



## Web App Design with React Final Project

**Points possible:** 70

Category	Criteria	% of Grade
<b>Functionality</b>	Does the code work?	25
<b>Organization</b>	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
<b>Creativity</b>	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
<b>Completeness</b>	All requirements of the assignment are complete.	25

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

### Coding Steps:

1. Using an online API of your choice, or multiple APIs (or if no API can be found, using an array for in-memory storage is okay as well), create a React project of your choice. You will be working on this for the next three weeks.
2. Project must meet the following criteria:
  - a. Use React Router and have at least 3 pages
  - b. Use React Bootstrap or an alternative styling library
  - c. Contain at least 10 components
  - d. Allow for all CRUD operations

### Screenshots of Code:



## Index.html

```
public > <> index.html > html > head
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="utf-8" />
5       <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6       <meta name="viewport" content="width=device-width, initial-scale=1" />
7       <meta name="theme-color" content="#000000" />
8       <meta
9         name="description"
10        content="Web site created using create-react-app"
11      />
12      <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13      <!--
14        manifest.json provides metadata used when your web app is installed on a
15        user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16      -->
17      <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18      <!--
19        Notice the use of %PUBLIC_URL% in the tags above.
20        It will be replaced with the URL of the `public` folder during the build.
21        Only files inside the `public` folder can be referenced from the HTML.
22        Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
23        work correctly both with client-side routing and a non-root public URL.
24        Learn how to configure a non-root public URL by running `npm run build`.
25      -->
26      <title>React App</title>
27    </head>
28    <body>
29      <noscript>You need to enable JavaScript to run this app.</noscript>
30      <div id="root"></div>
31      <!--
32        This HTML file is a template.
33        If you open it directly in the browser, you will see an empty page.
34        You can add webfonts, meta tags, or analytics to this file.
35        The build step will place the bundled scripts into the <body> tag.
36        To begin the development, run `npm start` or `yarn start`.
37        To create a production bundle, use `npm run build` or `yarn build`.
38      -->
39    </body>
40  </html>
41
42 |
```



