

# Fighttrack.xyz

Empowering Boxing Gyms  
for Seamless Management

2024

|   |   |           |
|---|---|-----------|
| 12<br>CardioBox<br>7:00 pm AEST (45m)<br>Delete | 13  |           |
| 19  | 20<br>Fighters Academy<br>6:00 pm AEST (1h)<br>Delete | K<br>4:00 |
| 26  | 27  |           |

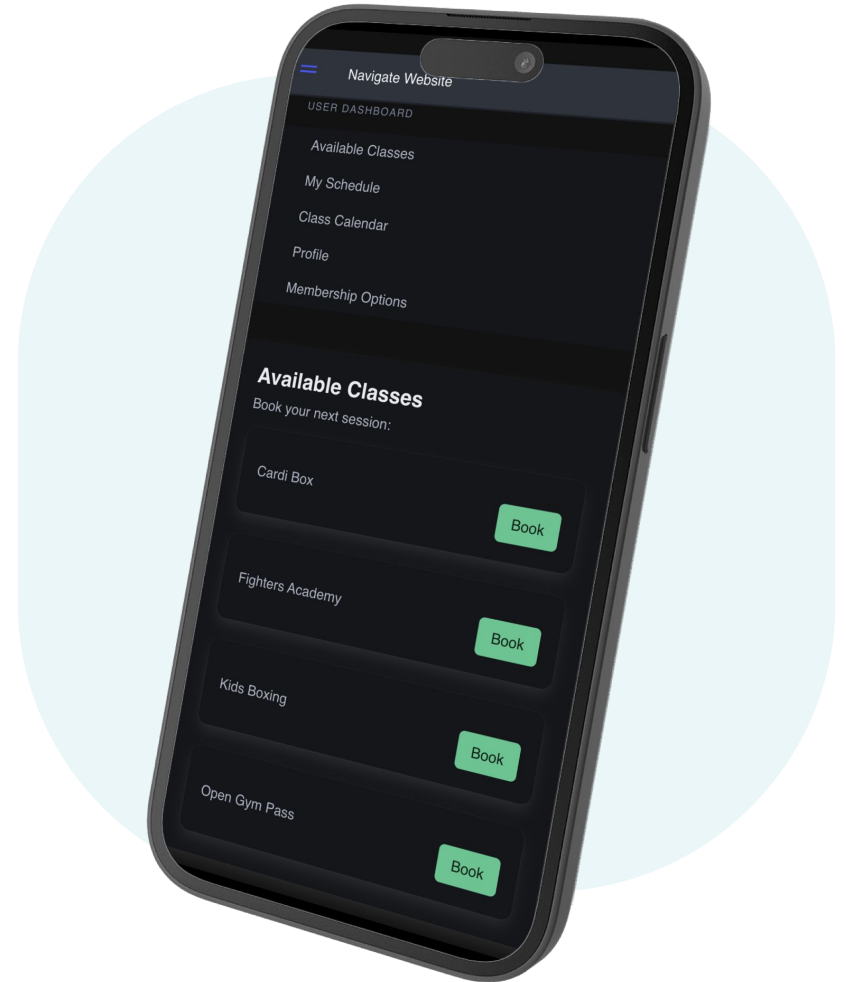
# Transforming Management for Boxing Gyms

The web application is designed to optimize the booking and management process for boxing gyms, providing an intuitive and efficient platform for clients and gym owners alike.

