether you prefer a short-term commitment or a long-term plan, we have flexible memberships that provide access to our extensive facilities and services, making it easier for you to stay consistent and achieve your fitness goals.

</section>

<section className="legacy">

<h2>Our Legacy</h2>

Founded in 2000: Peak Performance Gym has established itself as a leader in fitness and wellness, becoming a trusted name in the community.

Building a Strong Community: We have fostered a supportive environment through various events and workshops, creating lasting relationships and a sense of belonging.

Pioneering Fitness Innovations: We continually integrate the latest equipment and innovative training programs, ensuring our members benefit from cutting-edge solutions.

Celebrating Achievements: We take pride in our members' success stories and transformative journeys, celebrating milestones that reflect the effectiveness of our programs.

Future Aspirations: We are dedicated to evolving and enhancing our facilities and services to inspire and support individuals towards a healthier and more fulfilling life.

</section>

<section className="programs">

<h2>Our Programs</h2>

<div className="program">

<h3>Corporate Membership</h3>

<p>Enhance your employee wellness with our corporate membership packages. Enjoy exclusive benefits and discounts for your company.</p>

</div>

<div className="program">

<h3>Personal Training</h3>

<p>Our certified personal trainers will work with you to create a customized fitness plan that helps you achieve your personal goals.</p>

</div>

<div className="program">

<h3>Group Programs</h3>

<p>Join our group classes to stay motivated and fit. We offer a variety of classes, including yoga, HIIT, and strength training.</p>

</div>

</section>

<section className="contact-form">

<h2>Get in Touch</h2>

<form>

<label>Name:</label>

<input type="text" name="name" placeholder='Enter name'required />

<label>Email:</label>

<input type="email" name="email" placeholder='Enter email'required />

<label>Phone:</label>

<input type="tel" name="phone"placeholder='Enter phone number' required />

<label>State:</label>

<input type="text" name="state" placeholder='Enter State'required />

```
<label>City:</label>
```

```
<input type="text" name="city" placeholder='enter City'required />
```

```
<label>Gym:</label>
```

```
<input type="text" name="gym" required placeholder='from which gym ' />
```

```
<label>Message:</label>
```

```
<textarea name="message" placeholder='Enter Your Massage'required></textarea>
```

```
<div className="captcha">
```

```
  <label>Prove you are not a robot:</label>
```

```
  { /* Replace with a real CAPTCHA implementation */ }
```

```
  <input type="checkbox" name="captcha" required />
```

```
</div>
```

```
<button type="submit">Submit</button>
```

```
</form>
```

```
</section>
```

```
<footer className="footer">
```

```
<div className="footer-content">
```

```
<div className="footer-logo">
```

```
</div>
```

```
<div className="footer-links">
```

```
<h2>Quick Links</h2>
```

```
<ul>
```

```
<li><a href="/">Home</a></li>
```

```
<li><a href="/register">Register</a></li>
```

```
<li><a href="/login">Login</a></li>
```

```
<li><a href="/manage-slots">Manage Slots</a></li>
```

```
<li><a href="/book-slot">Book Slot</a></li>
```

```
<li><a href="/UD">Trainer List</a></li>
```

```
<li><a href="/gallery">Gallery</a></li>
```

```
<li><a href="/contact">Contact Us</a></li>
```

```
</ul>
```

```
</div>
```

```
<div className="footer-contact">
```

```
<h2>Contact Us</h2>
```

```
<p>Email: info@Titansgym.com</p>
<p>Phone: +919876543211</p>
<p>Address: 123 Fitness Lane, Muscle City, USA</p>
</div>
<div className="footer-social">
  <h2>Follow Us</h2>
  <a href="https://facebook.com" target="_blank" rel="noopener noreferrer">
    
  </a>
  <a href="https://instagram.com" target="_blank" rel="noopener noreferrer">
    
  </a>
  <a href="https://twitter.com" target="_blank" rel="noopener noreferrer">
    
  </a>
  <a href="https://youtube.com" target="_blank" rel="noopener noreferrer">
    
  </a>
  <a href="https://linkedin.com" target="_blank" rel="noopener noreferrer">
    
  </a>
</div>
</div>
<div className="footer-bottom">
  <p>&copy; Titan's Gym. All rights reserved. </p>
</div>
</footer>

</div>
);
}

export default App;
```

