

L1 REACTJS LAB ASSESSMENT

SRIENATH VAISAKH

EMP ID:12215

Q2) Online Campaign management System - campaign management consists of a sequence of marketing-related activities planned and executed to achieve the best results in an election.

Read the below instructions:

1. Analyze the given problem statement using W3H techniques and document it.
2. Create a web application for the given scenario using React as a front end.
3. Configure JSON server and consider JSON file as your database.
4. Perform all the CRUD operations using React HttpClient Service.
5. Perform the required validations for all the fields.
6. Application must be Responsive (Including Mobile View) and Single Page Application (SPA) containing Multiple Views.
7. UI must be a Rich UI and look & Feel.
8. Maintain the Code Quality

1)W3H

Online Campaign Management System	
What?	How?
<p>1. What are the modules required? Ans: a) Campaign module b) Candidate Module</p> <p>2. What are the Functionalities required for Party Module? Ans: Campaign Module: 1. Perform CRUD operations to add/update/delete Venue details for campaign. 2. List details of parties using the venue for campaign along with duration and candidate name.</p> <p>3. What are the Functionalities required for Candidate Module? Ans: Candidate Module: 1. Perform CRUD operations to add/update delete candidate details involved in campaign such as their name, party they belong to etc...</p> <p>4. What are the fields required for the Campaign Module? Ans: campaignID, campaignName,venueLocation, candidateName, startTime, endTime, partyName, email.</p> <p>5. What are the fields required for Candidate Module? Ans: candidateName, candidateID, age, gender, email, partyName.</p>	<p>1) Campaign: a) Add Campaign Details <u>Method 1:</u> To add Campaign details, click particular add button in ViewCampaign Page and add by entering all fields of the module in AddCampaign page. <u>Method 2:</u> To add Campaign details, click the add button in ViewCampaign Page and add by entering email and automatically adding information available for the email for each field in AddCampaign page.</p> <p>b) Update Campaign Details <u>Method 1:</u> To update Campaign details, click update button for a campaign based on campaignID and edit by entering all fields of the module in UpdateCampaign page, the campaignID cannot be updated. <u>Method 2:</u> To update Campaign details, click particular update button for a campaign based on campaignName and edit by entering all fields of the module in UpdateCampaign page, the campaignName cannot be updated.</p> <p>c) Delete Campaign Details <u>Method 1:</u> To delete Campaign details, click the delete button for a campaign based on campaignID and with confirmation the page will be reloaded, and the campaign record deleted. <u>Method 2:</u> To delete Campaign details, click the delete button for a campaign based on campaignName and with confirmation the page will be reloaded, and the campaign record deleted.</p>

