

# REACT-BASED APP DEVELOPMENT

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <link href='https://fonts.googleapis.com/css?family=Poppins'
rel='stylesheet'>
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the
      build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
```

```
</body>
</html>
```

# COMPONENTS

## APPLYJOBS

```
import React, { useState } from 'react';
import './index.css';
import { useNavigate } from 'react-router-dom';

const ApplyJobs = () => {
  const navigate=useNavigate();
  const [name,setName]=useState("")
  const OnformSubmit=()=>{
    if(name==="" ){
      alert("Please fill the name first")
    }
  }
  else{
    alert("Your Job Application has been Applied Successfully")
    navigate("/Jobs")
  }
}

return (
  <div className="apply-job">
    <div className="container">
      <header className="header">
        <h1 className="post-job">Fill the form </h1>
      </header>
      <form>
        <div className="form-group">
          <label id="name-label" for="name">
            Enter Your Name
          </label>
          <input
            type="text"
            name="name"
            value={name}
            className="form-control"
            placeholder="Enter Your Name"
            onChange={(e) => setName(e.target.value)}
            required
          />
        </div>
        <div className="form-group">
```

```

        <label>
          Upload Your Resume
        </label>
        <label>
          <input type="file" id="myFile" name="filename" required />
        </label>
      </div>
      <div className="form-group">
        <button
          type="submit"
          className="submit-button"
          onClick={OnformSubmit}
        >
          Submit
        </button>
      </div>
    </form>
  </div>
</div>
);
}

export default ApplyJobs;

```

## DISCUSSION

```

import { useState, useRef, useEffect } from "react";
import "./index.css";
import Navbar from "../Navbar";
import StarRating from "../rating";
import CommentCard from "../commentCard";
import Editor from "../editor";
import { BiUserCircle } from "react-icons/bi";
import sanityClient from "../../client";
import axios from "axios";

var imageUrlRegex = /\b(https?:\/\/[^\s]+)/g;

const config = {
  url: `${process.env.REACT_APP_BASE_URL}`,
  method: "post",
  headers: {
    "Content-type": "application/json",
    Authorization: `Bearer ${process.env.REACT_APP_BEARER_TOKEN}`,
  },
};

export default function Discussion() {

```

```

const [name, setName] = useState("");
const [rating, setRating] = useState(0);
const [imageUrl, setImageUrl] = useState("");
const commentRef = useRef();
const [commentList, setCommentList] = useState(null);
const [urllist, setUrllist] = useState([]);

const handleSubmit = (e) => {
  e.preventDefault();
  if (name === "" || commentRef.length === "") {
    window.alert("Enter the credentials");
  } else {
    axios({
      ...config,
      data: {
        mutations: [
          {
            create: {
              name,
              content: commentRef.current.innerText,
              rating,
              _type: "Comment",
            },
          },
        ],
      },
    })
      .then((result) => console.log(result))
      .catch((error) => console.error(error));

    const newComment = commentList.unshift({
      name,
      content: commentRef.current.innerText,
      rating,
      _type: "Comment",
    });
    setName("");
    setImageUrl("");
    commentRef.current.innerText = "";
    setCommentList(...newComment, ...commentList);
  }
};

const handleNameChange = (e) => {
  setName(e.target.value);
};

const handleCommentChange = (e) => {
  const value = e.currentTarget.innerText;

```

```

const newLength = value?.length;
if (newLength === 0) {
  setUrlList([]);
  setImageUrl();
  return;
}
if (value?.match(imageUrlRegex)) {
  setUrlList(value?.match(imageUrlRegex));
} else {
  setUrlList([]);
}
};

useEffect(() => {
  if (urlList?.length !== 0) {
    setImageUrl(urlList[0]);
  } else {
    setImageUrl("");
  }
}, [urlList]);

const deleteComment = (id) => {
  axios({
    ...config,
    data: {
      mutations: [
        {
          delete: {
            id: id,
          },
        },
      ],
    },
  })
    .then((result) => console.log(result))
    .catch((error) => console.error(error));
  let removeData = [...commentList];
  removeData.splice(id, 1);
  setCommentList(removeData);
};

useEffect(() => {
  sanityClient
    .fetch(
      `*[_type == "Comment"]{
        _id,
        name,
        content,
        rating,
      }`
    )
    .then((result) => {
      setCommentList(result);
    })
    .catch((error) => console.error(error));
}, []);

```

```

        hexCode,
      } | order(_createdAt desc)`
    )
    .then((commentList) => {
      setCommentList(commentList);
    })
    .catch(console.error);
  }, []);
  return (
    <>
      <Navbar />
      <div className="discussion">
        <div className="right">
          <div className="discuss">JOB DISQUS</div>

