## **Backend**

## server/routes/user.js

Roll Numbers: 39 & 41

```
const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");
const upload = require("../utils/multer");
const User = require("../model/user");
router.post("/", upload.single("image"), async (req, res) => {
 try {
  // Upload image to cloudinary
  const result = await cloudinary.uploader.upload(req.file.path);
  // Create new user
  let user = new User({
    name: req.body.name,
   avatar: result.secure url,
    cloudinary_id: result.public_id,
  });
  // Save user
  await user.save();
  res.json(user);
 } catch (err) {
  console.log(err);
}
});
router.get("/", async (req, res) => {
 try {
  let user = await User.find();
  res.json(user);
 } catch (err) {
  console.log(err);
});
router.delete("/:id", async (req, res) => {
 try {
  // Find user by id
  let user = await User.findById(req.params.id);
  // Check if the user is found
  if (!user) {
   return res
     .status(404)
     .json({ success: false, message: "User not found" });
```

```
}
  // Delete image from cloudinary
  await cloudinary.uploader.destroy(user.cloudinary_id);
  // Delete user from db
  await user.deleteOne();
  res.json(user);
 } catch (err) {
  console.log(err);
  res.status(500).json({ success: false, message: "Internal Server Error" });
}
});
router.put("/:id", upload.single("image"), async (req, res) => {
 try {
  let user = await User.findById(req.params.id);
  // Delete image from cloudinary
  // Check if the user is found
  if (!user) {
   return res
     .status(404)
     .json({ success: false, message: "User not found" });
  }
  await cloudinary.uploader.destroy(user.cloudinary_id);
  // Upload image to cloudinary
  let result;
  if (req.file) {
   result = await cloudinary.uploader.upload(reg.file.path);
  const data = {
    name: req.body.name || user.name,
    avatar: result?.secure url || user.avatar,
    cloudinary_id: result?.public_id || user.cloudinary_id,
  };
  user = await User.findByIdAndUpdate(req.params.id, data, { new: true });
  res.json(user);
 } catch (err) {
  console.log(err);
});
```

```
router.get("/:id", async (req, res) => {
 try {
  // Find user by id
  let user = await User.findById(req.params.id);
  // Check if the user is found
  if (!user) {
   return res
     .status(404)
     .json({ success: false, message: "User not found" });
  }
  res.json(user);
 } catch (err) {
  console.log(err);
});
module.exports = router;
server/model/user.js
const mongoose = require("mongoose");
const userSchema = new mongoose.Schema({
 name: {
  type: String,
 },
 avatar: {
  type: String,
 },
 cloudinary_id: {
  type: String,
},
});
module.exports = mongoose.model("Cloudinary-CRUD", userSchema);
server/Dockerfile
# Select a base image
FROM node:20-alpine3.17
# Create a directory and go to directory
WORKDIR /app
# Copy the package.json file to my current directory for install necessary dependence
```

```
COPY package.json.
# Install the dependence
RUN npm install
# Copy other file to my current directory
COPY ...
# Open the port for express server
EXPOSE 5000
# Run express rum in foreground
CMD ["npm", "start"]
Frontend
client/src/pages/AddUser.js
import { useState } from "react";
import { useNavigate } from "react-router-dom";
const AddUser = () => {
 const navigate = useNavigate();
 const [data, setData] = useState({
  name: "",
  image: "",
 });
 const handleChange = (name) => (e) => {
  const value = name === "image" ? e.target.files[0] : e.target.value;
  setData({ ...data, [name]: value });
 const handleSubmit = async () => {
  try {
   let formData = new FormData();
   formData.append("image", data.image);
   formData.append("name", data.name);
   const res = await fetch(`${process.env.REACT_APP_SERVER_URL}/user`, {
    method: "POST",
    body: formData,
   });
   if (res.ok) {
    setData({ name: "", image: "" });
    navigate("/", { replace: true });