	<p>2) Candidate:</p> <p>a) Add Candidate Details <u>Method 1:</u> To add Candidate details, click the particular add button in ViewCandidate Page and add by entering all fields of the module in AddCandidate page. <u>Method 2:</u> To add Candidate details, click the add button in ViewCandidate Page and add by entering email and automatically adding information available for the email for each field in AddCandidate page.</p> <p>b) Update Candidate Details <u>Method 1:</u> To update Candidate details, click particular update button for a candidate based on candidateID and edit by entering all fields of the module in UpdateCandidate page, the candidate ID cannot be updated. <u>Method 2:</u> To update Candidate details, click particular update button for a campaign based on candidateName and edit by entering all fields of the module in UpdateCandidate page, the candidateName cannot be updated.</p> <p>c) Delete Candidate Details <u>Method 1:</u> To delete Candidate details, click the delete button for a candidate based on candidateID and with confirmation the page will be reloaded, and the candidate record deleted. <u>Method 2:</u> To delete Candidate details, click the delete button for a candidate based on candidateName and with confirmation the page will be reloaded, and the candidate record deleted.</p>
<p>1) Campaign:</p> <p>a) Add Campaign Details <u>Method 1:</u> To add Campaign details, click particular add button in ViewCampaign Page and add by entering all fields of the module in AddCampaign page. (If we add Campaign by entering all fields of campaign module then it will be easier to provide accurate information required for each campaign details)</p> <p>b) Update Campaign Details <u>Method 1:</u> To update Campaign details, click update button for a campaign based on campaignID and edit by entering all fields of the module in UpdateCampaign page, the campaignID cannot be updated. (If we update Campaign based on CampaignID then it will be more efficient and less time consuming they can access all details of the campaign easily from one information and the campaignID will be unique for each.)</p> <p>c) Delete Campaign Details <u>Method 1:</u> To delete Campaign details, click the delete button for a campaign based on campaignID and with confirmation the page will be reloaded, and the campaign record deleted.</p>	<p>1) Campaign:</p> <p>a) Add Campaign Details <u>Method 2:</u> To add Campaign details, click the add button in ViewCampaign Page and add by entering email and automatically adding information available for the email for each field in AddCampaign page. (If we add Campaign by entering email and use automatic add information available for email as cookies then it will be harder to provide accurate information required for each campaign details as the fields may vary from information available in cookies for the email)</p> <p>b) Update Campaign Details <u>Method 2:</u> To update Campaign details, click particular update button for a campaign based on campaignName and edit by entering all fields of the module in UpdateCampaign page, the campaignName cannot be updated. (If we update Campaign based on CampaignName then it will be less efficient and not accurate since two campaigns may have similar name)</p> <p>c) Delete Campaign Details <u>Method 2:</u> To delete Campaign details, click the delete button for a campaign based on campaignName and with</p>

<p>(If we delete Campaign based on CampaignID then it will be more efficient and less time consuming they can access all details of the campaign easily from one information and the campaignID will be unique for each.)</p>	<p><u>confirmation the page will be reloaded, and the campaign record deleted.</u></p> <p>(If we delete Campaign based on CampaignName then it will be less efficient and not accurate since two campaigns may have similar name)</p>
<p>2) Candidate: a) Add Candidate Details <u>Method 1:</u> To add Candidate details, click the particular add button in ViewCandidate Page and add by entering all fields of the module in AddCandidate page. (If we add Candidate by entering all fields of candidate module then it will be easier to provide accurate information required for each campaign details)</p> <p>b) Update Candidate Details <u>Method 1:</u> To update Candidate details, click particular update button for a candidate based on candidateID and edit by entering all fields of the module in UpdateCandidate page, the candidate ID cannot be updated. (If we update Candidate based on CandidateID then it will be more efficient and less time consuming they can access all details of the candidate easily from one information and the candidateID will be unique for each.)</p> <p>c) Delete Candidate Details <u>Method 1:</u> To delete Candidate details, click the delete button for a candidate based on candidateID and with <u>confirmation the page will be reloaded, and the candidate record deleted.</u> (If we delete Candidate based on CandidateID then it will be more efficient and less time consuming they can access all details of the candidate easily from one information and the candidateID will be unique for each.)</p>	<p>2) Candidate: a) Add Candidate Details <u>Method 2:</u> To add Candidate details, click the add button in ViewCandidate Page and add by entering email and automatically adding information available for the email for each field in AddCandidate page. (If we add Candidate by entering email and use automatic add information available for email as cookies then it will be harder to provide accurate information required for each candidate details as the fields may vary from information available in cookies for the email)</p> <p>b) Update Candidate Details <u>Method 2:</u> To update Candidate details, click particular update button for a candidate based on candidateName and edit by entering all fields of the module in UpdateCandidate page, the candidateName cannot be updated. (If we update Candidate based on CandidateName then it will be less efficient and not accurate since two candidates may have similar name)</p> <p>c) Delete Candidate Details <u>Method 2:</u> To delete Candidate details, click the delete button for a candidate based on candidateName and with <u>confirmation the page will be reloaded, and the candidate record deleted.</u> (If we delete Candidate based on CandidateName then it will be less efficient and not accurate since two candidates may have similar name)</p>
<p>Why?</p>	<p>Why not?</p>