## APP.JS

```
src > js App.js > ...
1  import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
2  import { useEffect, useState } from 'react';
3  import { Container, Row, Col, Navbar, Nav } from 'react-bootstrap';
4  import { tournApi } from './rest/TournApi';
5  import TournamentList from './components/TennisTournamentList';
6  import { PlayerList } from './components/TennisPlayerList';
7  import { HomePage } from './components/HomePage';
8  import { Tournament } from './components/TennisTournament';
9  import { AddPlayerForm } from './components/AddTennisPlayerForm';
10 import { NewTournamentForm } from './components/NewTennisTournamentForm';
11 import { NewPlayerForm } from './components/NewTennisPlayerForm';
12 import {
13   BrowserRouter as Router,
14   Switch,
15   Route,
16 } from "react-router-dom";
17
18 function App() {
19   const [ playerList, setPlayerList ] = useState('');
20   const [ tournamentList, setTournamentList ] = useState('');
21   const [ loadedTournament, setLoadedTournament ] = useState('');
22   const [ loadedMatches, setLoadedMatches ] = useState('');
23
24   const addTournament = async (newTournamentName) => {
25     let newTournament = {
26       name: newTournamentName,
27       players: [],
28       matches: [],
29       timestamp: Date.now()
30     }
31
32     await tournApi.postTournament(newTournament);
33     await fetchTournaments();
34   }
35
36   const deleteTournament = async (tournamentId) => {
37     await tournApi.deleteTournament(tournamentId);
38     await fetchTournaments();
39   }
40
41   const addPlayer = async (newPlayerName) => {
42     let newPlayer = {
43       name: newPlayerName,
44       points: null,
45       matches: [],
46     }
47
48     await tournApi.postPlayer(newPlayer);
49     await fetchPlayers();
50   }
51
52   const deletePlayer = async (playerId) => {
53     await tournApi.deletePlayer(playerId);
54     await fetchPlayers();
55   }
56
```



```
56
57  const addTournamentPlayer = async ( newTournamentPlayerId, tournamentId ) => {
58    console.log(newTournamentPlayerId, tournamentId);
59    let playerTournament = await tournApi.getTournament(tournamentId);
60    console.log(playerTournament);
61
62    playerTournament.players.push(newTournamentPlayerId);
63    let newTournament = await tournApi.putTournament(playerTournament);
64    setLoadedTournament(newTournament);
65  }
66
67  const deleteTournamentPlayer = async ( playerId, tournamentId ) => {
68    let playerTournament = await tournApi.getTournament(tournamentId);
69    let newPlayers = [];
70    for (let i = 0; i < playerTournament.players.length; i++) {
71      if(playerTournament.players[i] !== playerId) {
72        newPlayers.push(playerTournament.players[i]);
73      }
74    }
75    playerTournament.players = newPlayers;
76    let newTournament = await tournApi.putTournament(playerTournament);
77    setLoadedTournament(newTournament);
78  }
79
80  const fetchTournaments = async () => {
81    const tournaments = await tournApi.getAllTournaments();
82
83    setTournamentList( tournaments );
84    console.log(`tournamentList line 58 App.js`, tournamentList);
85  }
86
87  const fetchPlayers = async () => {
88    const players = await tournApi.getAllPlayers();
89    console.log(players);
90    setPlayerList(players);
91  }
92
93  const loadTournament = async (tournamentId) => {
94    let tournament = {};
95    console.log('We ran loadTournament', tournamentList);
96    tournament = await tournApi.getTournament(tournamentId);
97    setLoadedTournament(tournament);
98  }
99
100 const addMatch = async (players, tournamentId, winner) => {
101  let newMatch = {
102    players: players,
103    winner: winner,
104  }
105  let updatedMatch = await tournApi.postTournamentMatch(tournamentId, newMatch);
106  if(updatedMatch.id) {
107    let tournamentToUpdate = await tournApi.getTournament(tournamentId);
108    tournamentToUpdate.matches.push(updatedMatch.id);
109    let updatedTournament = await tournApi.putTournament(tournamentToUpdate);
110    setLoadedTournament(updatedTournament);
```



```
111     |     fetchMatches(tournamentId);
112     |   }
113   }
114
115   const fetchMatches = async (tournamentId) => {
116     let matches = await tournApi.getAllTournMatches(tournamentId);
117     setLoadedMatches(matches);
118   }
119
120   const deleteMatch = async (matchId, tournamentId) => {
121     await tournApi.deleteTournamentMatch(tournamentId, matchId);
122     let tournamentToUpdate = await tournApi.getTournament(tournamentId);
123     let matchesToUpdate = [];
124     for (let i = 0; i < tournamentToUpdate.matches.length; i++) {
125       if(tournamentToUpdate.matches[i] !== matchId) {
126         matchesToUpdate.push(tournamentToUpdate.matches[i]);
127       }
128     }
129     tournamentToUpdate.matches = matchesToUpdate;
130     let updatedTournament = await tournApi.putTournament(tournamentToUpdate);
131     setLoadedTournament(updatedTournament);
132     fetchMatches(tournamentId);
133   }
134
135   useEffect(() => {
136     fetchTournaments();
137     fetchPlayers();
138   }, []);
139
140   return (
141     <div className="App">
142
143       <Container>
144         <Row>
145           <Col>
146             <Navbar bg="light" expand="lg">
147               <Navbar.Brand href="/">Tournament Tennis Manager</Navbar.Brand>
148               <Nav className="me-auto">
149                 <Nav.Link href="/tournaments">Tournaments</Nav.Link>
150                 <Nav.Link href="/players">Players</Nav.Link>
151               </Nav>
152             </Navbar>
153
154           </Col>
155         </Row>
156         <Row>
157           <Col>
158             <Router>
159               <Switch>
160                 <Route path={`/tournaments/:tournamentId`}*>
161                   <Tournament
162                     loadTournament={loadTournament}
163                     fetchMatches={fetchMatches}
164                     fetchPlayers={fetchPlayers}
165                     playerList={playerList}
```



```
150
151
152
153
154
155
156
157
158
159     <Router>
160         <Switch>
161             <Route path={`/tournaments/:tournamentId`}*>
162                 <Tournament
163                     loadTournament={loadTournament}
164                     fetchMatches={fetchMatches}
165                     fetchPlayers={fetchPlayers}
166                     playerList={playerList}
167                     loadedTournament={loadedTournament}
168                     loadedMatches={loadedMatches}
169                     deleteAction={deleteTournamentPlayer}
170                     deleteMatchAction={deleteMatch}
171                     addMatch={addMatch}
172                 />
173                 <AddPlayerForm playerList={playerList} addPlayer={addTournamentPlayer} />
174             </Route>
175             <Route path="/tournaments">
176                 <TournamentList
177                     tournamentList={tournamentList}
178                     fetchTournaments={fetchTournaments}
179                     deleteAction={deleteTournament}
180                 />
181                 <NewTournamentForm addTournament={addTournament} />
182             </Route>
183             <Route path="/players">
184                 <PlayerList
185                     fetchPlayers={fetchPlayers}
186                     playerList={playerList}
187                     deleteAction={deletePlayer}
188                 />
189                 <NewPlayerForm addPlayer={addPlayer} />
190             </Route>
191             <Route path='/'>
192                 <HomePage />
193             </Route>
194         </Switch>
195     </Router>
196 </Col>
197 </Row>
198 </Container>
199 </div>
200 );
201 }
202
203 export default App;
204
205
206
207 |
```



## INDEX.CSS

```
src > # index.css > body
1  body {
2    margin: 0;
3    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
4    | 'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
5    | sans-serif;
6    -webkit-font-smoothing: antialiased;
7    -moz-osx-font-smoothing: grayscale;
8  }
9
10 code {
11   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
12   | monospace;
13 }
14
```