AxioConf.js: -

```

1 import axios from 'axios';
2
3 const instance = axios.create({
4   baseURL: 'http://localhost:5000/api' // 'http://localhost:5000' only for admin
5   // Ensure this matches your backend base URL
6 });
7
8 // Add a request interceptor to include the token in the headers
9 instance.interceptors.request.use(
10   (config) => {
11     const token = localStorage.getItem('adminToken');
12     if (token) {
13       config.headers.Authorization = `Bearer ${token}`;
14     }
15     return config;
16   },
17   (error) => Promise.reject(error)
18 );
19
20 const instance: AxiosInstance
21
22 export default instance;

```

Admin.js

```

1 import React, { useState } from 'react';
2 import axios from './axiosConfig'; // Ensure this path is correct
3 import './Admin.css';
4
5 const Admin = () => {
6   const [userId, setUserId] = useState('');
7   const [slot, setSlot] = useState({ date: '', time: '', trainer_id: '' });
8   const [trainer, setTrainer] = useState({ trainer_id: '', name: '', specialty: '', bio: '', contact_info: '', availability: '' });
9   const [membership, setMembership] = useState({ name: '', description: '', price: '', duration_months: '' });
10  const [selectedTrainerId, setSelectedTrainerId] = useState('');
11  const [loading, setLoading] = useState(false);
12
13  const handleDeleteUser = async () => {
14    if (userId) {
15      alert('Please enter a User ID.');
```

