

BRIGHT

FANTASY

LEAGUE



AGENDA



- Planning
- Demo
- Code
- Challenges
- Extensions

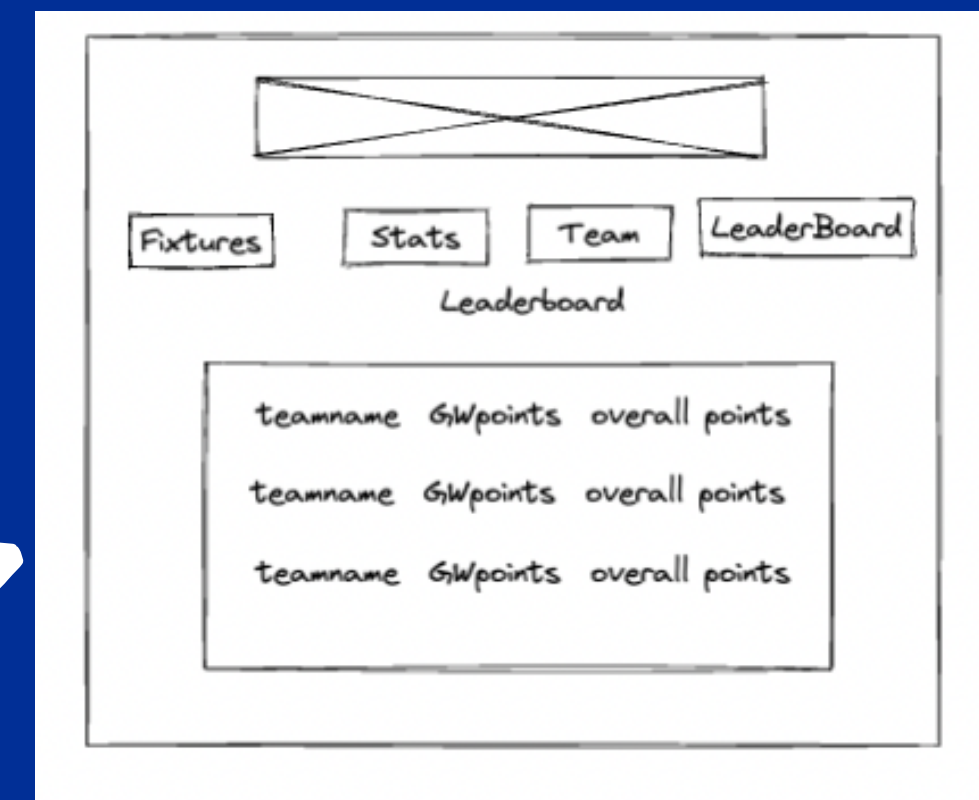
