**PRACTICE REPORT**

**Subject: Web application**

**Lab 4: React**

*Instructor: Trần Vĩnh Khiêm*

*Report date: 20/12/2023*

1. **GENERAL INFORMATION:**

Class: MSIS207.O12.CTTT.1

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Student ID** | **Email** |
| 1 | Nguyễn Văn Trường Khoa | 20521472 | [20521472@gm.uit.edu.vn](mailto:20521472@gm.uit.edu.vn) |
| 2 |  |  |  |

## DETAILED REPORT

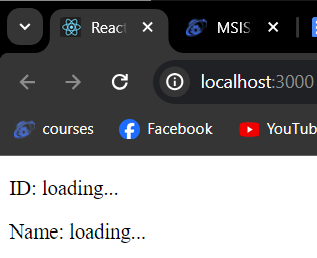
**Link Github contain code:**

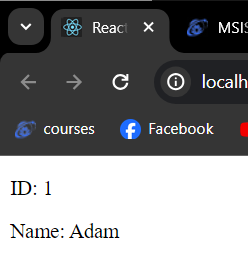
<https://github.com/KRocker3512/Web-app-practice-class>

**Note** : If you can’t access link above, please contact to me via email :20521472@gm.uit.edu.vn .

**Exercise 1.1:**

Demo:



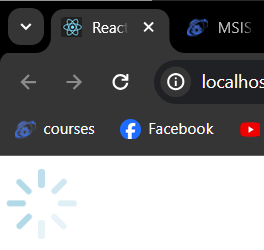


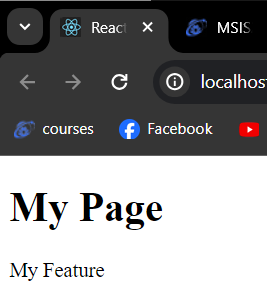
Code:

|  |  |
| --- | --- |
|  |  |

**Exercise 1.2:**

Demo:





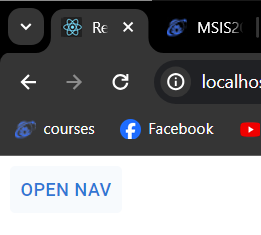
Code:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**Exercise 1.3:**

Demo:



A screenshot of a computer

Description automatically generated

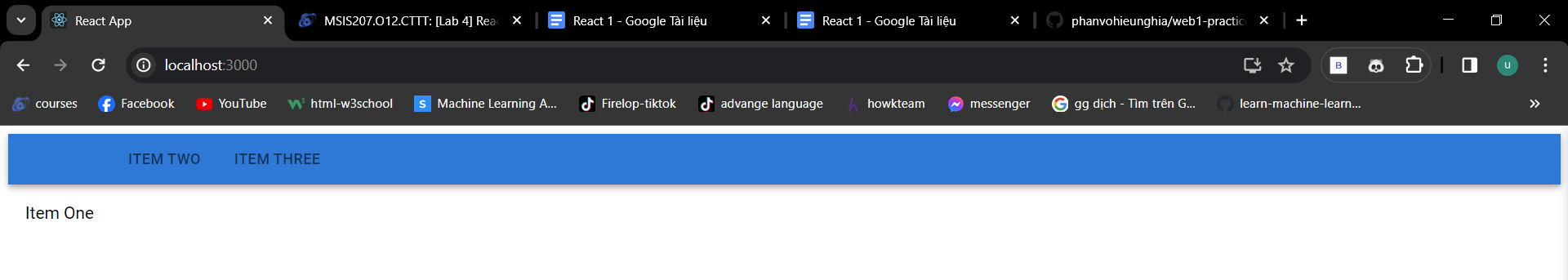
Code:

|  |
| --- |
| import "typeface-roboto";  import React, { useState } from "react"; import Drawer from "@mui/material/Drawer"; import Button from "@mui/material/Button";  import List from "@mui/material/List";  import ListItem from "@mui/material/ListItem";  import ListItemText from "@mui/material/ListItemText";  import { BrowserRouter as Router, Route, Switch, Link } from "react-router-dom";  import First from "./First";  import Second from "./Second";  import Third from "./Third";  export default function App({ links }) {      const [open, setOpen] = useState(false);      function toggleDrawer({ type, key}){          if (type === "keydown" && (key = "Tab" || key ==="Shift")) {              return;          }          setOpen (!open);      }  return (  <Router>      <Button onClick={toggleDrawer}>Open Nav</Button>          <section>              <Route path="/first" component={First} />              <Route path="/second" component={Second} />              <Route path="/third" component={Third} />          </section>          <Drawer open={open} onClose={toggleDrawer}>          <Drawer open={open} onClose={toggleDrawer}>              <div                  style={{ width: 250 }}                  role="presentation"                  onClick={toggleDrawer}                  onkeyDown={toggleDrawer}              >                  <List>                      {links.map((link) => (                          <ListItem button key={link.url} component={Link} to={link.url}>                              <Switch>                                  <Route                                      exact                                      path={link.url}                                      render={() => (                                          <ListItemText                                              primary={link.name}                                              primaryTypographyProps={{ color: "primary" }}                                          />                                      )}                                  />                                  <Route                                      path="/"                                      render={() => <ListItemText primary={link.name} />}                                  />                                  </Switch>                              </ListItem>                          ))}                      </List>                  </div>              </Drawer>          </Drawer>      </Router>      );  }  App.defaultProps = {      links: [          { url: "/first", name: "First Page"},          { url: "/second", name: "Second Page"},          { url: "/third", name: "Third Page"},      ],  }; |

|  |
| --- |
|  |

**Exercise 1.3.1:**

Demo:



Code:

|  |
| --- |
| import "typeface-roboto";  import React from "react";  import { BrowserRouter as Router, Route, Link } from "react-router-dom";  import AppBar from "@mui/material/AppBar";  import Tabs from "@mui/material/Tabs";  import Tab from "@mui/material/Tab";  import Typography from "@mui/material/Typography";  const tabContentStyle = {      padding: 16,  };  function TabContainer({ value }) {      return (          <AppBar position="static">              <Tabs value={value}>                  <Tab label="Item One" component={Link} to="/" />                  <Tab label="Item Two" component={Link} to="/page2" />                  <Tab label="Item Three" component={Link} to="/page3" />              </Tabs>          </AppBar>      );  }  export default function App() {      return (          <Router>              <Route                  exact                  path="/"                  render={() => (                      <>                          <TabContainer value={0} />                          <Typography component="div" style={tabContentStyle}>                              Item One                          </Typography>                      </>                  )}              />              <Route                  exact                  path="/page2"                  render={() => (                      <>                          <TabContainer value={1} />                          <Typography component="div" style={tabContentStyle}>                              Item Two                          </Typography>                      </>                  )}              />              <Route                  exact                  path="/page3"                  render={() => (                      <>                          <TabContainer value={2} />                          <Typography component="div" style={tabContentStyle}>                              Item Three                          </Typography>                      </>                  )}              />          </Router>      );  } |

|  |
| --- |
| import App1\_3\_1 from './Exercise/AppEx1.3.1';  const root = ReactDOM.createRoot(document.getElementById('root'));  root.render(    <React.StrictMode>      <App1\_3\_1 />    </React.StrictMode>  );  reportWebVitals(); |

**Exercise 1.3.2:**

Demo:

Code:

**Exercise 1.4:**

Demo:

A screenshot of a computer

Description automatically generated

Code:

|  |  |
| --- | --- |
| import React from 'react'  import "typeface-roboto"  import FormGroup from "@mui/material/FormGroup"  import MyTextInput from '../Component/MyTextInput'  import MySelect from '../Component/MySelect'  export default function App() {    return (      <FormGroup style={{width: 200, margin: 10}}>          <MyTextInput/>          <MySelect/>      </FormGroup>    )  } | import React, { useState } from 'react'  import TextField from "@mui/material/TextField"  export default function MyTextInput() {      const [value, setValue] = useState("")    return (      <TextField          label="Name"          value={value}          onChange={(e) => setValue(e.target.value)}          margin="normal"      />    )  } |

|  |
| --- |
| import React, { useState } from 'react'  import InputLabel from "@mui/material/InputLabel"  import MenuItem from "@mui/material/MenuItem"  import FormControl from "@mui/material/FormControl"  import Select from "@mui/material/Select"  export default function MySelect() {      const [value, setValue] = useState("first")    return (      <FormControl>          <InputLabel htmlFor='my-select'>My Select</InputLabel>          <Select              value={value}              onChange={(e) => setValue(e.target.value)}              inputProps={{id: "my-select"}}          >              <MenuItem value="first">First</MenuItem>              <MenuItem value="second">Second</MenuItem>              <MenuItem value="third">Third</MenuItem>          </Select>      </FormControl>    )  } |

**Exercise 1.5:**

Demo:

A screenshot of a computer

Description automatically generated

Code:

|  |
| --- |
| import "typeface-roboto"  import React, { useState } from 'react'  import Button from "@mui/material/Button"  import Grid from "@mui/material/Grid"  import IconButton from "@mui/material/IconButton"  import AndroidIcon from "@mui/icons-material/Android"  const buttonStyle = {margin: 10}  function toggleColor(setter, value) {      setter(value === "default" ? "primary" : "default")  }  export default function App() {      const [contained, setContained] = useState("default")      const [text, setText] = useState("default")      const [outlined, setOutlined] = useState("default")      const [icon, setIcon] = useState("default")    return (      <Grid container>              <Grid                  item                  component={Button}                  variant="contained"                  style={buttonStyle}                  color={contained}                  onClick={() => toggleColor(setContained, contained)}              >                  Contained              </Grid>              <Grid                  item                  component={Button}                  style={buttonStyle}                  color={text}                  onClick={() => toggleColor(setText, text)}              >                  Text              </Grid>              <Grid                  item                  component={Button}                  variant="outlined"                  style={buttonStyle}                  color={outlined}                  onClick={() => toggleColor(setOutlined, outlined)}              >                  Outlined              </Grid>              <Grid                  item                  component={IconButton}                  style={buttonStyle}                  color={icon}                  onClick={() => toggleColor(setIcon, icon)}              >                  <AndroidIcon/>              </Grid>      </Grid>    )  } |

**Exercise 1.6:**

Demo:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Code:

|  |
| --- |
| import { createContext } from "react"  const FormContext = createContext({})  export default FormContext |

|  |
| --- |
| import SimpleForm from './SimpleForm'  import InputField from './InputField'  import { useState, useEffect } from 'react'  const FormExample0 = ({ onSubmit, onChange, initialValue = {}}) => {      const [formFields, setFormFields] = useState(initialValue)      const [valid, setValid] = useState(true)      const [errors, setErrors] = useState({})      useEffect(() => {          if(onChange) {              onChange(formFields, valid, errors)          }      }, [onChange, formFields, valid, errors])      return (          <div className="TheForm">              <h1>Single field</h1>              <SimpleForm                          value={formFields}                          onChange={setFormFields}                          onValid={(v, errs) => {                              setValid(v)                              setErrors(errs)                          }}              >                  <InputField                  name="field1"                  onValidate={(v) =>                      !v || v.length < 3 ? 'Too short!' : null                  }                  />                  <button                  onClick={() => onSubmit && onSubmit(formFields)}                  disabled={!valid}                  >                      Submit!                  </button>              </SimpleForm>          </div>      )  }  export default FormExample0 |

|  |
| --- |
| import { useState, useEffect } from 'react'  import SimpleForm from './SimpleForm'  import InputField from './InputField'  const FormExample1 = ({ onSubmit, onChange, initialValue = {}}) => {      const [formFields, setFormFields] = useState(initialValue)      const [valid, setValid] = useState(true)      const [errors, setErrors] = useState({})      useEffect(() => {          if(onChange) {              onChange(formFields, valid, errors)          }      }, [onChange, formFields, valid, errors])      return (          <div className="TheForm">              <h1>Mutiple fields</h1>              <SimpleForm              value={formFields}              onChange={setFormFields}              onValid={(v, errs) => {                  setValid(v)                  setErrors(errs)              }}              >                  <InputField                  name="address1"                  onValidate={(v) =>                      !v || v.length < 3 ? 'Too short!' : null                  }                  />                  <InputField                  name="address2"                  onValidate={(v) => (v ? null: 'Required')}                  />                  <InputField                  name="price"                  type="number"                  onValidate={(v) =>                      !v || parseInt(v) < 102 ? 'Must be at least 102!' : null                  }                  />                  <InputField                  name="requiredBy"                  type="date"                  onValidate={(v) => (v ? null: 'Required')}                  />                  <button                  onClick={() => onSubmit && onSubmit(formFields)}                  disabled={!valid}                  >                      Submit!                  </button>              </SimpleForm>          </div>      )  }  export default FormExample1 |