Admin login.js

```
1 import React, { useState } from 'react';
2 import axios from '../axiosConfig'; // Ensure this path is correct
3 import './Admin.css';
4
5 const Admin = () => {
6   const [userId, setUserId] = useState('');
7   const [slot, setSlot] = useState({ date: '', time: '', trainer_id: '' });
8   const [trainer, setTrainer] = useState({ trainer_id: '', name: '', specialty: '', bio: '', contact_info: '', availability: '' });
9   const [membership, setMembership] = useState({ name: '', description: '', price: '', duration_months: '' });
10  const [selectedTrainerId, setSelectedTrainerId] = useState('');
11  const [loading, setLoading] = useState(false);
12
13  const handleDeleteUser = async () => {
14    if (!userId) {
15      alert('Please enter a User ID.');
```

```
    return;
16    }
17    try {
18      const response = await axios.delete(`/admin/users/${userId}`); // Ensure this matches your backend route
19      alert(response.data.message);
20    } catch (error) {
21      console.error('Error deleting user:', error.response?.data?.error || error.message);
22      alert('Failed to delete user: ' + (error.response?.data?.error || error.message));
23    }
24  };
25
26  const handleAddSlot = async () => {
27    if (!slot.date || !slot.time || !slot.trainer_id) {
28      alert('Please fill out all fields.');
```

```
    return;
29    }
30    setLoading(true);
31    try {
32      const response = await axios.post('/admin/slots', slot); // Note the URL should match your backend route
33      alert(response.data.message);
34    } catch (error) {
35      console.error('Error adding slot:', error.response?.data?.error || error.message);
36    }
37  };
38
39  const Admin = () => {
40    const handleAddSlot = async () => {
41      console.error('Error adding slot:', error.response?.data?.error || error.message);
42      alert('Failed to add slot: ' + (error.response?.data?.error || error.message));
43    } finally {
44      setLoading(false);
45    }
46  };
47
48  const handleAddTrainer = async () => {
49    if (!trainer.trainer_id || !trainer.name || !trainer.specialty) {
50      alert('Please fill out all fields.');
```

```
    return;
51    }
52    setLoading(true);
53    try {
54      const response = await axios.post('/admin/trainers', trainer); // Ensure this matches your backend route
55      alert(response.data.message);
56    } catch (error) {
57      console.error('Error adding trainer:', error.response?.data?.error || error.message);
58      alert('Failed to add trainer: ' + (error.response?.data?.error || error.message));
59    } finally {
60      setLoading(false);
61    }
62  };
63
64  const handleDeleteTrainer = async () => {
65    if (!selectedTrainerId) {
66      alert('Please enter a Trainer ID.');
```

```
    return;
67    }
68    try {
69      const response = await axios.delete(`/admin/trainers/${selectedTrainerId}`); // Ensure this matches your backend route
70      alert(response.data.message);
71    } catch (error) {
72      console.error('Error deleting trainer:', error.response?.data?.error || error.message);
73      alert('Failed to delete trainer: ' + (error.response?.data?.error || error.message));
74    }
75  };
76}
```

```

72     }
73   };
74
75   const handleAddMembership = async () => {
76     if (!membership.name || !membership.description || !membership.price || !membership.duration_months) {
77       alert('Please fill out all fields.');
```

```

78       return;
79     }
80     setloading(true);
81     try {
82       const response = await axios.post('/admin/memberships', membership); // Ensure this matches your backend route
83       alert(response.data.message);
84     } catch (error) {
85       console.error('Error adding membership:', error.response?.data?.error || error.message);
86       alert('Failed to add membership: ' + (error.response?.data?.error || error.message));
87     } finally {
88       setloading(false);
89     }
90   };
91
92   return (
93     <div className="admin-container">
94       <h1>Admin Dashboard</h1>
95
96       <section>
97         <h2>Delete User</h2>
98         <input
99           type="text"
100           placeholder="User ID"
101           value={userId}
102           onChange={(e) => setUserId(e.target.value)}
103         />
104         <button onClick={handleDeleteUser}>Delete User</button>
105       </section>

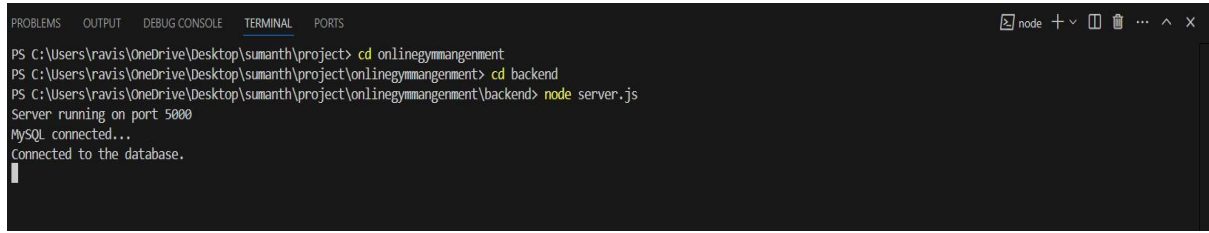
```

TERMINALS: -

BACKEND COMMANDS:

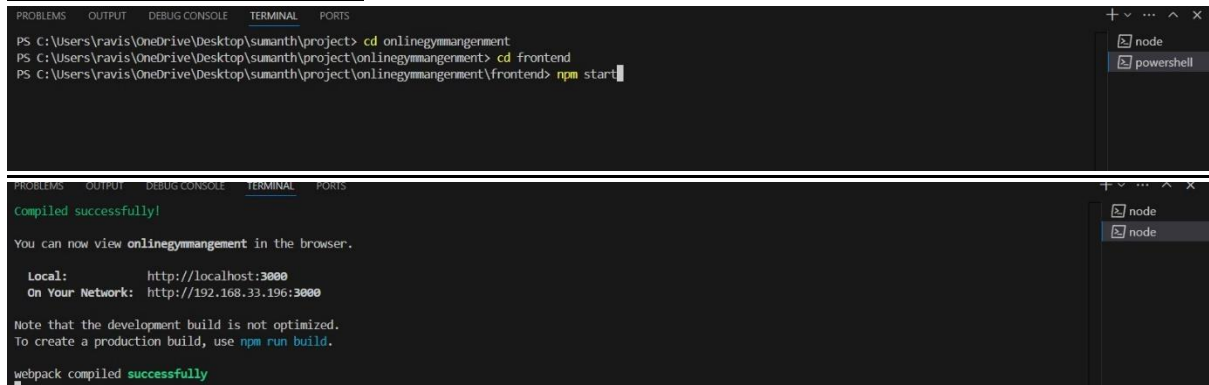


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project> cd onlinegymmanagement
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement> cd backend
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement\backend> node server.js
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project> cd onlinegymmanagement
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement> cd backend
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement\backend> node server.js
Server running on port 5000
MySQL connected...
Connected to the database.