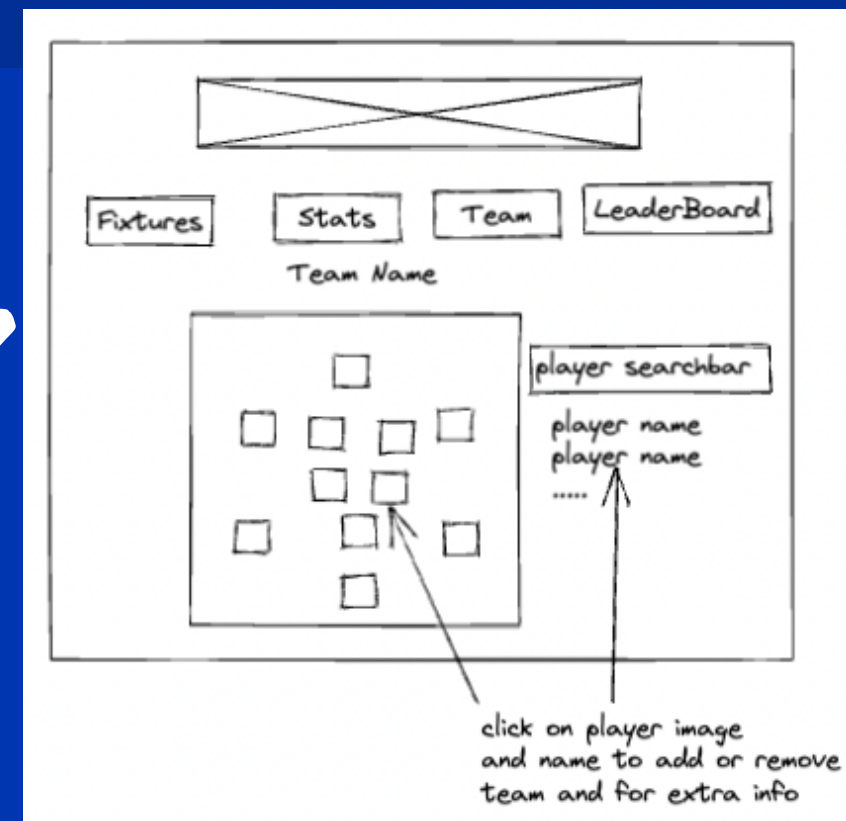
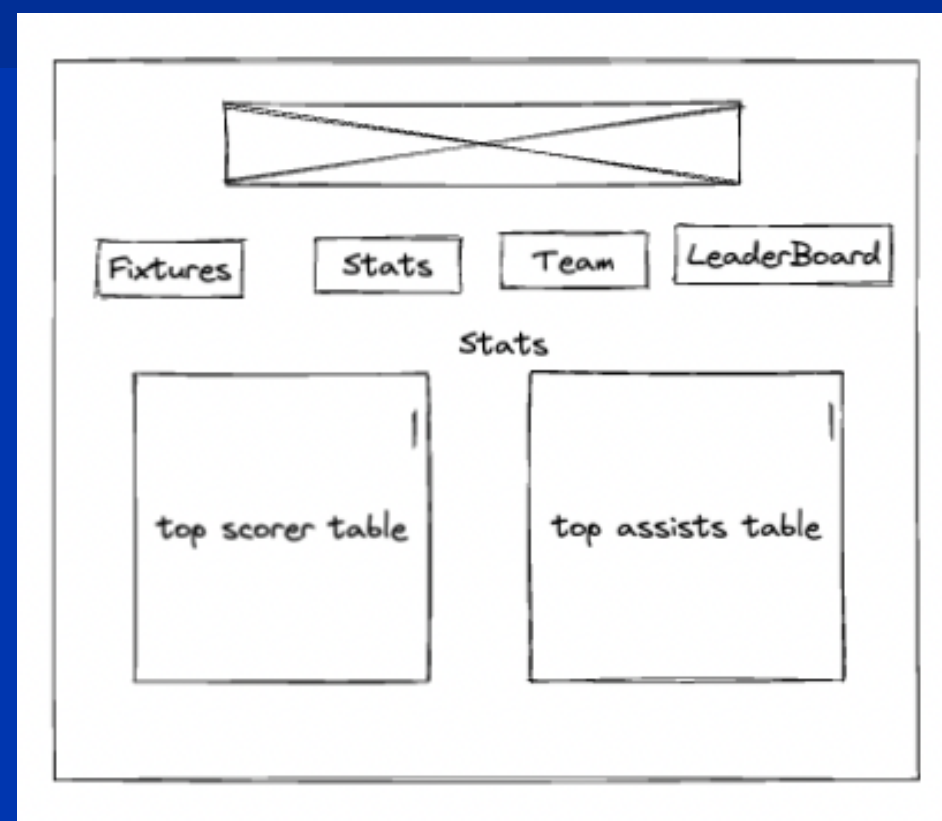