## REST / TOURNAPI.JS

```
8  class TournApi {
9    getAllTournaments = async () => {
10      try{
11        const resp = await fetch(TournApiController);
12        const json = await resp.json();
13        return json;
14      } catch (e) {
15        console.log('Error fetching tournaments: ', e);
16      }
17    }
18
19    getTournament = async (tournId) => {
20      if(tournId){
21        try {
22          const resp = await fetch(`/${TournApiController}/${tournId}`);
23          const json = await resp.json();
24          return json;
25        } catch (e) {
26          console.log(`Error fetching tournament with id ${tournId}: `, e);
27        }
28      } else {
29        console.log('No tournament ID provided');
30      }
31    }
32  }
```



```
53
54     putTournament = async (tourn) => {
55         try {
56             const resp = await fetch(` ${TournApiUrl}/ ${tourn.id}` , {
57                 method: 'PUT',
58                 headers: {
59                     'Content-Type': 'application/json'
60                 },
61                 body: JSON.stringify(tourn)
62             });
63
64             return await resp.json();
65         }
66         catch(e) {
67             console.log(`Error updating tournament with id ${tourn.id}` , e);
68         }
69     }
70
71     postTournament = async (tourn) => {
72         try {
73             const resp = await fetch(TournApiUrl, {
74                 method: 'POST',
75                 headers: {
76                     'Content-Type': 'application/json'
77                 },
78                 body: JSON.stringify(tourn)
79             });
80             return await resp.json();
81         } catch(e) {
82             console.log('Error adding new tourn' , e);
83         }
84     }
85
86     deleteTournament = async (tournId) => {
87         console.log(`Time to delete ${tournId}!` );
88         try {
89             const resp = await fetch(` ${TournApiUrl}/ ${tournId}` , {
90                 method: 'DELETE',
91                 headers: {
92                     'Content-type': 'application/json'
93                 }
94             });
95             const json = await resp.json();
96             return json;
97         } catch(e) {
98             console.log(`Error deleting tourn with id ${tournId}` , e);
99         }
100    }
```



```
81
82     getAllTournMatches = async (tournId) => {
83         try {
84             const resp = await fetch(` ${TournApiUrl}/ ${tournId}/match`);
85             const json = await resp.json();
86             return json;
87         } catch (e) {
88             console.log(`Error getting matches from tournament ${tournId}: `, e);
89         }
90     }
91
92     getTournamentMatch = async (tournId, matchId) => {
93         try {
94             const resp = await fetch(` ${TournApiUrl}/ ${tournId}/match/${matchId}`);
95             const json = await resp.json();
96             return json;
97         } catch (e) {
98             console.log(`Error fetching tournament (${tournId}) with id ${matchId}: `, e);
99         }
100    }
101
102    putTournamentMatch = async (tournId, match) => {
103        try {
104            const resp = await fetch(` ${TournApiUrl}/ ${tournId}/match/${match.id}/`, {
105                method: 'PUT',
106                headers: {
107                    'Content-Type': 'application/json'
108                },
109                body: JSON.stringify(match)
110            });
111
112            return await resp.json();
113
114        } catch(e) {
115            console.log(`Error updating tournament (${tournId}) match with id ${match.id}`, e);
116        }
117    }
118
119    postTournamentMatch = async (tournId, match) => {
120        try {
121            const resp = await fetch(` ${TournApiUrl}/ ${tournId}/match`, {
122                method: 'POST',
123                headers: {
124                    'Content-Type': 'application/json'
125                },
126                body: JSON.stringify(match)
127            });
128            return await resp.json();
129        } catch(e) {
130            console.log('Error adding new tourn', e);
131        }
132    }
133}
```



```
134     deleteTournamentMatch = async (tournId, matchId) => {
135         console.log(`Time to delete ${tournId} + ${matchId} !`);
136         try {
137             const resp = await fetch(`${TournApiUrl}/${tournId}/match/${matchId}`, {
138                 method: 'DELETE',
139                 headers: {
140                     'Content-type': 'application/json'
141                 }
142             });
143             return await resp.json();
144         } catch(e) {
145             console.log(`Error deleting tourn match with id ${tournId} + ${matchId}`, e);
146         }
147     }
148
149     getAllPlayers = async () => {
150         try {
151             const resp = await fetch(PlayerApiUrl);
152             const json = await resp.json();
153             return json;
154         } catch (e) {
155             console.log(`Error getting players`, e);
156         }
157     }
158
159     getPlayer = async (playerId) => {
160         try {
161             const resp = await fetch(`${PlayerApiUrl}/${playerId}`);
162             const json = await resp.json();
163             return json;
164         } catch (e) {
165             console.log(`Error getting player ${playerId}`, e);
166         }
167     }
168
169     postPlayer = async (player) => {
170         try {
171             const resp = await fetch (PlayerApiUrl, {
172                 method: 'POST',
173                 headers: {
174                     'Content-Type': 'application/json'
175                 },
176                     body: JSON.stringify(player),
177             });
178             return await resp.json();
179         } catch(e) {
180             console.log(`Error adding new player`, e);
181         }
182     }
```



```
184     putPlayer = async (player) => {
185         try {
186             const resp = await fetch(` ${PlayerApiUrl}/ ${player.id}` , {
187                 method: 'PUT',
188                 headers: {
189                     'Content-Type': 'application/json'
190                 },
191                 body: JSON.stringify(player)
192             });
193
194             return await resp.json();
195
196         } catch(e) {
197             console.log(`Error updating tournament with id ${player.id}` , e);
198         }
199     }
200
201     deletePlayer = async (playerId) => {
202         console.log(`Time to delete ${playerId}!`);
203         try {
204             const resp = await fetch(` ${PlayerApiUrl}/ ${playerId}` , {
205                 method: 'DELETE',
206                 headers: {
207                     'Content-type': 'application/json'
208                 }
209             );
210             return await resp.json();
211         } catch(e) {
212             console.log(`Error deleting tourn with id ${playerId}` , e);
213         }
214     }
215
216 }
217
218 export const tournApi = new TournApi();
```