```

FRONTEND COMMANDS:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project> cd onlinegymmanagement
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement> cd frontend
PS C:\Users\ravis\OneDrive\Desktop\sumanth\project\onlinegymmanagement\frontend> npm start

compiled successfully!

You can now view onlinegymmanagement in the browser.

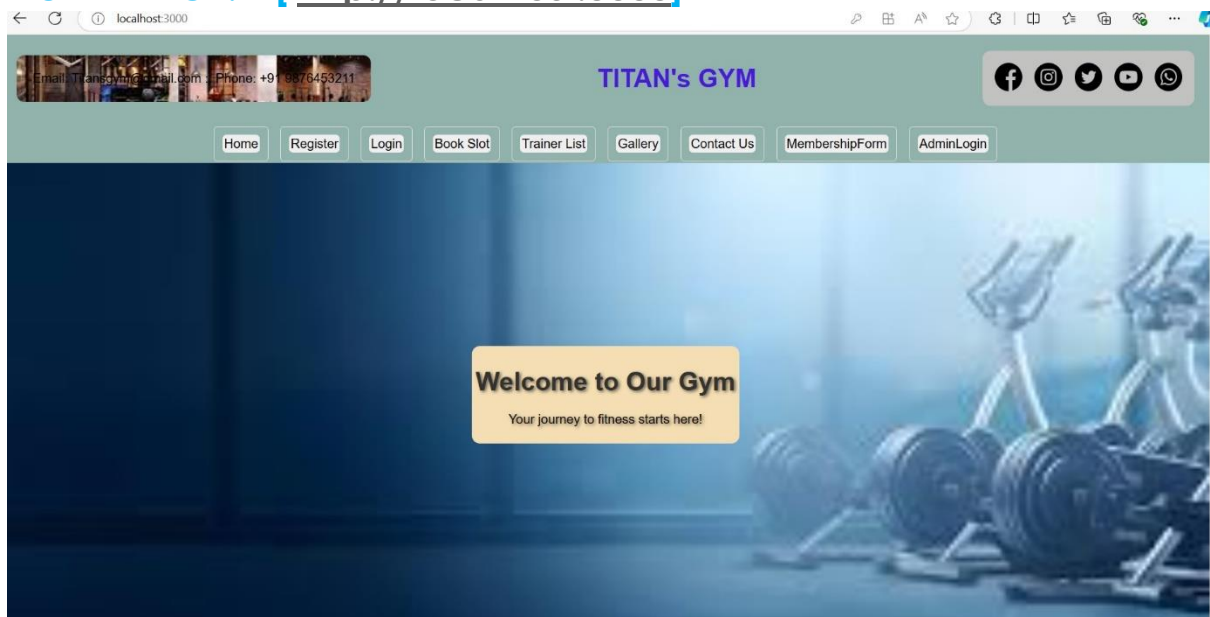
Local:      http://localhost:3000
On Your Network:  http://192.168.33.196:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

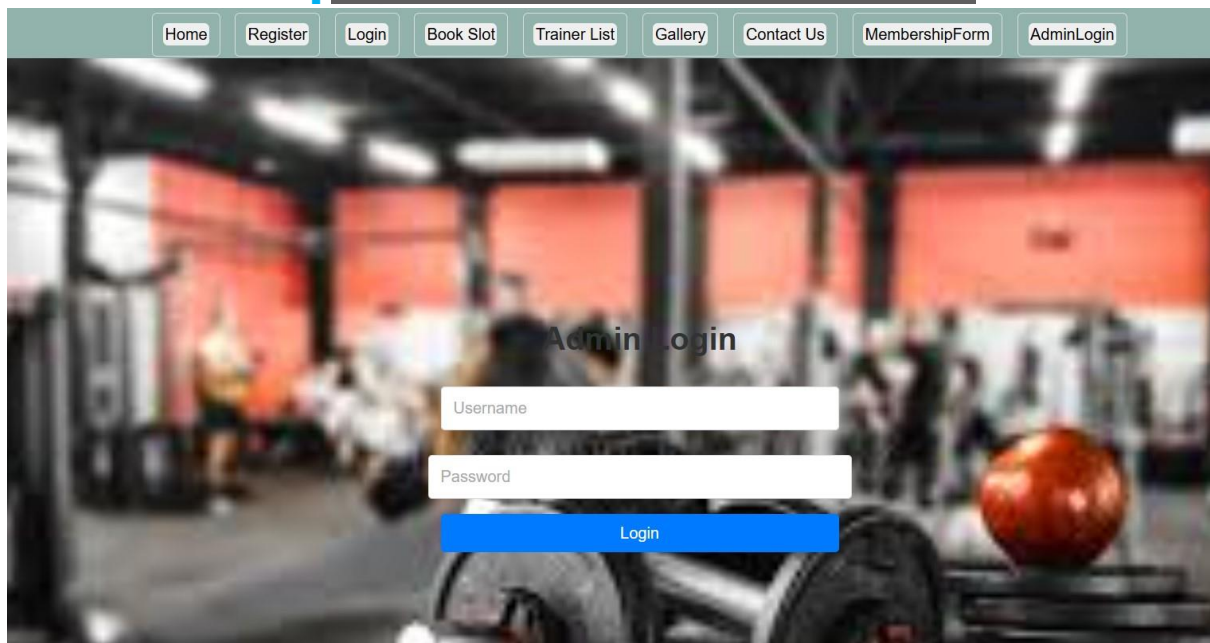
webpack compiled successfully
```