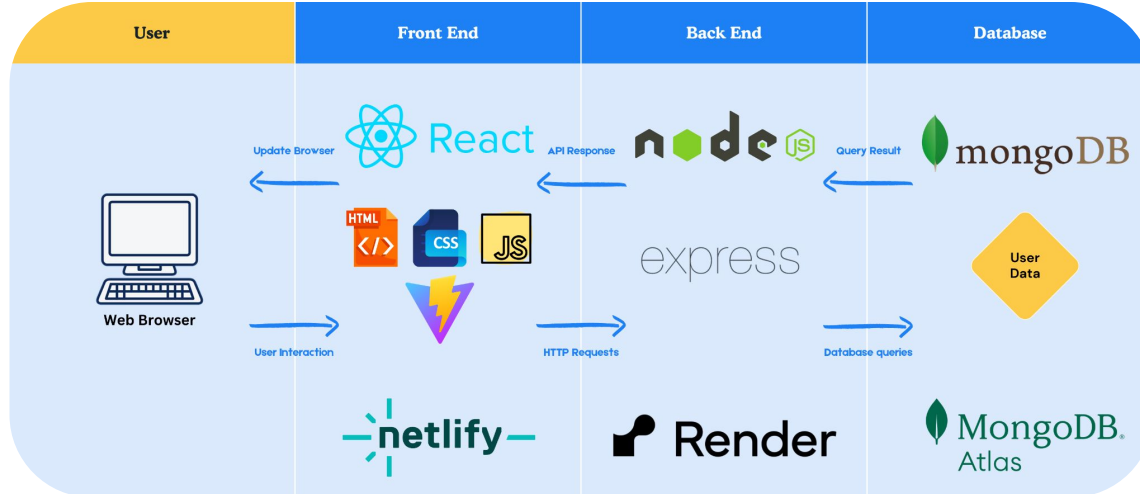
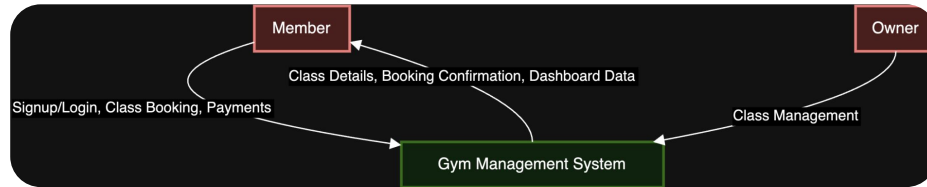
- Monitor class enrollments and track attendance.
- Manage and edit class details with ease.
- Comprehensive payment collection for various membership packages.
- Easy account creation and profile management.

Pain points being solved

**Scheduling, Management, Payments**



# Architecture



# User Stories



## Josh

### Boxing Instructor

Josh is a boxing instructor passionate about fitness and teaching others. He runs a local boxing gym. Josh is committed to providing the best experience for his clients but finds it challenging to manage all administrative tasks while focusing on training his clients.

#### Needs:

- Easily manage class schedules, enrollments, and payments.
- Monitor class attendance.

#### Frustrations:

- Struggles with balancing administrative tasks and training responsibilities between classes.
- Finds it a hassle to track payments and attendance manually.

#### 3x User Stories:

- As a user, I want to see how many people have enrolled in each class so that I can prepare the right resources and plan my sessions effectively.
- As a user, I want to track whether attendees have paid for classes so that I can manage payments efficiently without interrupting training sessions.



## Claire

### Boxing Class Member

Claire is a university student who stays active through boxing classes. With her busy study schedule, Claire needs a flexible and convenient way to book her boxing classes, track her progress, and manage her membership.

#### Needs:

- See when classes are available to plan her schedule.
- Manage her class enrollments easily.
- Search for classes available at specific times and days.
- Pay for her membership securely and conveniently.

#### Frustrations:

- Finds it hard to locate classes that fit into her busy schedule.
- Tends to forget when she has booked a class for.

#### 3x User Stories:

- As a user, I want to see when classes are available so that I can plan my schedule.
- As a user, I want to pay for my membership directly through the app so that I can handle all my gym-related tasks in one place.
- As a user, I want to track my progress so that I can handle all my gym-related tasks in one place.

# Wireframes

## "Name" BOXING CLUB

HEALTH & FITNESS CENTRE

Call us today on  
(PHONE NUMBER) to start getting fit!

| HOME | FACILITIES | MEMBER | LOCATION |
|------|------------|--------|----------|
|------|------------|--------|----------|

(Photo)

ARE YOU READY?  
Join us today and jumpstart your health!

JOIN US

7 DAYS  
FREE TRIAL  
Join Us Now

Let's Go >

(Photo)

Schedule & Classes >

(Photo)

Membership Packages >

### OPENING HOURS

Monday - Friday  
7.00am - 10.00pm

### ADDRESS

123 Address Street,  
Suburbs, QLD 1234  
email@domain.com  
Tel: 123-456-789

### LINKS

Social  
Media

Social  
Media

Social  
Media

Copyright 2024 by BOXING CLUB

## "Name" BOXING CLUB

HEALTH & FITNESS CENTRE

Call us today on  
(PHONE NUMBER) to start getting fit!

### FACILITIES

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(Photo)

#### FACILITY 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

#### FACILITY 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(Photo)

#### FACILITY 3 - CLASS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### OPENING HOURS

Monday - Friday  
7.00am - 10.00pm

### ADDRESS

123 Address Street,  
Suburbs, QLD 1234  
email@domain.com  
Tel: 123-456-789

### LINKS

Social  
Media

Social  
Media

Social  
Media

Copyright 2024 by BOXING CLUB

Logo

Menu

## "Name" BOXING CLUB

HEALTH & FITNESS CENTRE

Call us today on  
(Phone Number)  
Start getting fit!