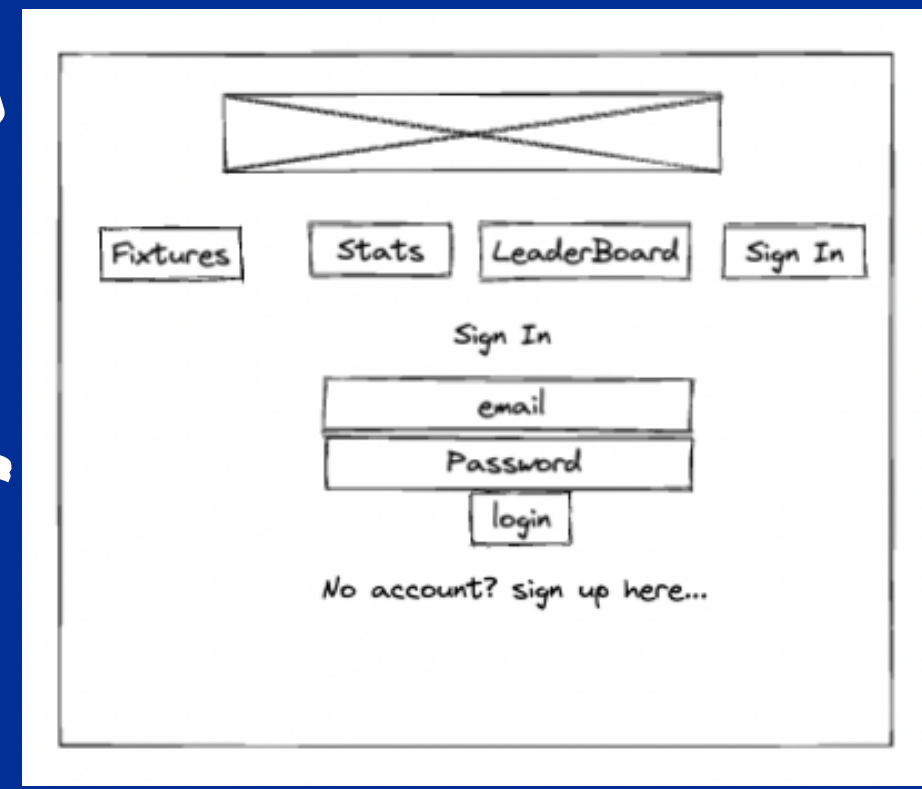
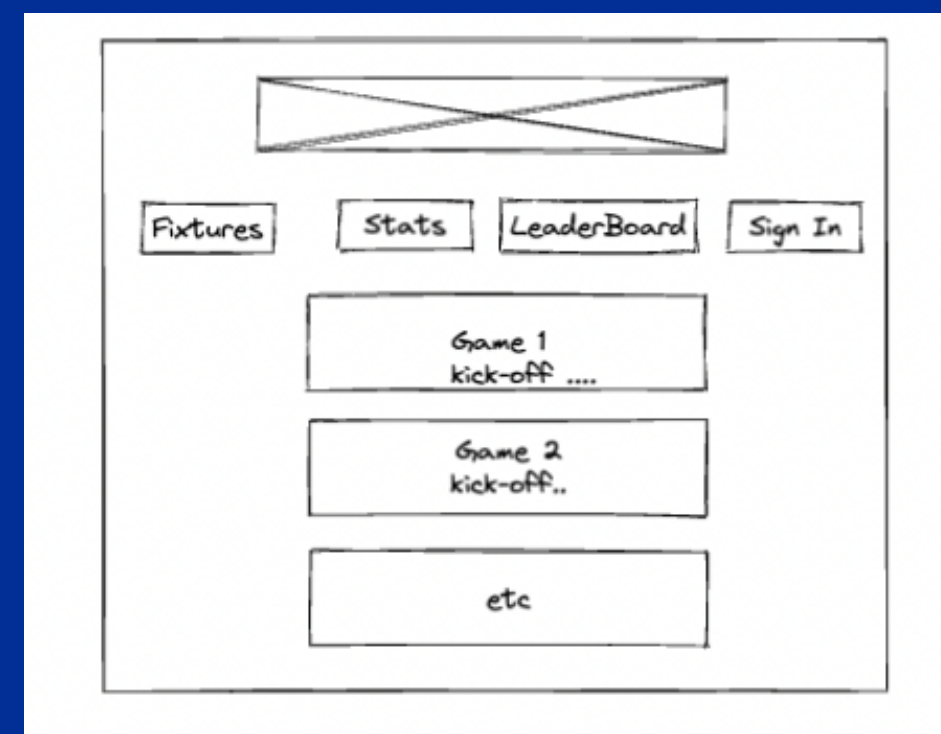
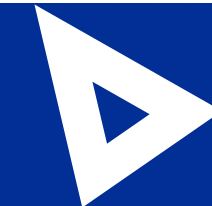