FRONTEND WEBSITE: -

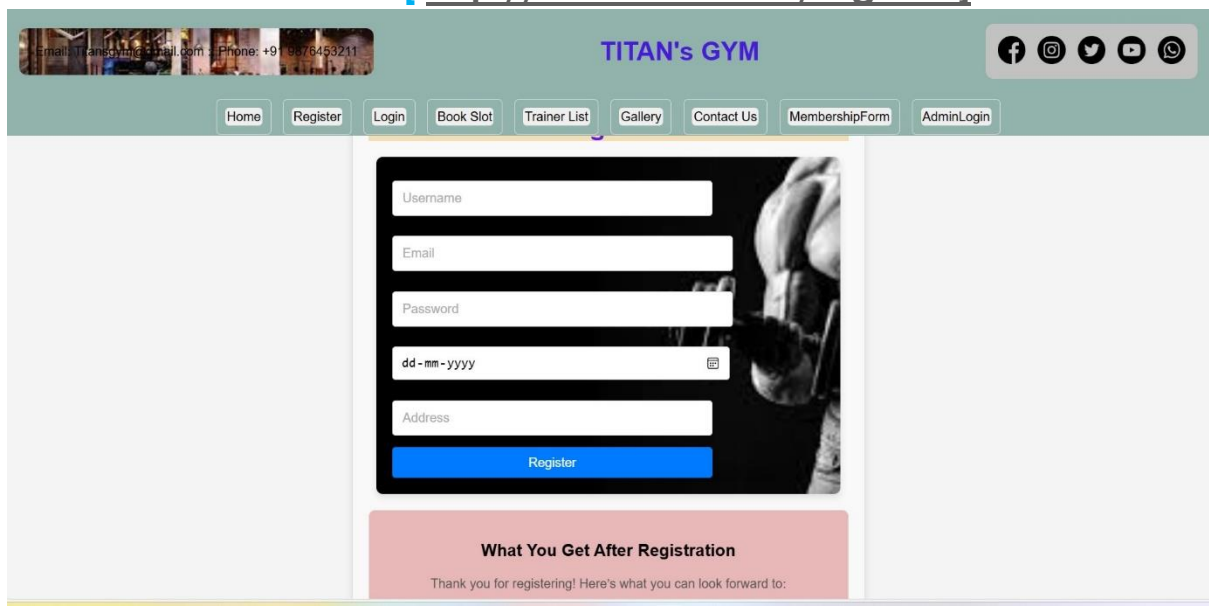
HOME PAGE: - [<http://localhost:3000>]



ADMIN LOGIN: - [<http://localhost:3000/ADMINLOGIN>]

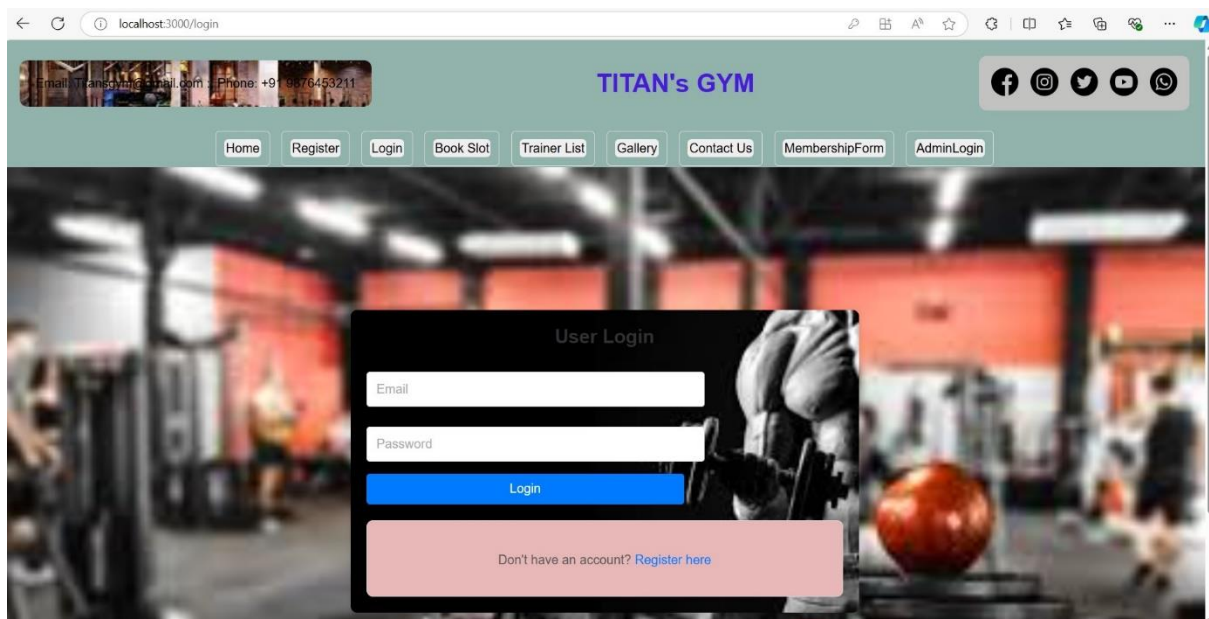


USER REGISTRATION: - [<http://localhost:3000/register>]