(Photo)

ARE YOU READY?  
Join us today and jumpstart your health!

JOIN US

7 DAYS  
FREE TRIAL  
Join Us Now

Let's Go >

(Photo)

Schedule & Classes >

(Photo)

Membership Packages >

OPENING HOURS  
Monday - Friday  
7.00am - 10.00pm

### ADDRESS

123 Address Street,  
Suburb QLD 1234  
Tel: 123-456-789

Social  
Media

Social  
Media

Social  
Media

Copyright 2024 by BOXING CLUB

Logo

Menu

## "Name" BOXING CLUB

HEALTH & FITNESS CENTRE

Call us today on  
(Phone Number)  
Start getting fit!

### FACILITIES

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

#### FACILITY 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(Photo)

#### FACILITY 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(Photo)

#### FACILITY 3 - CLASS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(Photo)

Class & Schedule >

### OPENING HOURS

Monday - Friday  
7.00am - 10.00pm

### ADDRESS

123 Address Street,  
Suburb QLD 1234  
email@domain.com  
Tel: 123-456-789

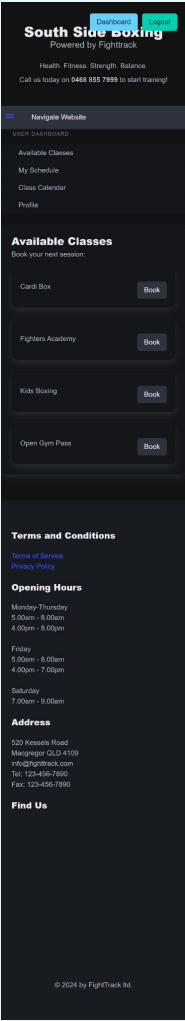
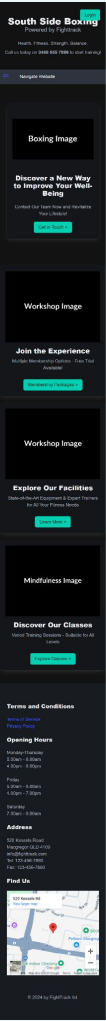
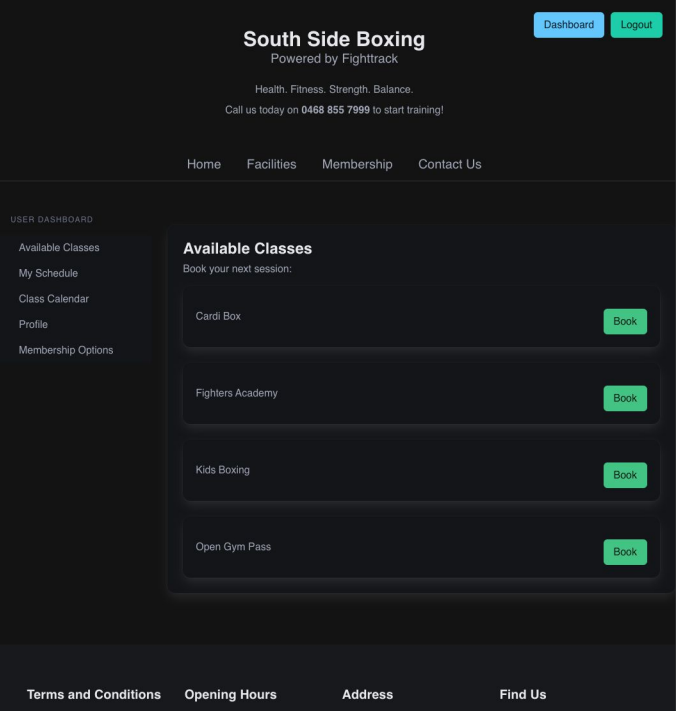
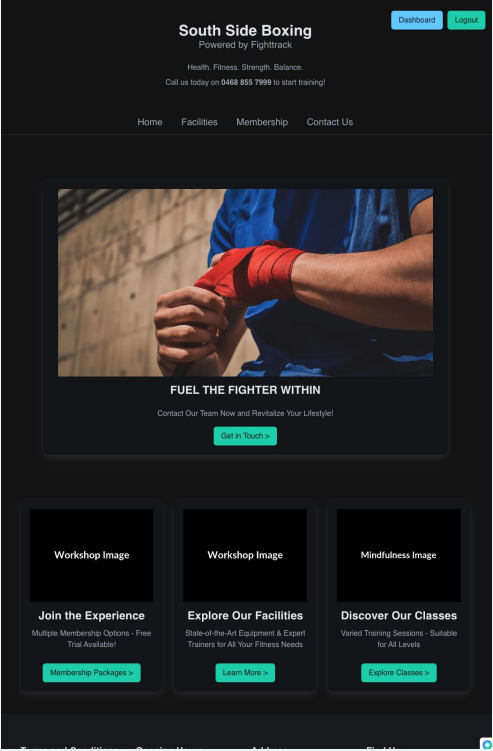
Social  
Media