## /SRC/COMPONENTS/ADDTENNISMATCHFORM.JS

```
1 import React, { useState } from 'react';
2 import Form from 'react-bootstrap/Form';
3 import Button from 'react-bootstrap/Button';
4 import { useParams } from 'react-router-dom';
5
6 export const AddMatchForm = ( props ) => {
7     const [ player1Id, setPlayer1Id ] = useState('');
8     const [ player2Id, setPlayer2Id ] = useState('');
9     const [ winner, setWinner ] = useState('');
10    const params = useParams();
11    const tournamentId = params.tournamentId;
12
13    const handleSubmit = ( e ) => {
14
15        e.preventDefault();
16        if( player1Id && player2Id && winner ) {
17            let players = [player1Id, player2Id];
18            props.addMatch(players, tournamentId, winner);
19        } else {
20            console.log(['Missing Tennis player or players']);
21        }
22    }
23
24
25    return(
26        <div>
27            <h3>Record Match</h3>
28            <Form onSubmit={handleSubmit}>
29                <Form.Group controlId="matchPlayer1">
30                    <Form.Control as="select" onChange={(e) => setPlayer1Id(e.target.value)}>
31                        <option>Select Tennis Player 1</option>
32                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
33                    </Form.Control>
34                </Form.Group>
35                <Form.Group controlId="matchPlayer2">
36                    <Form.Control as="select" onChange={(e) => setPlayer2Id(e.target.value)}>
37                        <option>Select Tennis Player 2</option>
38                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
39                    </Form.Control>
40                </Form.Group>
41                <Form.Group controlId="matchWinner">
42                    <Form.Control as="select" onChange={(e) => setWinner(e.target.value)}>
43                        <option>Set Winning Tennis Player</option>
44                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
45                    </Form.Control>
46                </Form.Group>
47                <Button variant="primary" type="submit">
48                    Submit
49                </Button>
50            </Form>
51        </div>
52    )
53}
54
55 export default AddMatchForm;
```



## SRC/COMPONENTS/ADDPENNISPLAYERFORM.JS

```
1 import React, { useState } from 'react';
2 import Form from 'react-bootstrap/Form';
3 import Button from 'react-bootstrap/Button';
4 import { useParams } from 'react-router-dom';
5
6 export const AddMatchForm = ( props ) => {
7     const [ playerId, setPlayer1Id ] = useState('');
8     const [ player2Id, setPlayer2Id ] = useState('');
9     const [ winner, setWinner ] = useState('');
10    const params = useParams();
11    const tournamentId = params.tournamentId;
12
13    const handleSubmit = ( e ) => {
14
15        e.preventDefault();
16        if( playerId && player2Id && winner ) {
17            let players = [playerId, player2Id];
18            props.addMatch(players, tournamentId, winner);
19        } else {
20            console.log('Missing Tennis player or players');
21        }
22    }
23
24
25    return(
26        <div>
27            <h3>Record Match</h3>
28            <Form onSubmit={handleSubmit}>
29                <Form.Group controlId="matchPlayer1">
30                    <Form.Control as="select" onChange={(e) => setPlayer1Id(e.target.value)}>
31                        <option>Select Tennis Player 1</option>
32                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
33                    </Form.Control>
34                </Form.Group>
35                <Form.Group controlId="matchPlayer2">
36                    <Form.Control as="select" onChange={(e) => setPlayer2Id(e.target.value)}>
37                        <option>Select Tennis Player 2</option>
38                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
39                    </Form.Control>
40                </Form.Group>
41                <Form.Group controlId="matchWinner">
42                    <Form.Control as="select" onChange={(e) => setWinner(e.target.value)}>
43                        <option>Set Winning Tennis Player</option>
44                        {props.playerList.map((player) => <option key={player.id} value={player.name}>{player.name}</option> )}
45                    </Form.Control>
46                </Form.Group>
47                <Button variant="primary" type="submit">
48                    Submit
49                </Button>
50            </Form>
51        </div>
52    )
53}
54
55 export default AddMatchForm;
```



## SRC/COMPONENTS/TENNISDELETEBUTTON.JS

```
1 import React, { useEffect, useState } from 'react';
2 import Button from 'react-bootstrap/Button';
3
4 export const DeleteButton = ( props ) => {
5     const [isLoading, setLoading] = useState(false);
6     const deleteAction = props.deleteAction;
7     const deleteId = props.id;
8     const tid = props.tid;
9
10    useEffect(() => {
11        if(isLoading) {
12            deleteAction(deleteId, tid).then(() => {
13                setLoading(false);
14            });
15        }
16    }, [isLoading]);
17
18    const handleClick = () => setLoading(true);
19
20    return (
21        <Button
22            //variant="primary"
23            variant="danger"
24            disabled={isLoading}
25            onClick={!isLoading ? handleClick : null}
26        >
27            {isLoading ? 'Loading...' : 'Delete'}
28        </Button>
29    );
30}
31
32 export default DeleteButton;
```