          <form className="comment-form">
            <div className="input-group">
              <div className="comment-name">
                <div className="line"> {commentList?.length} Comments</div>
                <div className="icon-name">
                  <div className="user-icon">
                    <BiUserCircle />
                  </div>
                  <input
                    type="text"
                    placeholder="Your name"
                    className="form-control"
                    value={name}
                    onChange={handleNameChange}
                  />
                </div>
              </div>
            </div>
          </div>
          <div className="star-rating">
            <div className="rating-text">You rated this </div>
            <StarRating onChange={setRating} />
          </div>
          <div className="input-group">
            <div
              type="text"
              className="form-control comment"
              contentEditable="true"
              data-testid="comment-section-test"
              data-placeholder="Join the discussion...."
              onInput={handleCommentChange}
              ref={commentRef}
            />
            <div className="icon-button">

```

```

        <Editor onChange={setUrlList} commentRef={commentRef} />

        <button
          className="submit"
          data-testid="reset-comment"
          onClick={handleSubmit}
        >
          Comment
        </button>
      </div>
      <div className="only-image">
        {imageUrl && (
          <img
            src={imageUrl}
            data-testid="image-preview"
            className={` ${imageUrl ? "image-text-editor" : ""} `}
          />
        )}
      </div>
    </div>
  </form>
</div>
</div>
<div className="comment-section">
  <div className="comment-box">
    {commentList &&
      commentList?.map((comments) => {
        return (
          <CommentCard
            key={comments._id}
            value={comments}
            deleteComment={() => deleteComment(comments._id)}
          />
        );
      })}
  </div>
</div>
</>
);
}

```

ERROR PAGE

```

import React from 'react'
import './index.css';
import { Link } from 'react-router-dom';

```

```

const ErrorPage = () => {
  return (
    <>
      <div className="error-page">
        <div class="error">404</div>
        <br />
        <br />
        <span class="info">File not found</span>
        
        <div className="home-button">
          <Link to="/Home">Go To Home</Link>
        </div>
      </div>
    </>
  );
}

export default ErrorPage;

```

## FILTER

```

import { useState } from "react";
import "./index.css";
const experience = [
  { min: 0, max: 1 },
  { min: 2, max: 3 },
  { min: 4, max: 5 },
  { min: 5, max: 10 },
];

const Filter = ({
  setFilteredJobs,
  handleJobFilter,
  handleExperienceFilter,
  searchEvent,
}) => {
  const [checkedState, setCheckedState] = useState(
    new Array(experience.length).fill(false)
  );

  const handleOnChange = (position) => {
    const updatedCheckedState = checkedState.map((item, index) =>
      index === position ? !item : item
    );
  };

```



```

    );

    setCheckedState(updatedCheckedState);
    handleExperienceFilter(updatedCheckedState);
  };
  return (
    <>
      <div className="filter-page">
        <div className="search-box">
          <div className="search">
            <h3>Search Jobs</h3>
            <div className="job-search">
              <input
                type="text"
                className="search-term"
                placeholder="Search Here"
                onChange={searchEvent}
              />
            </div>
          </div>
        </div>
        <div className="filter">
          <div className="job-category">
            <h4>Categories</h4>
            <ul>
              <li onClick={handleJobFilter}>Frontend</li>
              <li onClick={handleJobFilter}>Backend</li>
              <li onClick={handleJobFilter}>Devops</li>
              <li onClick={handleJobFilter}>Full Stack</li>
              <li onClick={handleJobFilter}>Digital Marketing</li>
            </ul>
          </div>

          <div className="job-category">
            <h4>Experience</h4>
            <ul className="checkbox">
              <li>
                <input
                  name="0-1"
                  type="checkbox"
                  checked={checkedState[0]}
                  onChange={() => handleOnChange(0)}
                />
                0-1 year
              </li>
              <li>
                <input
                  name="2-3"
                  type="checkbox"

```

```

        checked={checkedState[1]}
        onChange={() => handleOnChange(1)}
      />
      2-3 year
    </li>
    <li>
      <input
        name="4-5"
        type="checkbox"
        checked={checkedState[2]}
        onChange={() => handleOnChange(2)}
      />
      4-5 year
    </li>
    <li>
      <input
        name="4-5"
        type="checkbox"
        checked={checkedState[3]}
        onChange={() => handleOnChange(3)}
      />
      5+ year
    </li>
  </ul>
</div>
</div>
</div>
</div>
</div>
);
};

export default Filter;

```

HOME

```

import React from "react";
import Navbar from "../Navbar";
import { Link } from "react-router-dom";
import "../index.css";

const Home = () => {
  return (
    <>
      <Navbar />
      <div className="banner-img">
        <div className="title">
          <h3>

```