Social  
Media

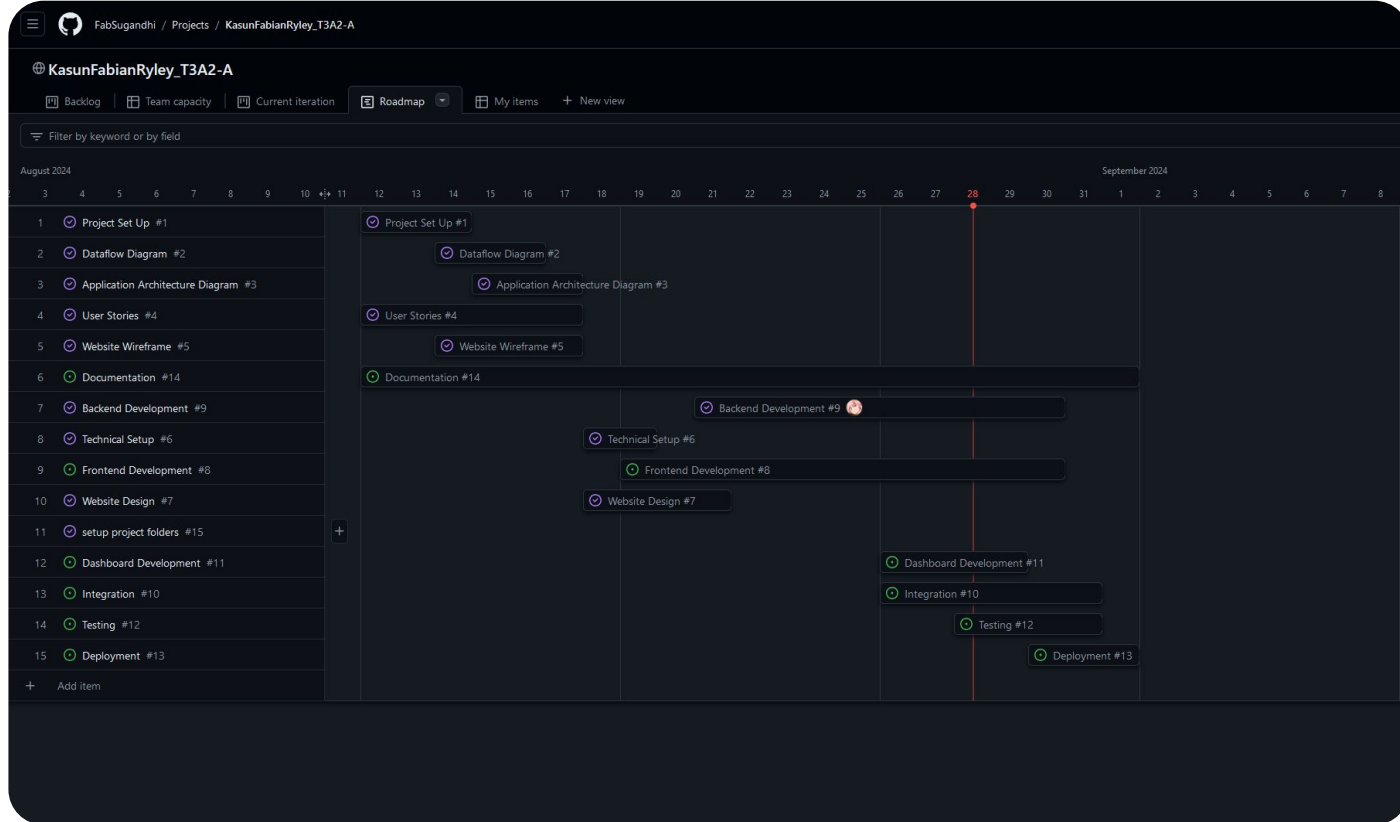
Social  
Media

Copyright 2024 by BOXING CLUB

# Screen Captures



# Project Management Kanban



# Integral pieces of code – Backend

## Verify Function

The verify function extracts the JWT token from the header and then verifies it with the `jwt.verify` method. If it is valid it identifies the logged in user using the `findById` method, then all the user information (excluding the password) is attached to the request object as `req.user`