ReactJS Output Screenshots

Online Campaign Management System Options ▼

Online Campaign Management System Options ▼

- Campaign Details
- Candidate Details

Online Campaign Management System

[Add Campaign +](#)

id	campaignName	venueLocation	partyName	candidateName	startTime	endTime	email	Action
2cf6	Abki Baar Modi Sarkar	Periyar, Madurai	BJP	Radhika Sarathkumar	9:30 A.M	12:00 P.M	radhikas123@gmail.com	Update Delete
3dg7	Garibi Hatao	Ponmeni, Madurai	Congress	Kamal Haasan	11:30 A.M	2:00 P.M	aandavar111@gmail.com	Update Delete
4eh8	Udhikkum Suriyan	Periyar, Madurai	DMK	Udhayanidhi Stalin	4:30 P.M	7:00 P.M	udhayas222@gmail.com	Update Delete
08a9	Sarkar	Koyambedu	TVK	Joseph Vijay	12:00 P.M	3:00 P.M.	VijayJ68@gmail.com	Update Delete

ADD CAMPAIGN DATA

Campaign Name :

Venue Location :

Party Name :

Candidate Name :

Start Time :

End Time :

Email :

[Submit](#)

UPDATE CAMPAIGN DATA

ID :

2cf6

Campaign Name :

Venue Location :

Party Name :

Candidate Name :

Start Time :

End Time :

Email :

Update

Online Campaign Management System

Add Candidate +

id	candidateName	Age	Gender	Email	partyName	Action
4ab7	Radhika Sarathkumar	53	Female	radhikas123@gmail.com	BJP	Update Delete
5bc8	Kamal Haasan	68	Male	aandavar1111gmail.com	Congress	Update Delete
6cd7	Udhayanidhi Stalin	37	Male		DMK	Update Delete

add candidate.js

```
import axios from "axios";
import React from "react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import "./App2.css";

function AddCandidate() {
  const [inputData, setInputData] = useState({
    candidateName: "",
    Age: "",
    Gender: "",
    Email: "",
    partyName: ""
  });

  const naviget = useNavigate();

  let handleSubmit = (e) => {
    e.preventDefault();
    let result = validateValues(inputData);
    // setSubmitting(true);

    if (result === true) {
      axios
        .post("http://localhost:5000/candidate", inputData)
        .then((res) => {
          alert("Data added Successfully");
          naviget("/viewcandidate");
          console.log(res.data);
        })
        .catch((err) => console.log(err));
    } else {
      alert("Please Enter the Valid Inputs!!!");
    }
  };

  const validateValues = (inputData) => {
    if (inputData.candidateName.length === 0) {
      alert("Please enter Candidate Name !!! ");
      return false;
    } else if (inputData.Age.length === 5) {
      alert("Please enter Age !!!");
      return false;
    }
  };
}
```



```

    } else if (inputData.Email.length === 0) {
      alert("Please enter Email !!!");
      return false;
    } else if (inputData.partyName.length === 0) {
      alert("Please enter Party Name !!!");
      return false;
    } else {
      return true;
    }
  };

  return (
    <div
      id="add2"
      className="d-flex w-100 vh-100 justify-content-center align-items-center "
    >
      <div className="w-50 border bg-light p-5">
        <form onSubmit={handleSubmit}>
          <h1>ADD CANDIDATE DATA</h1>
          <div>
            <label htmlFor="candidatename">Candidate Name :</label>
            <input
              type="text"
              name="candidatename"
              className="form-control"
              onChange={(e) =>
                setInputData({ ...inputData, candidateName: e.target.value })
              }
            />
          </div>

          <div>
            <label htmlFor="age">Age :</label>
            <input
              type="text"
              name="age"
              className="form-control"
              onChange={(e) =>
                setInputData({ ...inputData, Age: e.target.value })
              }
            />
          </div>

          <div>
            <label htmlFor="gender">Gender :</label>