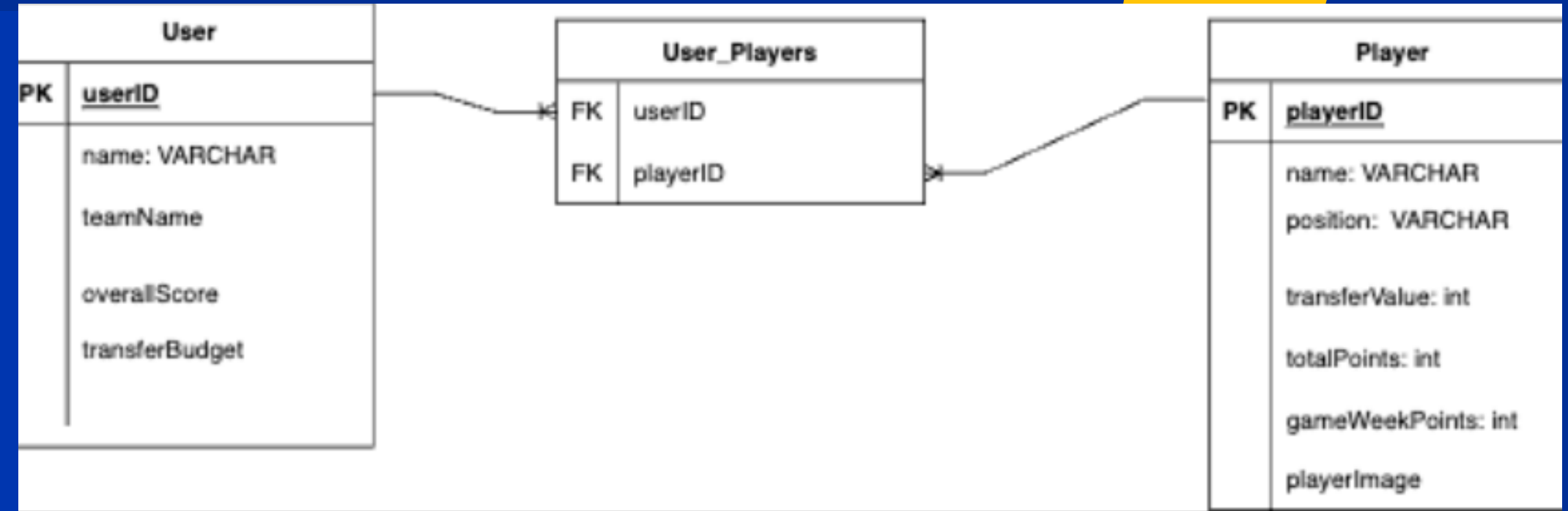