This code is integral as it's our application's way of authenticating all of our private routes.

```
const verify = async (req, res, next) => {
  let token;

  if (req.headers.authorization && req.headers.authorization.startsWith('Bearer ')) {
    try {
      token = req.headers.authorization.split(' ')[1];
      const decoded = jwt.verify(token, process.env.JWT_KEY);
      req.user = await User.findById(decoded.id).select('-password');
      next();
    } catch (error) {
      res.status(401).json({ message: 'Not authorised' });
    }
  }

  if (!token) {
    res.status(401).json({ message: 'Not authorised, no token' });
  }
};
```

## Booking function

The `bookClass` function handles booking a class for a user. It first extracts the `classId` from the request body and attempts to find the class in the database. If the class is not found, it responds with a `404 Not Found` status. If the class is full, it responds with a `400 Bad Request` status. The function then checks if a booking already exists for the user and class. If a booking exists, it updates the booking status to `'booked'` and sets the booking date to the current date. If no booking exists, it creates a new booking with the user's ID, class ID, status `'booked'`, and the current date, and increments the class's `currentAttendees` count.

```
exports.bookClass = async (req, res) => {
  const { classId } = req.body;

  try {
    const selectedClass = await Class.findById(classId);

    if (!selectedClass) {
      return res.status(404).json({ message: 'Class not found' });
    }

    if (selectedClass.currentAttendees >= selectedClass.maxAttendees) {
      return res.status(400).json({ message: 'Class is full' });
    }

    // Check if a booking already exists for the user and class
    let booking = await Booking.findOne({ user: req.user._id, class: classId });

    if (booking) {
      // Update existing booking
      booking.status = 'booked';
      booking.bookingDate = new Date();
    } else {
      // Create new booking
      booking = new Booking({
        user: req.user._id,
        class: classId,
        status: 'booked',
        bookingDate: new Date(),
      });

      selectedClass.currentAttendees += 1;
      await selectedClass.save();
    }

    const savedBooking = await booking.save();

    res.status(201).json(savedBooking);
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
  }
};
```

Finally, it saves the booking to the database and responds with a `201 Created` status and the saved booking information. If an error occurs, it responds with a `500 Internal Server Error` status.



# Integral pieces of code – Conditional Statement

## Toggle Login/Logout button

```
const Header = () => {
  const [isLoading, setIsLoading] = useState(false);
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const navigate = useNavigate();
  const location = useLocation();

  useEffect(() => {
    const authStatus = localStorage.getItem("isAuthenticated");
    setIsAuthenticated(authStatus === "true");
  }, [location]);

  const handleAuthClick = () => {
    setIsLoading(true);
    if (isAuthenticated) {
      localStorage.removeItem("isAuthenticated");
      localStorage.removeItem("token");
      setIsAuthenticated(false);
      navigate('/');
    } else {
      navigate('/login');
    }
    setIsLoading(false);
  };
};
```

```
<button
  className={`button is-primary ${isLoading ? 'is-loading' : ''}`}
  onClick={handleAuthClick}
  disabled={isLoading}
>
  {isAuthenticated ? 'Logout' : 'Login'}
</button>
```

`useState(false)`: Initializes a state variable `isAuthenticated` as false. This state will track whether the user is logged in.

`useEffect(() => {...}, [location])`: Runs this effect whenever the location (i.e., current route) changes. It checks the `localStorage` for a value of `isAuthenticated` to determine if the user is logged in. If the value is "true", `setIsAuthenticated` is called with true, updating the state to reflect that the user is logged in.

`handleAuthClick`: This function is called when the user clicks the login/logout button.

If the user is logged in (`isAuthenticated` is true), it removes the authentication details (`isAuthenticated` and token) from `localStorage`, updates `isAuthenticated` to false, and navigates the user to the home page ('/').

If the user is not logged in, it navigates them to the login page ('/login').

The button label changes based on `isAuthenticated`. If the user is logged in, the button shows "Logout", otherwise it shows "Login".