```

```

        <input
          type="text"
          name="gender"
          className="form-control"
          onChange={e =>
            setInputData({ ...inputData, Gender: e.target.value })
          }
        />
      </div>

      <div>
        <label htmlFor="email">Email :</label>
        <input
          type="email"
          name="email"
          className="form-control"
          onChange={e =>
            setInputData({ ...inputData, Email: e.target.value })
          }
        />
      </div>

      <div>
        <label htmlFor="partyname">Party Name :</label>
        <input
          type="text"
          name="partyname"
          className="form-control"
          onChange={e =>
            setInputData({ ...inputData, partyName: e.target.value })
          }
        />
      </div>

      <br />

      <button className="btn btn-info ">Submit</button>
    </form>

  </div>
</div>
);
}

export default AddCandidate;

```

Update Candidate.js

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "../App2.css";

function UpdateCandidate() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const navigate = useNavigate();
  useEffect(() => {
    axios
      .get("http://localhost:5000/candidate/" + id)
      .then((response) => setData(response.data))
      .catch((err) => console.log(err));
  }, []);

  let handleSubmit = (e) => {
    e.preventDefault();
    axios.put("http://localhost:5000/candidate/" + id, data).then((res) => {
      alert("User Updated Successfully");
      navigate("/viewcandidate");
    });
  };

  return (
    <div>
      <div
        id="edit2"
        className="d-flex w-100 vh-100 justify-content-center align-items-center "
      >
        <div className="w-50 border bg-light p-5">
          <form onSubmit={handleSubmit}>
            <h1>UPDATE PATIENTS DATA'S</h1>
            <div>
              <lable htmlFor="id">ID :</lable>
              <input
                type="text"
                disabled
                name="id"
                className="form-control"
                value={data.id}
                onChange={(e) => setData({ ...data, candidateName: e.target.value })}
              />
            </div>
          </form>
        </div>
      </div>
    </div>
  );
}
```

```
<div>
  <label htmlFor="candidatename">Candidate Name :</label>
  <input
    type="text"
    name="candidatename"
    className="form-control"
    onChange={(e) =>
      setData({ ...data, candidateName: e.target.value })
    }
  />
</div>
```

```
<div>
  <label htmlFor="age">Age :</label>
  <input
    type="text"
    name="age"
    className="form-control"
    onChange={(e) =>
      setData({ ...data, Age: e.target.value })
    }
  />
</div>
```

```
<div>
  <label htmlFor="gender">Gender :</label>
  <input
    type="text"
    name="gender"
    className="form-control"
    onChange={(e) =>
      setData({ ...data, Gender: e.target.value })
    }
  />
</div>
```

```
<div>
  <label htmlFor="email">Email :</label>
  <input
    type="email"
    name="email"
    className="form-control"
    onChange={(e) =>
      setData({ ...data, Email: e.target.value })
    }
  />
</div>
```

```

        />
    </div>

    <div>
        <label htmlFor="partyname">Party Name :</label>
        <input
            type="text"
            name="partyname"
            className="form-control"
            onChange={e =>
                setData({ ...data, partyName: e.target.value })
            }
        />
    </div>

    <br />

    <button className="btn btn-info ">Update</button>
</form>
</div>
</div>
</div>
);
}

export default UpdateCandidate;

```

AddCampaign.js

```

import axios from "axios";
import React from "react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import "./App2.css";

function AddCampaign() {
    const [inputData, setInputData] = useState({
        campaignName: "",
        venueLocation: "",
        partyName: "",
        candidateName: "",
        startTime: "",
        endTime: "",
        email: "",
    });
}

```

```

const naviget = useNavigate();

let handleSubmit = (e) => {
  e.preventDefault();
  let result = validateValues(inputData);
  // setSubmitting(true);

  if (result === true) {
    axios
      .post("http://localhost:5000/campaign", inputData)
      .then((res) => {
        alert("Data added Successfully");
        naviget("/viewcampaign");
        console.log(res.data);
      })
      .catch((err) => console.log(err));
  } else {
    alert("Please Enter the Valid Inputs!!!");
  }
};

```