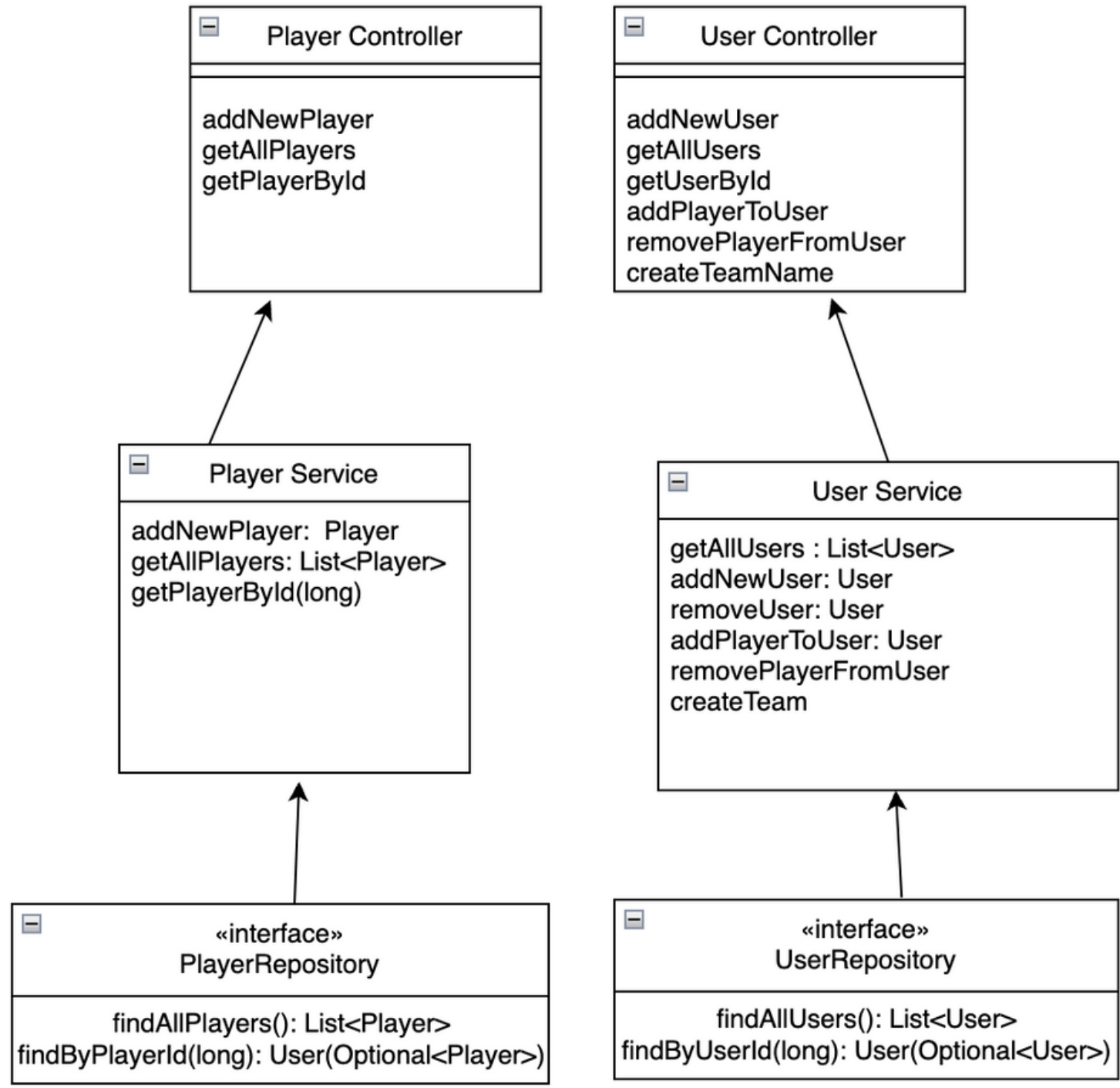
PLANNING