   }
```

```
} catch (error) {
   console.log(error);
 };
 return (
  <div style={{ maxWidth: 500, margin: "auto" }}>
   <div className="mb-3">
     <input
      className="form-control"
      placeholder="Enter name"
      type="text"
      name="name"
      value={data.name}
      onChange={handleChange("name")}
    />
   </div>
   <div className="mb-3">
     <input
      className="form-control"
      type="file"
      accept="image/*"
      name="image"
      onChange={handleChange("image")}
    />
   </div>
   <div className="text-center">
     <button className="btn btn-primary" onClick={handleSubmit}>
      Submit
     </button>
   </div>
  </div>
 );
};
export default AddUser;
client/src/pages/EditUser.is
import { useEffect, useState } from "react";
import { useNavigate, useLocation } from "react-router-dom";
const EditUser = ({ match }) => {
 const navigate = useNavigate();
 const { pathname } = useLocation();
```

```
const userid = pathname.substring(pathname.lastIndexOf("/") + 1);
const [data, setData] = useState({
 name: "",
 image: "",
});
useEffect(() => {
 fetch(`${process.env.REACT_APP_SERVER_URL}/user/${userid}`)
  .then((res) => res.json())
  .then((data) => setData(data));
}, []);
const handleChange = (name) => (e) => {
 const value = name === "image" ? e?.target?.files[0] : e?.target?.value;
 setData({ ...data, [name]: value });
};
const handleSubmit = async () => {
 try {
  let formData = new FormData();
  formData.append("image", data?.image);
  formData.append("name", data?.name);
  const res = await fetch(
    `${process.env.REACT_APP_SERVER_URL}/user/${userid}`,
    method: "PUT",
     body: formData,
   }
  );
  if (res.ok) {
   setData({ name: "", image: "" });
   navigate("/", { replace: true });
  }
 } catch (error) {
  console.log(error);
 }
};
return (
 <div style={{ maxWidth: 500, margin: "auto" }}>
  <div className="mb-3">
    <input
     className="form-control"
```

```
type="text"
      name="name"
      value={data.name}
      onChange={handleChange("name")}
    />
   </div>
   <div className="mb-3">
     <input
      className="form-control"
      type="file"
      accept="image/*"
      name="image"
      onChange={handleChange("image")}
    />
   </div>
   <div className="text-center">
     <button className="btn btn-primary" onClick={handleSubmit}>
      Update
     </button>
   </div>
  </div>
 );
};
export default EditUser;
client/src/pages/Home.is
import { Link } from "react-router-dom";
import { useEffect, useState } from "react";
const Home = () => {
 const [users, setUsers] = useState();
 useEffect(() => {
  const fetchUsers = async () => {
   const res = await fetch(`${process.env.REACT_APP_SERVER_URL}/user`);
   const data = await res.json();
   setUsers(data);
  };
  fetchUsers();
 }, []);
 const handleDelete = async (id) => {
  try {
   const res = await fetch(
```

```
`${process.env.REACT_APP_SERVER_URL}/user/${id}`,
      method: "DELETE",
    }
   );
   if (res.ok) {
     const updatedUsers = users.filter((user) => user._id !== id);
     setUsers(updatedUsers);
  } catch (error) {
   console.log(error);
  }
 };
 return (
  <div className="row center">
   \{users?.map((user) => (
     <div className="card me-3 mt-2 p-0" key={user._id}>
      <img src={user.avatar} alt="" width={"100%"} height={200} />
      <div className="p-4">
       <h4 className="text-center">{user.name}</h4>
       <div className="d-flex justify-content-between align-items-center">
        <Link
          to={\'/edit/\${user._id}\'\}
          style={{ textDecoration: "none" }}
          className="edit-button"
        >
          Edit
        </Link>
        <but
         className="btn btn-danger btn-sm"
         onClick={() => handleDelete(user._id)}
        >
          Χ
        </button>
       </div>
      </div>
     </div>
   ))}
  </div>
 );
};
export default Home;
```