```

const validateValues = (inputData) => {
  if (inputData.campaignName.length === 0) {
    alert("Please enter Campaign Name !!! ");
    return false;
  } else if (inputData.venueLocation.length === 0) {
    alert("Please enter Venue Location !!!");
    return false;
  } else if (inputData.partyName.length === 0) {
    alert("Please enter Party Name !!!");
    return false;
  } else if (inputData.candidateName.length === 0) {
    alert("Please enter Candidate Name !!!");
    return false;
  } else if (inputData.startTime.length === 0) {
    alert("Please enter Start Time !!!");
    return false;
  } else if (inputData.endTime.length === 0) {
    alert("Please enter End Time !!!");
    return false;
  } else if (inputData.email.length === 0) {
    alert("Please enter Email !!!");
    return false;
  } else {
    return true;
  }
};

```



```
<div>
  <label htmlFor="candidatename">Candidate Name :</label>
  <input
    type="text"
    name="candidatename"
    className="form-control"
    onChange={(e) =>
      setInputData({ ...inputData, candidateName: e.target.value })
    }
  />
</div>

<div>
  <label htmlFor="starttime">Start Time :</label>
  <input
    type="text"
    name="starttime"
    className="form-control"
    onChange={(e) =>
      setInputData({ ...inputData, startTime: e.target.value })
    }
  />
</div>

<div>
  <label htmlFor="endtime">End Time :</label>
  <input
    type="text"
    name="endtime"
    className="form-control"
    onChange={(e) =>
      setInputData({ ...inputData, endTime: e.target.value })
    }
  />
</div>

<div>
  <label htmlFor="email">Email :</label>
  <input
    type="email"
    name="email"
    className="form-control"
    onChange={(e) =>
      setInputData({ ...inputData, email: e.target.value })
    }
  />
</div>
```



```

        />
    </div>

    <br />

    <button className="btn btn-info ">Submit</button>
</form>

</div>
</div>
);
}

export default AddCampaign;

```

UpdateCampaign.js

```

import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "./App2.css";

function UpdateCampaign() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const navigate = useNavigate();
  useEffect(() => {
    axios
      .get("http://localhost:5000/campaign/" + id)
      .then((response) => setData(response.data))
      .catch((err) => console.log(err));
  }, []);

  let handleSubmit = (e) => {
    e.preventDefault();
    axios.put("http://localhost:5000/campaign/" + id, data).then((res) => {
      alert("User Updated Successfully");
      navigate("/viewcampaign");
    });
  };

  return (
    <div>
      <div
        id="edit2"

```

```

className="d-flex w-100 vh-100 justify-content-center align-items-center "
>
<div className="w-50 border bg-light p-5">
  <form onSubmit={handleSubmit}>
    <h1>UPDATE CAMPAIGN DATA</h1>

    <div>
      <lable htmlFor="id">ID :</lable>
      <input
        type="text"
        disabled
        name="id"
        className="form-control"
        value={data.id}
        onChange={(e) => setData({ ...data, campaignName: e.target.value })}
      />
    </div>

    <div>
      <lable htmlFor="campaignname">Campaign Name :</lable>
      <input
        type="text"
        name="campaignname"
        className="form-control"
        onChange={(e) =>
          setData({ ...data, campaignName: e.target.value })
        }
      />
    </div>

    <div>
      <lable htmlFor="venueLocation">Venue Location :</lable>
      <input
        type="text"
        name="venueLocation"
        className="form-control"
        onChange={(e) =>
          setData({ ...data, venueLocation: e.target.value })
        }
      />
    </div>

    <div>
      <lable htmlFor="partyname">Party Name :</lable>
      <input
        type="text"

```

```

        name="partyname"
        className="form-control"
        onChange={(e) =>
            setData({ ...data, partyName: e.target.value })
        }
    />
</div>

<div>
    <label htmlFor="candidatename">Candidate Name :</label>
    <input
        type="text"
        name="candidatename"
        className="form-control"
        onChange={(e) =>
            setData({ ...data, candidateName: e.target.value })
        }
    />
</div>