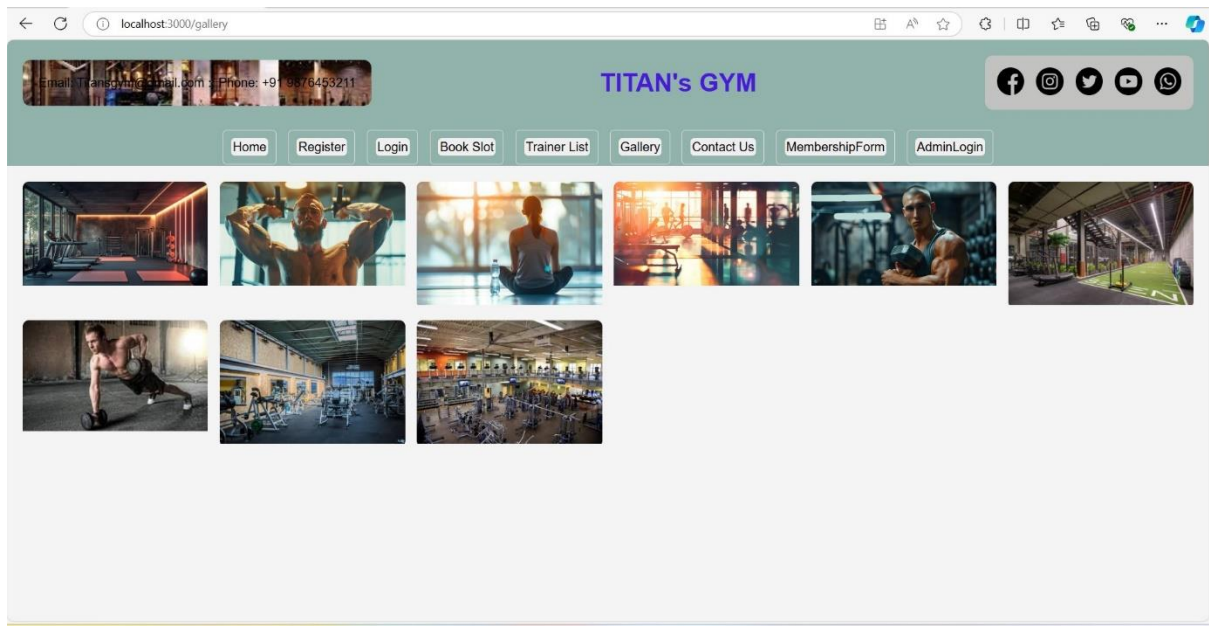
The screenshot shows the registration page for TITAN's GYM. The header includes the gym's name, contact information (Email: Transcymg@gmail.com, Phone: +91 8876453211), and social media icons. A navigation bar contains links for Home, Register, Login, Book Slot, Trainer List, Gallery, Contact Us, MembershipForm, and AdminLogin. The main content area features a registration form with fields for Username, Email, Password, a date picker (dd-mm-yyyy), and Address. A blue 'Register' button is at the bottom of the form. Below the form, a pink box titled 'What You Get After Registration' contains the text: 'Thank you for registering! Here's what you can look forward to:'.