## SRC/COMPONENTS/HOMEPAGE.JS

```
src > components > JS HomePage.js > [@] HomePage
1  import React from 'react';
2  import { Row, Col, Card, Button } from 'react-bootstrap';
3  import { Link } from 'react-router-dom';
4
5  export const HomePage = () => {
6      return(
7          <><div>
8              <Row>
9                  <Col>
10                     <h1>Welcome: Tennis Tournament Manager CRUD</h1>
11                     <p>This application is designed to record TENNIS PLAYERS and TENNIS TOURNAMENT matches.
12                         It's best to add your Tennis players in first place.
13                         and second add your Tennis Tournament to add them.</p>
14
15                     <p>    We Start?</p>
16                     <p>    Jorge Cruz</p>
17                 </Col>
18             </Row>
19         </div><div>
20             <Row>
21                 <Col>
22                     <Card style={{ width: '30rem' }}>
23                         <Card.Body>
24                             <Card.Title>Add Tennis Tournament</Card.Title>
25                             <Card.Text>
26                                 | Add a new tennis tournament and add tennis players and record matches to it.
27                             </Card.Text>
28                             <Link to='/tournaments'><Button variant="primary">Tournaments</Button></Link>
29
30                         </Card.Body>
31                     </Card>
32                 </Col>
33             </Row>
34             <Row>
35                 <Col>
36                     <Card style={{ width: '30rem' }}>
37                         <Card.Body>
38                             <br><br>
39                             <Card.Title>Add Tennis Players</Card.Title>
40                             <Card.Text>
41                                 | Add new Tennis players available for tournaments.
42                             </Card.Text>
43                             <Link to='/players'><Button variant="primary">Players</Button></Link>
44
45                         </Card.Body>
46                     </Card>
47                 </Col>
48             </Row>
49         </div></>
50     )
51
52     export default HomePage;
53 }
```



## SRC/COMPONENTS/TENNISMATCHLIST.JS

```
1 import React from 'react';
2 import Table from 'react-bootstrap/Table';
3 import Alert from 'react-bootstrap/Alert';
4 import { DeleteButton } from './TennisDeleteButton';
5
6 export const MatchList = ( props ) => {
7     const matchPlayers = props.matchPlayers;
8     const loadedMatches = props.loadedMatches;
9
10    if(!loadedMatches || loadedMatches.length === 0) {
11        return(
12            <Alert variant='primary'>
13                No Tennis Matches Loaded
14            </Alert>
15        )
16    } else if(loadedMatches && loadedMatches.length !== 0) {
17        console.log('Loaded Matches', loadedMatches);
18        console.log('Loaded Match Players', matchPlayers);
19        return(
20            <Table striped bordered hover>
21                <thead>
22                    <tr>
23                        <th>ID</th>
24                        <th>Player 1</th>
25                        <th>Player 2</th>
26                        <th>Winner</th>
27                        <th>Delete</th>
28                    </tr>
29                </thead>
30                <tbody>
31                    {loadedMatches.map((match) => (
32                        <tr key={match.id}>
33                            <td>{match.id}</td>
34                            <td>{match.players[0]}</td>
35                            <td>{match.players[1]}</td>
36                            <td>{match.winner === null ? "TBA" : match.winner}</td>
37                            <td><DeleteButton tid={match.tournamentId} id={match.id} deleteAction={props.deleteAction} /></td>
38                        </tr>
39                    )));
40                </tbody>
41            </Table>
42        )
43    } else {
44        <Alert variant='primary'>
45            No Tennis Matches Loaded
46        </Alert>
47    }
48
49    return(
50        <div>Tennis Match List Goes Here</div>
51    )
52}
53
54
55 export default MatchList;
56
57
58
```



## SRC/COMPONENTS/NEWTENNISPLAYFORM.JS

```
1 import React, { useState } from 'react';
2 import Form from 'react-bootstrap/Form';
3 import Button from 'react-bootstrap/Button';
4
5 export const NewPlayerForm = ( props ) => {
6     const [ playerName, setPlayerName ] = useState( '' );
7
8     const handleSubmit = ( e ) => {
9         e.preventDefault();
10        if( playerName ) {
11            props.addPlayer( playerName );
12        } else {
13            console.log('invalid input - player form');
14            console.log( playerName );
15        }
16        setPlayerName( '' );
17    }
18
19    return (
20        <div>
21            <h3>Add New Tennis Player</h3>
22            <Form onSubmit={handleSubmit}>
23                <Form.Group controlId="newPlayerName">
24                    <Form.Label>Tennis Player Name</Form.Label>
25                    <Form.Control
26                        type="text"
27                        placeholder="Enter Tennis Player Name"
28                        onChange={(e) => setPlayerName(e.target.value)}
29                        value={playerName}
30                    />
31                </Form.Group>
32                <Button variant="primary" type="submit" size="lg">
33                    Submit
34                </Button>
35            </Form>
36        </div>
37    )
38}
39
40 export default NewPlayerForm;
41
42
```