# Integral pieces of code – Conditional Statement

## Dashboard button

```
{isAuthenticated && (  
  <button  
    className="button is-info"  
    onClick={handleDashboardClick}  
  >  
    Dashboard  
  </button>  
)}
```

`{isAuthenticated && ( ... )}`: This is a common pattern in React to conditionally render elements. The `&&` operator ensures that the button is only rendered if `isAuthenticated` is true.

If `isAuthenticated` is true (i.e., the user is logged in), the Dashboard button is displayed.

If `isAuthenticated` is false (i.e., the user is not logged in), the button is not rendered.

`handleDashboardClick`: This function navigates the user to the dashboard page ('/dashboard') when the button is clicked.

# Challenges

## Dynamically updating Calendar

Entering the class data into the calendar was challenging due to the need to accurately map events to specific days, handle different event types, and dynamically generate links for each event. This was solved by fetching the events, filtering them based on the current month and year, organizing them by day, and then rendering them in the calendar grid. The solution involved using React state and effects to manage the data and update the UI accordingly.

| < August 2024 > |                                       |   |   |     |     |     |
|-----------------|---------------------------------------|---|---|-----|-----|-----|
| Sun             | Mon                                   | Tue   | Wed                                     | Thu | Fri | Sat |
|                 |                                       |   |   | 1   | 2   | 3   |
| 4               | 5                                     | 6   | 7                                       | 8   | 9   | 10  |
| 11              | 12<br>CardioBox<br>7:00 pm AEST (45m) | 13  | 14                                      | 15  | 16  | 17  |
| 18              | 19                                    | 20<br>Fighters Academy<br>8:00 pm AEST (1h) | 21<br>Kids Boxing<br>4:00 pm AEST (30m) | 22  | 23  | 24  |
| 25              | 26                                    | 27  | 28                                      | 29  | 30  | 31  |

# Challenges - Conditional Statement

## Toggling Login/Logout button

```
const Header = () => {
  const [isLoading, setIsLoading] = useState(false);
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const navigate = useNavigate();
  const location = useLocation();

  useEffect(() => {
    const authStatus = localStorage.getItem("isAuthenticated");
    setIsAuthenticated(authStatus === "true");
  }, [location]);

  const handleAuthClick = () => {
    setIsLoading(true);
    if (isAuthenticated) {
      localStorage.removeItem("isAuthenticated");
      localStorage.removeItem("token");
      setIsAuthenticated(false);
      navigate('/');
    } else {
      navigate('/login');
    }
    setIsLoading(false);
  };
};
```

```
<button
  className={`button is-primary ${isLoading ? 'is-loading' : ''}`}
  onClick={handleAuthClick}
  disabled={isLoading}
>
  {isAuthenticated ? 'Logout' : 'Login'}
</button>
```

### 1. Inconsistent Authentication State:

Problem: The application might display the wrong authentication state (e.g., showing 'Login' when the user is actually logged in) due to incorrect state management or outdated values in localStorage.

Solution: Ensure that 'isAuthenticated' is accurately retrieved and updated from localStorage and is correctly synchronized with the state.

### 2. Delayed UI Updates:

Problem: There might be a delay or flicker in updating the UI elements (like the login/logout button) due to state updates not being immediately reflected.

Solution: Use state management libraries or hooks effectively to handle asynchronous updates and ensure the UI reflects the current state promptly.

### 3. Authentication Status Fluctuations:

Problem: Authentication status might inadvertently change if multiple instances of the application or browser tabs are open, leading to inconsistent behavior.

Solution: Use event listeners or synchronization mechanisms to manage changes across multiple instances or tabs.

# Challenges – Conditional Statement

## Dashboard button

```
{isAuthenticated && (  
  <button  
    className="button is-info"  
    onClick={handleDashboardClick}  
  >  
    Dashboard  
  </button>  
)}
```

### 1. Conditional Rendering Bugs:

Problem: The dashboard button might not appear or disappear as expected due to bugs in conditional rendering logic.

Solution: Carefully test the conditional rendering logic to ensure that the button appears only when the user is logged in and disappears otherwise.

### 2. State Synchronization Problems:

Problem: The `isAuthenticated` state might not sync properly with the UI, causing discrepancies between the displayed and actual state.

Solution: Use state management tools or hooks effectively to keep the authentication state synchronized across the application.

# Demo!

Try our boxing gym owner dashboard @ **fighttrack.xyz**

Login:

- Email: [admin@fighttrack.com](mailto:admin@fighttrack.com)
- Pass: admin

**Transform  
Your Gym  
Management  
Today!**