```

        Find the <span>Right Job</span> In the
        <br />
        <span> Right Companies</span>
    </h3>
    <div className="small-tagline">
        <p>Got fired...!! Get Ready to be hired</p>
    </div>
</div>
<div className="button" data-testid="btn">
    <Link to="/Jobs">Browse Jobs</Link>
</div>
</div>
<div className="social-media" data-testid="images">
    
    
    
    
</div>
</>
);
};

export default Home;

```

## JOBS

```

import React, { useState } from "react";
import Navbar from "../Navbar";
import { Link } from "react-router-dom";
import "./index.css";
import Job from "../../Assets/jobs.json";
import Filter from "../Filter";
import { AiOutlineHeart, AiFillHeart } from "react-icons/ai";

const experience = [
    { min: 0, max: 1 },
    { min: 2, max: 3 },
    { min: 4, max: 5 },
    { min: 5, max: 10 },
];

```

```

const Jobs = () => {
  const JobData = JSON.parse(localStorage.getItem("item")) || [];
  const [filteredJobs, setFilteredJobs] = useState([...JobData, ...Job]);
  const [searchterm, setSearchTerm] = useState("");
  const [active, setActive] = useState(false);
  function handleJobFilter(event) {
    const value = event.target.innerText;
    event.preventDefault();
    setFilteredJobs(
      Job.filter((job) => {
        return job.role === value;
      })
    );
  }
  function saveClick(id, logo, company, position, location, posted) {
    window.localStorage.setItem(
      "Job",
      JSON.stringify(id, logo, company, position, location, posted)
    );
    console.log(JobData);
  }
  const searchEvent = (event) => {
    const data = event.target.value;
    setSearchTerm(data);
    if (searchterm !== "" || searchterm.length > 2) {
      const filterData = Job.filter((item) => {
        if (item) {
          return Object.values(item)
            .join("")
            .toLowerCase()
            .includes(searchterm.toLowerCase());
        } else {
          return 0;
        }
      });
      setFilteredJobs(filterData);
    } else {
      setFilteredJobs(Job);
    }
  };
  function handleExperienceFilter(checkedState) {
    let filters = [];
    checkedState.forEach((item, index) => {
      if (item === true) {
        const filterS = Job.filter((job) => {
          return (
            job.experience >= experience[index].min &&

```



```

        <Link
          to="/Jobs"
          onClick={() => {
            saveClick(
              {
                id,
                logo,
                company,
                position,
                location,
                posted,
              },
              setActive(!active)
            );
          }}
        >
          {JSON.parse(localStorage.getItem("Job"))?.id ===
            id ? (
              <AiFillHeart />
            ) : (
              <AiOutlineHeart />
            )}
        </Link>
      </div>
    </div>
  </div>
</div>
);
}
})}
</div>

<Filter
  setFilteredJobs={setFilteredJobs}
  handleJobFilter={handleJobFilter}
  handleExperienceFilter={handleExperienceFilter}
  searchEvent={searchEvent}
/>
</div>
</div>
</>
);
};

export default Jobs;

```

NAVBAR

```

import React from "react";
import { Link } from "react-router-dom";
import "./index.css";

const Navbar = () => {
  return (
    <>
      <div className="main-page">
        <nav id="navbar">
          <h1 className="logo">
            Job<span>Hunt</span>
          </h1>

          <ul>
            <li>
              <Link to="/home">Home</Link>
            </li>
            <li>
              <Link to="/jobs">Jobs</Link>
            </li>
            <li>
              <Link to="/post-job">Post Job</Link>
            </li>
            <li>
              <Link to="/saved-job">Saved Job</Link>
            </li>
          </ul>
        </nav>
      </div>
    </>
  );
};

export default Navbar;

```

## POSTJOB

```

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

const PostJob = () => {
  const [company, setCompany] = useState("");
  const [logo, setLogo] = useState("");
  const [position, setPosition] = useState("");

```

```

const [salary, setSalary] = useState("");
const [experience, setExperience] = useState("");
const [role, setRole] = useState("");
const [location, setLocation] = useState("");

const navigate=useNavigate();
const getBase64 = (file) => {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.onload = () => resolve(reader.result);
    reader.onabort = (error) => reject(error);
    reader.readAsDataURL(file);
  });
};
const handleImg = (e) => {
  const file = e.target.files[0];
  getBase64(file).then((base64) => {
    localStorage["logo"] = base64;
    setLogo(base64);
  });
};

const handleSubmitButton = (e) => {
  const jobPost = {
    company,
    position,
    salary,
    experience,
    role,
    location,
    logo
  };
  e.preventDefault();
  if (company === "") {
    window.alert("Enter name");
  } else if (position === "") {
    window.alert("Enter position");
  } else if (experience === "") {
    window.alert("Enter Experience");
  } else if (salary === "") window.alert("Enter Salary");
  else {
    let savedItem = [];
    if (localStorage.getItem("item")) {
      savedItem = JSON.parse(localStorage.getItem("item"));
    }
    localStorage.setItem("item", JSON.stringify([...savedItem, {jobPost}]));
    window.alert("Form Submitted Successfully");
    navigate("/Jobs");
  }
};

```