USER LOGIN: - [<http://localhost:3000/login>]

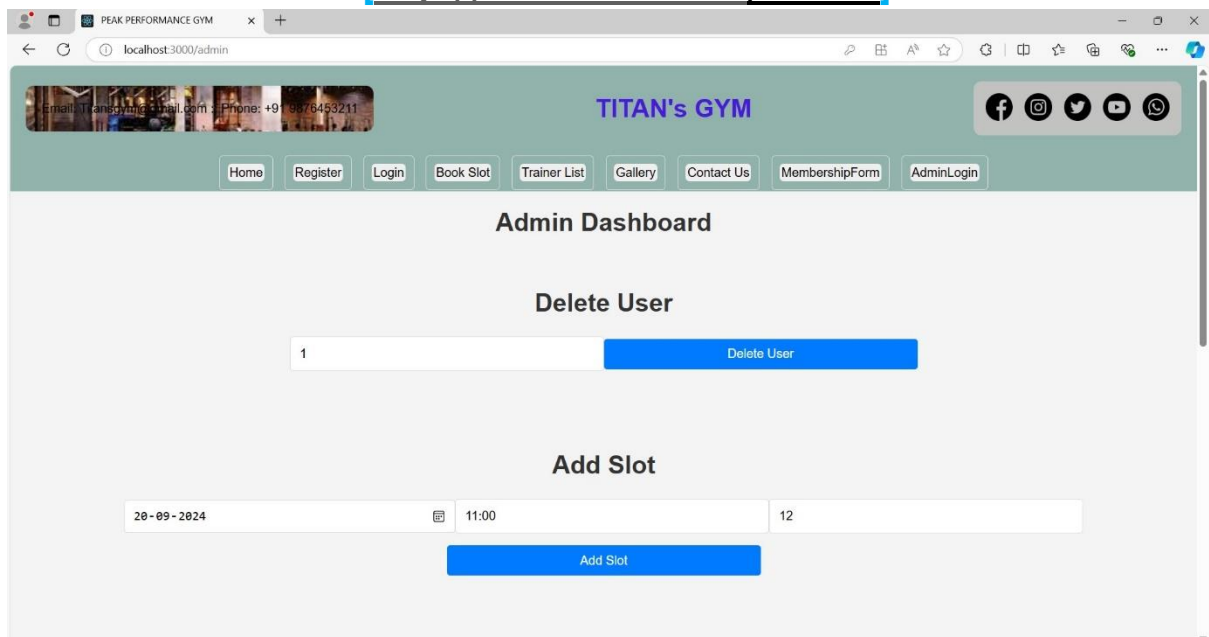


The screenshot shows the login page for TITAN's GYM. The header and navigation bar are identical to the registration page. The main content area features a login form titled 'User Login' with fields for Email and Password. A blue 'Login' button is at the bottom of the form. Below the form, a pink box contains the text: 'Don't have an account? [Register here](#)'.

GALLERY: - [<http://localhost:3000/gallery>]



ADMIN PROCESS: - [<http://localhost:3000/admin>]



The image displays two screenshots of a web application interface for gym management, running on localhost:3000/admin.

Top Screenshot:

- Add Slot:** A form with three input fields: a date field containing "20-09-2024", a time field containing "11:00", and a number field containing "12". Below these fields is a blue button labeled "Add Slot".
- Add Trainer:** A form with three input fields: a number field containing "12", a text field containing "ssc", and a text field containing "gymmanagment". Below these fields is a large text area containing "hdiabobuo". At the bottom, there is a number field containing "1241584", a text field containing "ervy", and a blue button labeled "Add Trainer".
- Delete Trainer:** A section header for the delete functionality.

Bottom Screenshot:

- Delete Trainer:** A form with a single input field labeled "Trainer ID" and a blue button labeled "Delete Trainer".
- Add Membership:** A form with three input fields: "Membership Name", "Description", and "Price". Below these fields is a text field labeled "Duration (months)" and a blue button labeled "Add Membership".

Here, is the overview of Frontend webpage that includes admin login, user login, trainer login, new registration, existed deleting and other modules includes notification.

Thanking you.,

Presented by: - Chaluvadi Saisumanth

Email: - saisumanthchaluvadi@gmail.com

GitHub: - "https://github.com/SumanthChaluvadi/Sumanth-wipro-capstone-onlinegymmanagement.git"