## SRC/COMPONENTS/NEWTENNISTOURNAMENTFORM.JS

```
1 import React, { useState } from 'react';
2 import Form from 'react-bootstrap/Form';
3 import Button from 'react-bootstrap/Button';
4
5 export const NewTournamentForm = ( props ) => {
6     const [ tournamentName, setTournamentName ] = useState('');
7
8     const handleSubmit = ( e ) => {
9         e.preventDefault();
10        if( tournamentName ) {
11            props.addTournament( tournamentName );
12        } else {
13            console.log('invalid input - tournament form');
14            console.log( tournamentName );
15        }
16        setTournamentName( '' );
17    }
18
19    return (
20        <div>
21            <h3>Add Tennis Tournament</h3>
22            <Form onSubmit={handleSubmit}>
23                <Form.Group controlId="newTournamentName">
24                    <Form.Label>Tennis Tournament Name</Form.Label>
25                    <Form.Control
26                        type="text"
27                        placeholder="Enter Tennis Tournament Name"
28                        onChange={ ( e ) => setTournamentName( e.target.value ) }
29                        value={ tournamentName }
30                    />
31                </Form.Group>
32                <Button variant="primary" type="submit">
33                    Submit
34                </Button>
35            </Form>
36        </div>
37    )
38}
39
40 export default NewTournamentForm;
```



## /SRC/COMPONENTS/TENNISPLAYERLIST.JS

```
1  import React from 'react';
2  import Table from 'react-bootstrap/Table';
3  import Alert from 'react-bootstrap/Alert';
4  import { DeleteButton } from './TennisDeleteButton';
5
6  export const PlayerList = ( props ) => {
7      const players = props.playerList;
8
9      console.log(props.playerList);
10     if(players !== undefined && players.length > 0) {
11         return(
12             <div className="player-list col">
13                 <Table striped bordered hover>
14                     <thead>
15                         <tr>
16                             <th>ID</th>
17                             <th>Player Name</th>
18                             <th>Delete</th>
19                         </tr>
20                     </thead>
21                     <tbody>
22                         {players.map((player) => (
23                             <tr key={player.id}>
24                                 <td>{player.id}</td>
25                                 <td>{player.name}</td>
26                                 <td><DeleteButton id={player.id} deleteAction={props.deleteAction} tid={props.tid}></td>
27                             </tr>
28                         ))}
29                     </tbody>
30                 </Table>
31             </div>
32         )
33     } else {
34         return(
35             <div>
36                 <Alert variant='primary'>
37                     No Players Found
38                 </Alert>
39             </div>
40         )
41     }
42 }
43
44 export default PlayerList;
45
```



## /SRC/COMPONENTS/TENNISTOURNAMENT.JS

```
1 import React, { useEffect } from 'react';
2 import Table from 'react-bootstrap/Table';
3 import { PlayerList } from './TennisPlayerList';
4 import { useParams } from 'react-router-dom';
5 import { AddMatchForm } from './AddTennisMatchForm';
6 import { MatchList } from './TennisMatchList';
7
8 export const Tournament = (props) => {
9     const params = useParams();
10    const loadTournament = props.loadTournament;
11    const playerList = props.playerList;
12    const loadedTournament = props.loadedTournament;
13    const fetchMatches = props.fetchMatches;
14    const loadedMatches = props.loadedMatches;
15    const fetchPlayers = props.fetchPlayers;
16
17    useEffect(() => {
18        if(params.tournamentId && !loadedTournament) {
19            console.log('Tournament loaded on Line 13 in Tournament.js');
20            loadTournament(params.tournamentId);
21            fetchMatches(params.tournamentId);
22        }
23        if(!playerList) {
24            fetchPlayers();
25        }
26    })
27
28    if(params.tournamentId && loadedTournament && props.playerList.length > 0) {
29        const tournamentPlayers = props.playerList.filter((player) => loadedTournament.players.includes(player.id));
30        console.log('Line 12 Tournament', loadedTournament);
31        console.log('Line 13 Tournament', loadedTournament.players);
32        return(
33            <div className="tournament">
34                <Table striped bordered hover>
35                    <tbody>
36                        <tr>
37                            <td>Tournament ID: {loadedTournament.id}</td>
38                            <td>Name: {loadedTournament.name}</td>
39                            <td>Date Created: {new Date(loadedTournament.timestamp).toLocaleString('en-us')}</td>
40                        </tr>
41                    </tbody>
42                </Table>
43                <PlayerList
44                    playerList={tournamentPlayers}
45                    deleteAction={props.deleteAction}
46                    tid={loadedTournament.id}
47                />
48                <MatchList loadedMatches={loadedMatches} deleteAction={props.deleteMatchAction} matchPlayers={playerList} />
49                <AddMatchForm playerList={tournamentPlayers} loadedTournament={loadedTournament} addMatch={props.addMatch} />
50            </div>
51        )
52    }
53
54    return (
55        <div className="tournament">
56            Tennis Tournament Not Found
57        </div>
58    )
59 }
60 }
61 export default Tournament;
```