<div>
    <label htmlFor="starttime">Start Time :</label>
    <input
        type="text"
        name="starttime"
        className="form-control"
        onChange={(e) =>
            setData({ ...data, startTime: e.target.value })
        }
    />
</div>

<div>
    <label htmlFor="endtime">End Time :</label>
    <input
        type="text"
        name="endtime"
        className="form-control"
        onChange={(e) =>
            setData({ ...data, endTime: e.target.value })
        }
    />
</div>

<div>
    <label htmlFor="email">Email :</label>

```

```

        <input
          type="email"
          name="email"
          className="form-control"
          onChange={e =>
            setData({ ...data, email: e.target.value })
          }
        />
      </div>
      <br />

      <button className="btn btn-info ">Update</button>
    </form>
  </div>
</div>
</div>
);
}

export default UpdateCampaign;

```

Home.js

```

import { Container, Nav, Navbar, NavDropdown } from 'react-bootstrap';
import { Link } from 'react-router-dom';

import React from 'react'

function Home() {
  return (
    <div>
      <Navbar expand="lg" className="bg-body-tertiary">
        <Container>
          <Navbar.Brand href="#home">Online Campaign Management System</Navbar.Brand>
          <Navbar.Toggle aria-controls="basic-navbar-nav" />
          <Navbar.Collapse id="basic-navbar-nav">
            <Nav className="me-auto">
              <NavDropdown title="Options" id="basic-nav-dropdown">
                <NavDropdown.Item href="action/3.1"><Link
to="/viewcampaign">Campaign Details</Link></NavDropdown.Item>
                <NavDropdown.Divider />
                <NavDropdown.Item href="action/3.2"><Link
to="/viewcandidate">Candidate Details</Link></NavDropdown.Item>
              </NavDropdown>
            </Nav>
          </Navbar.Collapse>
        </Container>
      </Navbar>
    </div>
  );
}

```

```

        </Container>
      </Navbar>
    </div>
  )
}

export default Home

```

db.json

```

{
  "campaign": [
    {
      "id": "2cf6",
      "campaignName": "Abki Baar Modi Sarkar",
      "venueLocation": "Periyar, Madurai",
      "partyName": "BJP",
      "candidateName": "Radhika Sarathkumar",
      "startTime": "9:30 A.M",
      "endTime": "12:00 P.M",
      "email": "radhikas123@gmail.com"
    },
    {
      "id": "3dg7",
      "campaignName": "Garibi Hatao",
      "venueLocation": "Ponmeni, Madurai",
      "partyName": "Congress",
      "candidateName": "Kamal Haasan",
      "startTime": "11:30 A.M",
      "endTime": "2:00 P.M",
      "email": "aandavar111@gmail.com"
    },
    {
      "id": "4eh8",
      "campaignName": "Udhikkum Suriyan",
      "venueLocation": "Periyar, Madurai",
      "partyName": "DMK",
      "candidateName": "Udhayanidhi Stalin",
      "startTime": "4:30 P.M",
      "endTime": "7:00 P.M",
      "email": "udhayas222@gmail.com"
    },
    {
      "id": "08a9",
      "campaignName": "Sarkar",
      "venueLocation": "Koyambedu",

```

```
    "partyName": "TVK",
    "candidateName": "Joseph Vijay",
    "startTime": "12:00 P.M",
    "endTime": "3:00 P.M.",
    "email": "VijayJ68@gmail.com"
  }
],
"candidate": [
  {
    "id": "4ab7",
    "candidateName": "Radhika Sarathkumar",
    "Age": "53",
    "Gender": "Female",
    "Email": "radhikas123@gmail.com",
    "partyName": "BJP"
  },
  {
    "id": "5bc8",
    "candidateName": "Kamal Haasan",
    "Age": "68",
    "Gender": "Male",
    "Email": "aandavar1111gmail.com",
    "partyName": "Congress"
  },
  {
    "id": "6cd7",
    "candidateName": "Udhayanidhi Stalin",
    "Age": "37",
    "Gender": "Male",
    "email": "udhayas222@gmail.com",
    "partyName": "DMK"
  }
]
}
```