DESIGNING OUR WIRE FRAMES



UML DIAGRAM & ERD



SECURING OUR API

JWT Authentication

- JSON Web Token: String representing a set of claims as a JSON object
- Can be verified as it is digitally signed using a "secret key"

SECURING OUR API

Pojo's and Controllers

- AuthRequest
- AuthResponse
- AuthController – end point

SECURING OUR API

Generating Access Token

```
@Value("${app.jwt.secret}")
private String SECRET_KEY;

public String generateAccessToken(User user) {
    return Jwts.builder()
        .setSubject(String.format("%s,%s", user.getUserId(),
user.getEmail()))setIssuer("BFL")
        .setIssuedAt(new Date())
        .setExpiration(new Date(System.currentTimeMillis() +
EXPIRE_DURATION))signWith(SignatureAlgorithm.HS512, SECRET_KEY)
        .compact();
}
```

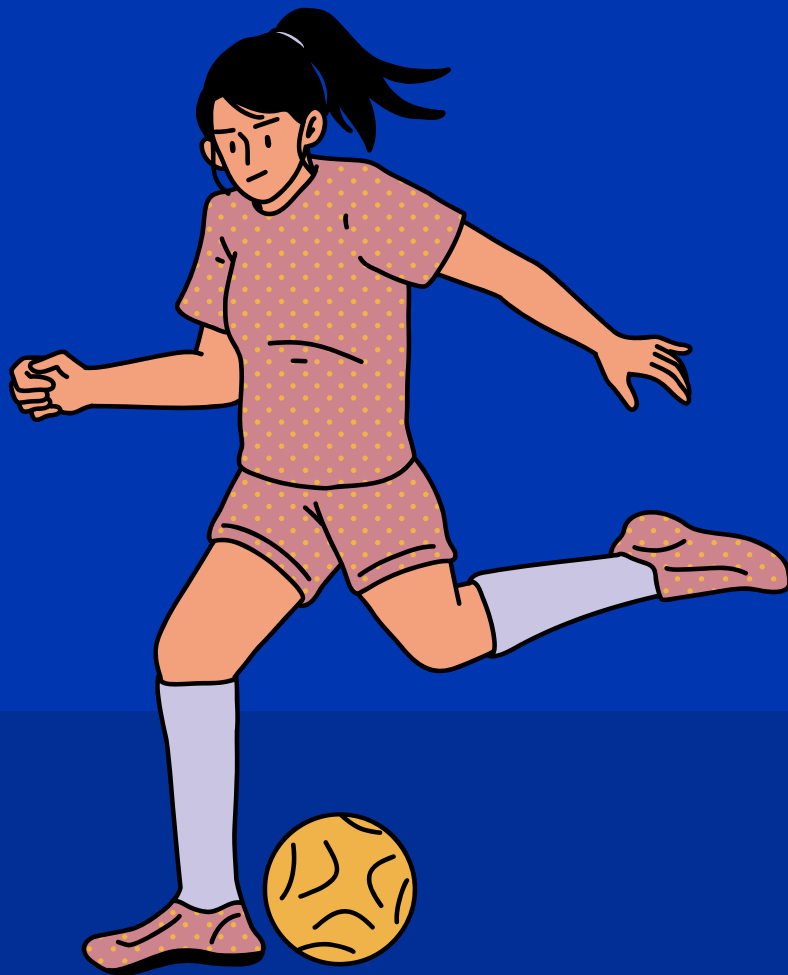

SECURING OUR API

Security Configuration

- Authentication manager to authenticate user
- For invalid password, 401 error occurs
- For valid password, generate token for authResponse

APP DEMO

OUR CODE



SCORERS TABLE

```
let top20Scorers = data.elements.sort((a, b) => b.goals_scored - a.goals_scored);
let topS = top20Scorers.slice(0,20)

const topScorers = topS.map((player, key) => {
  return (
    <tr key={key}>
      <td>{player.first_name} {player.second_name}</td>
      <td>{teamNames[player.team]}</td>
      <td>{player.goals_scored}</td>
    </tr>
  )
})
```

FIXTURES LIST

```
const fixturesList = fixtures.map((fixture)=>{
  if (fixture.event === selectedWeek){
    return(
      <div key={fixture.id} className='fixture_card'>
        <h3><img></img>{teamNames[fixture.team_h]} Vs {teamNames[fixture.team_a]}</h3>
        {displayScoreboard(fixture)}
        <div className='players-scored'>
          <ul className='left'>
            {homeGoalsScored(fixture,normalGoalIndex)}
            {awayGoalsScored(fixture,ownGoalIndex)}
          </ul>
          <ul className='right'>
            {awayGoalsScored(fixture,normalGoalIndex)}
            {homeGoalsScored(fixture,ownGoalIndex)}
          </ul>
        </div>
      </div>
    )
  }
})
```

ADDING A PLAYER TO YOUR TEAM

```
const handleClick = (e) => {
  let copiedplayer = (data.elements.find((player) => player.web_name ===
searchedPlayer.web_name))
  if (trueUser.players.length === 11) {
    alert("You've already got a full team")

  } else if (trueUserNames.includes(copiedplayer.web_name)) {
    alert("You already have that player")
  } else if (100 - transBudget < copiedplayer.now_cost/10){
    alert("You don't have enough money!")
  }
  else{
    findTrueUser();
    createPlayer({
      name: copiedplayer.web_name,
      transferValue: copiedplayer.now_cost,
      apiid: copiedplayer.id,
      position: copiedplayer.element_type
    },trueUser)
  }
}
```

```
const createPlayer = async (player, trueUser) => {
  const response = await fetch(`http://localhost:8080/players`, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(player),
  });

  const savedPlayer = await response.json();
  await fetch(
    `http://localhost:8080/user/addPlayer?userId=${trueUser.userId}&playerId=${savedPlayer.id}`,
    {
      method: "PUT",
      headers: { "Content-Type": "application/json" },
    }
  );
  await fetchUsers();
  const newUser = trueUser;
  const newPlayer = player;
  newPlayer.id = newUser.players.length + 1;
  newUser.players.push(player);
  setTrueUser(newUser);
}
```

CHALLENGES

- **Early CORS issues**
- **Implementing Spring jwt into our front end.**
- **Adding a player to the user**

IMPROVEMENTS

- Adding extra clone features
- Increase multi-user functionality
- Dynamically Responsive
- Improve pre-coding planning
- Explore Test Driven Dev (TDD)

THANK YOU



Are there any questions?