## /SRC/COMPONENTS/TENNISTOURNAMENTLIST.JS

```
1 import React from 'react';
2 import Table from 'react-bootstrap/Table';
3 import Alert from 'react-bootstrap/Alert';
4 import { Link, useRouteMatch } from 'react-router-dom';
5 import { DeleteButton } from './TennisDeleteButton';
6
7 export const TournamentList = ( props ) => {
8
9     const tournamentList = props.tournamentList;
10    let rMatch = useRouteMatch();
11
12    if(!tournamentList || tournamentList.length === 0) {
13        return(
14            <div>
15                <Alert variant='primary'>
16                    No Tournaments Found
17                </Alert>
18            </div>
19        )
20    } else {
21        return (
22            <div className="tournament-list col">
23                <Table striped bordered hover>
24                    <thead>
25                        <tr>
26                            <th>ID</th>
27                            <th>Name</th>
28                            <th>Player Count</th>
29                            <th>Date</th>
30                            <th>Delete</th>
31                        </tr>
32                    </thead>
33                    <tbody>
34                        {tournamentList.map((tournament) => (
35                            <tr key={tournament.id}>
36                                <td>{tournament.id}</td>
37                                <td><Link to={`${rMatch.url}/${tournament.id}`}>{tournament.name}</Link></td>
38                                <td>{tournament.players.length}</td>
39                                <td>{new Date(tournament.timestamp).toLocaleString('en-us')}
```



## PUBLIC/REPORTWEBITALS.JS

```
public > JS reportWebitals.js > ...
1  const reportWebVitals = onPerfEntry => {
2    if (onPerfEntry && onPerfEntry instanceof Function) {
3      import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
4        getCLS(onPerfEntry);
5        getFID(onPerfEntry);
6        getFCP(onPerfEntry);
7        getLCP(onPerfEntry);
8        getTTFB(onPerfEntry);
9      });
10    }
11  };
12
13  export default reportWebVitals;
14
15
16  i
```



## Screenshots of Running Application:

Welcome: Tennis Tournament Manager CRUD

This application is designed to record TENNIS PLAYERS and TENNIS TOURNAMENT matches. It's best to add your Tennis players in first place. and second add your Tennis Tournament to add them.

We Start?

Jorge Cruz

Add Tennis Tournament

Add a new tennis tournament and add tennis players and record matches to it.

Tournaments

Add Tennis Players

Add new Tennis players available for tournaments.

Players

## TENNIS TOURNAMENT MAIN SCREEN

ID	Name	Player Count	Date	Delete
3	<a href="#">Masters de Indian Wells</a>	2	3/16/2022, 11:57:03 AM	<button>Delete</button>
4	<a href="#">Miami Open</a>	0	3/16/2022, 11:57:27 AM	<button>Delete</button>
5	<a href="#">Rolex Monte-Carlo Masters</a>	0	3/16/2022, 11:57:47 AM	<button>Delete</button>
6	<a href="#">Australian Open</a>	0	3/16/2022, 11:57:56 AM	<button>Delete</button>
7	<a href="#">ATP Masters 1000 Rome</a>	0	3/16/2022, 11:58:11 AM	<button>Delete</button>

Add Tennis Tournament

Tennis Tournament Name

Enter Tennis Tournament Name

Submit



## ADD TENNIS TOURNAMENT

### BEFORE “ADD TOURNAMENT”.

localhost:3000/tournaments

ID	Name	Player Count	Date	Delete
3	<a href="#">Masters de Indian Wells</a>	2	3/16/2022, 11:57:03 AM	<button>Delete</button>
4	<a href="#">Miami Open</a>	0	3/16/2022, 11:57:27 AM	<button>Delete</button>
5	<a href="#">Rolex Monte-Carlo Masters</a>	0	3/16/2022, 11:57:47 AM	<button>Delete</button>
6	<a href="#">Australian Open</a>	0	3/16/2022, 11:57:56 AM	<button>Delete</button>
7	<a href="#">ATP Masters 1000 Rome</a>	0	3/16/2022, 11:58:11 AM	<button>Delete</button>

### Add Tennis Tournament

Tennis Tournament Name

Submit

### AFTER “ADD TOURNAMENT”

7	<a href="#">ATP Masters 1000 Rome</a>	0	3/16/2022, 11:58:11 AM	<button>Delete</button>
---	---------------------------------------	---	------------------------	-------------------------