```

    }
  };
  return (
    <div>
      <Navbar />

      <div className="job-background">
        <div className="title">
          <h2>Post a Job</h2>
        </div>
      </div>
      <div className="container">
        <header className="header">
          <h1 className="post-job">Fill the form </h1>
        </header>
        <form>
          <div className="form-group">
            <label id="name-label" htmlFor="name">
              Company Name
            </label>
            <input
              type="text"
              name="name"
              className="form-control"
              placeholder="Enter Company Name"
              onChange={(e) => setCompany(e.target.value)}
              required
            />
          </div>
          <div className="form-group">
            <label id="name-label" htmlFor="name">
              Enter Job Location
            </label>
            <input
              type="text"
              name="name"
              className="form-control"
              placeholder="Enter Job Location"
              onChange={(e) => setLocation(e.target.value)}
              required
            />
          </div>
          <div className="form-group">
            <label id="logo-label" htmlFor="logo">
              Company logo
            </label>
            <label>
              <input

```

```

        type="file"
        id="myFile"
        name="filename"
        onChange={handleImg}
        required
      />
    </label>
  </div>
  <div className="form-group">
    <label>What position are you posting for?</label>
    <select
      id="dropdown"
      name="role"
      className="form-control"
      onChange={(e) => setPosition(e.target.value)}
      required
    >
      <option disabled selected value>
        Select position
      </option>
      <option>Frontend</option>
      <option>Backend</option>
      <option>Full Stack</option>
      <option>Devops</option>
      <option>Digital Marketing</option>
    </select>
  </div>
  <div className="form-group">
    <label id="name-label" htmlFor="name">
      Enter Job Role
    </label>
    <input
      type="text"
      name="name"
      className="form-control"
      placeholder="Enter Job Role"
      onChange={(e) => setRole(e.target.value)}
      required
    />
  </div>

  <div
    className="form-group"
    onChange={(e) => setExperience(e.target.value)}
  >
    <label>Experience </label>
    <label>
      <input

```

```

        name="user-recommend"
        value="0-1 Year"
        type="radio"
        className="input-radio"
    />
    0-1 Year
</label>
<label>
    <input
        name="user-recommend"
        value=" 2-3 Years"
        type="radio"
        className="input-radio"
    />
    2-3 Years
</label>
<label>
    <input
        name="user-recommend"
        value=" 4-5 Years"
        type="radio"
        className="input-radio"
    />
    4-5 Years
</label>
<label>
    <input
        name="user-recommend"
        value="5+ Years"
        type="radio"
        className="input-radio"
    />
    5+ Years
</label>
</div>

<div className="form-group">
    <label>Salary</label>
    <select
        className="form-control"
        onChange={(e) => setSalary(e.target.value)}
        required
    >
        <option disabled selected value>
            Select Salary
        </option>
        <option value="0-15K">0-15K</option>
        <option value="15-30K">15-30K</option>
    </select>

```

```

      <option value="30K-50K">30K-50K</option>
      <option value="50K-80K">50K-80K</option>
      <option value="80K+">80K+</option>
    </select>
  </div>
  <div className="form-group">
    <button type="submit" className="submit-button"
onClick={handleSubmit}>
      Submit
    </button>
  </div>
</form>
</div>
</div>
);
};

export default PostJob;
```

SAVEJOBS

```
import React, { useState } from "react";
import Navbar from "../Navbar";

const SaveJobs = () => {
  const jobs = [JSON.parse(localStorage.getItem("Job"))];
  return (
    <div>
      <Navbar />
      <div className="jobs-for-you">
        <div className="job-background">
          <div className="title">
            <h2>Saved Jobs</h2>
          </div>
        </div>
        <div className="job-section">
          <div className="job-page">
            {jobs.map(({ logo, company, position, location, role }) => {
              return (
                <div className="job-list">
                  <div className="job-card">
                    <div className="job-name">
                      <img
                        src={require(`../../Assets/images/${logo}`)}
                        alt="logo"
                        className="job-profile"
                      />
                    <div className="job-detail">
```

```

        <h4>{company}</h4>
        <h3>{position}</h3>
        <div className="category">
          <p>{location}</p>
          <p>{role}</p>
        </div>
      </div>
    </div>
    <div className="job-posting">
    </div>
  </div>
</div>
  );
  })}
</div>
</div>
</div>
  );
};

export default SaveJobs;

```