client/Dockerfile
# Stage-1 & spacify a name 'builder'
FROM node:latest AS builder

# Create a directory and go to directory WORKDIR /app

# Copy the package.json file to my current directory for install necessary dependence COPY package.json .

# Install the dependence RUN npm install

# Copy other file to my current directory COPY . .

# Build optimise static file RUN npm run build

# Stage-2

FROM nginx:1.25.2-alpine-slim

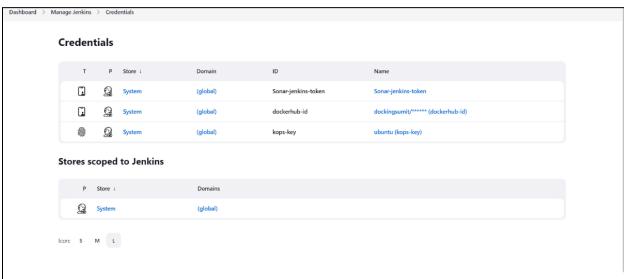
# Copy static file to my nignx folder for serve static contain COPY --from=builder /app/build /usr/share/nginx/html

# Open the port for react EXPOSE 80

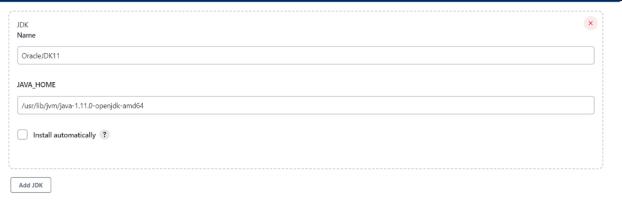
# Run nginx in foreground CMD ["nginx", "-g", "daemon off;"]

Screenshots:

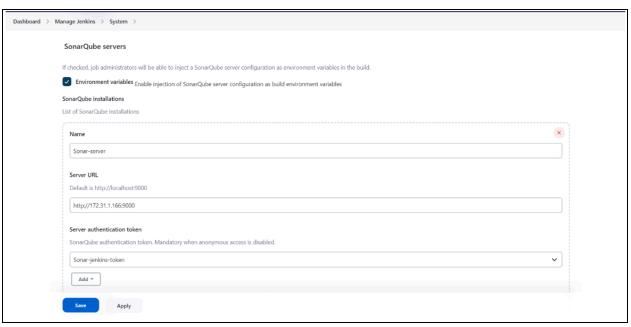


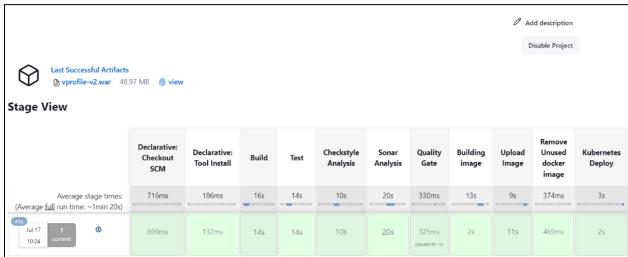


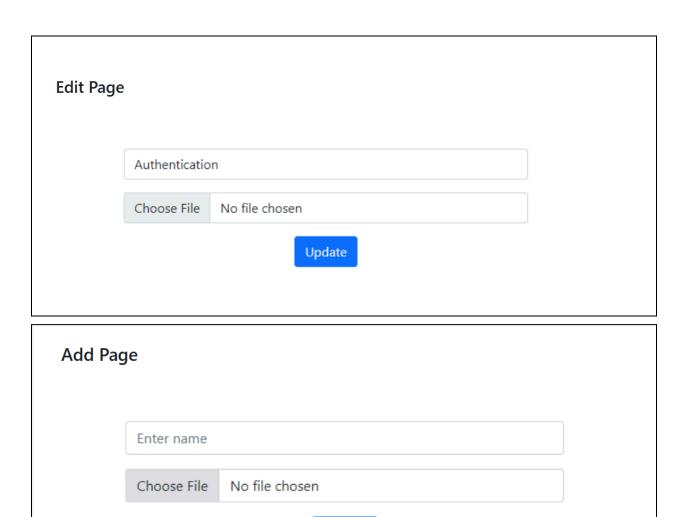












Submit