### Add Tennis Tournament

Tennis Tournament Name

Submit

### Add Tennis Tournament

Tennis Tournament Name

Submit

13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29

### Add Tennis Tournament

Tennis Tournament Name

Submit



## AFTER “ADD TOURNAMENT”. ADD 3 TOURNAMENTS

1. Rolland Garros
2. US Open
3. Wimbledon

Tournament Tennis Manager				
ID	Name	Player Count	Date	Delete
3	<a href="#">Masters de Indian Wells</a>	2	3/16/2022, 11:57:03 AM	<button>Delete</button>
4	<a href="#">Miami Open</a>	0	3/16/2022, 11:57:27 AM	<button>Delete</button>
5	<a href="#">Rolex Monte-Carlo Masters</a>	0	3/16/2022, 11:57:47 AM	<button>Delete</button>
6	<a href="#">Australian Open</a>	0	3/16/2022, 11:57:56 AM	<button>Delete</button>
7	<a href="#">ATP Masters 1000 Rome</a>	0	3/16/2022, 11:58:11 AM	<button>Delete</button>
8	<a href="#">Rolland Garros</a>	0	3/16/2022, 4:09:54 PM	<button>Delete</button>
9	<a href="#">US Open</a>	0	3/16/2022, 4:11:14 PM	<button>Delete</button>
10	<a href="#">Wimbledon</a>	0	3/16/2022, 4:11:38 PM	<button>Delete</button>

### Add Tennis Tournament

Tennis Tournament Name

Submit

## ADD NEW TOURNAMENT “WIMBLEDON 2”. ( to delete)

10	<a href="#">Wimbledon</a>	0	3/16/2022, 4:11:38 PM	<button>Delete</button>
11	<a href="#">Wimbledon 2</a>	0	3/16/2022, 4:13:25 PM	<button>Delete</button>

### Add Tennis Tournament

Tennis Tournament Name

Submit



# PROMINEO TECH

## AFTER DELETE

10	<a href="#">Wimbleon</a>	0	3/16/2022, 4:11:38 PM	<button>Delete</button>
11	<a href="#">Wimbledon 2</a>	0	3/16/2022, 4:13:25 PM	<button>Loading...</button>

### Add Tennis Tournament

Tennis Tournament Name

Submit

localhost:3000/tournaments

[Android] Root pa... 🔍 Inicio 📌 Pestañas guardadas... 📸 FOTOGRAFIA 🕒 De actualidad 🕒 20+min 📰 DERECHO PROPIE... 🌎 mundial 🌐 Desconexión 🌐 » | 📌 Otros favoritos | 📱 Lista de le...

Tournament Tennis Manager Tournaments Players

ID	Name	Player Count	Date	Delete
3	<a href="#">Masters de Indian Wells</a>	2	3/16/2022, 11:57:03 AM	<button>Delete</button>
4	<a href="#">Miami Open</a>	0	3/16/2022, 11:57:27 AM	<button>Delete</button>
5	<a href="#">Rolex Monte-Carlo Masters</a>	0	3/16/2022, 11:57:47 AM	<button>Delete</button>
6	<a href="#">Australian Open</a>	0	3/16/2022, 11:57:56 AM	<button>Delete</button>
7	<a href="#">ATP Masters 1000 Rome</a>	0	3/16/2022, 11:58:11 AM	<button>Delete</button>
8	<a href="#">Roland Garros</a>	0	3/16/2022, 4:09:54 PM	<button>Delete</button>
9	<a href="#">US Open</a>	0	3/16/2022, 4:11:14 PM	<button>Delete</button>
10	<a href="#">Wimbledon</a>	0	3/16/2022, 4:11:38 PM	<button>Delete</button>

### Add Tennis Tournament

Tennis Tournament Name

Submit



## TENNIS “PLAYERS” MAIN SCREEN.

localhost:3000/players

ID	Player Name	Delete
1	Rafael Nadal	Delete
2	Roger Federer	Delete
3	Novak Djokovic	Delete
4	Alexander Zverev	Delete
5	Matteo Berrettini	Delete
6	Andrey Rublev	Delete
7	Pablo Carreno Busta	Delete
8	Cameron Norrie	Delete
9	Diego Schwartzman	Delete

### Add New Tennis Player

Tennis Player Name

Submit

## ADD NEW TENNIS PLAYER

		Delete
9	Diego Schwartzman	Delete

### Add New Tennis Player

Tennis Player Name

Submit



## AFTER ADD → 9 PLAYERS

localhost:3000/players

ID	Player Name	Delete
1	Rafael Nadal	Delete
2	Roger Federer	Delete
3	Novak Djokovic	Delete
4	Alexander Zverev	Delete
5	Matteo Berrettini	Delete
6	Andrey Rublev	Delete
7	Pablo Carreno Busta	Delete
8	Cameron Norrie	Delete
9	Diego Schwartzman	Delete

## BEFORE → 12 PLAYERS

localhost:3000/players

Tournament Tennis Manager   Tournaments   Players

ID	Player Name	Delete
1	Rafael Nadal	Delete
2	Roger Federer	Delete
3	Novak Djokovic	Delete
4	Alexander Zverev	Delete
5	Matteo Berrettini	Delete
6	Andrey Rublev	Delete
7	Pablo Carreno Busta	Delete
8	Cameron Norrie	Delete
9	Diego Schwartzman	Delete
10	Reilly Opelka	Delete
11	Taylor Fritz	Delete
12	Lorenzo Sonego	Delete

URL to GitHub Repository:

[https://github.com/rafacruz78gt/finalproject\\_FRONTEND\\_ELCAMINO](https://github.com/rafacruz78gt/finalproject_FRONTEND_ELCAMINO)