|  |
| --- |
| import { useState, useEffect, useContext } from "react"  import FormContext from "./FormContext"  const splitCamelCase = (s) =>      s        .replace(/([a-z0-9])([A-Z0-9])/g, "$1 $2")        .replace(/^([a-z])/, (x) => x.toUpperCase())  const InputField = (props) => {      const form = useContext(FormContext)      const [error, setError] = useState('')      const { onValidate, name, label, ...otherProps} = props      let value = form.value && form.value(name)      useEffect(() => {          if(onValidate) {              setError(onValidate(value))          }      }, [onValidate, value])      const setInvalid = form.setInvalid      useEffect(() => {          if(setInvalid){              setInvalid(name, error)          }      }, [setInvalid, name, error])      if(!form.value){          return "InputField should be wrapped in a for"      }    return (      <div className="InputField">          <label htmlFor={name}>{label || splitCamelCase(name)}:</label>          <input          id={name}          onBlur={() => form.setDirty(name)}          value={value || ''}          onChange={(event) => {              form.setDirty(name)              form.setValue(name, event.target.value)          }}          {...otherProps}           />{' '}           {              <div className="InputField-error">                  {form.isDirty(name) && error ? error : <>&nbsp;</>}              </div>           }      </div>    )  }  export default InputField |

|  |
| --- |
| const ShowData = ({formFields, valid, errors}) => (      <div className="ShowData">          <dl>              <dt>Current value:</dt>              <dd>{JSON.stringify(formFields, null, 2)}</dd>              <dt>Valid?</dt>              <dd>{JSON.stringify(valid)}</dd>              <dt>Errors?</dt>              <dd>{JSON.stringify(errors, null, 2)}</dd>          </dl>      </div>  )  export default ShowData |

|  |
| --- |
| import { useCallback, useState, useEffect } from "react"  import FormContext from "./FormContext"  const SimpleForm = ({ children, value, onChange, onValid}) => {      const [values, setValues] = useState(value || {})      const [dirtyFields, setDirtyFields] = useState({})      const [invalidFields, setInvalidFields] = useState({})      useEffect(() => {          setValues(value || {})      }, [value])      useEffect(() => {          if(onChange){              onChange(values)          }      }, [onChange, values])      useEffect(() => {          if(onValid){              onValid(                  Object.keys(invalidFields).every((i) => !invalidFields[i]),                  invalidFields              )          }      }, [onValid, invalidFields])      const setValue = useCallback(          (field, v) => setValues((vs) => ({...vs, [field]: v})),          [setValues]      )      const getValue = useCallback((field) => values[field], [values])      const setDirty = useCallback(          (field) => setInvalidFields((df) => ({...df, [field]:true})),          [setDirtyFields]      )      const getDirty = useCallback(          (field) => Object.keys(dirtyFields).includes(field),          [dirtyFields]      )      const setInvalid = useCallback(          (field, error) => {              setInvalidFields((i) => ({                  ...i,                  [field]: error ? error : undefined,              }))          },          [setInvalidFields]      )      const form = {          setValue: setValue,          value: getValue,          setDirty: setDirty,          isDirty: getDirty,          setInvalid: setInvalid,      }      return(          <div className="SimpleForm-container">              <FormContext.Provider value={form}>                  {children}              </FormContext.Provider>          </div>      )  }  export default SimpleForm |

**Exercise 1.7:**

Demo:

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

Code:

Client :

|  |
| --- |
| import { useState, useEffect } from "react";  const useMessages = (forum) => {    const [data, setData] = useState([]);    const [loading, setLoading] = useState(false);    const [error, setError] = useState(null);    useEffect(() => {      let didCancel = false;      setError(null);        if (forum) {          ; (async () => {          try {            setLoading(true);            const response = await fetch(              `http://localhost:5000/messages/${forum}`, {method :"GET"}            );            if (!response.ok) {              const text = await response.text()              throw new Error(                  `Unable to fetch messages for ${forum}: ${text}`              )            }            const body = await response.json();            if (!didCancel) {              setData(body);            }          } catch (err) {            setError(err);          } finally {            setLoading(false);          }        })()      } else {          setData([]);          setLoading(false);      }      return () => {        didCancel = true;      }    }, [forum])    return { data, loading, error }  };  export default useMessages; |

|  |
| --- |
| import { useState } from "react";  import useMessages from "../Component/useMessages";  function App() {    const [forum, setForum] = useState("nasa");    const {      data: messages,      loading: messagesLoading,      error: messagesError,    } = useMessages(forum);    return (      <div className="App">        <button onClick={() => setForum("nasa")}>NASA</button>        <button onClick={() => setForum("notNasa")}>Not NASA</button>        {messagesError ? (          <div className="error">            Something went wrong:            <div className="error-contents">              {messagesError.message}            </div>          </div>        ) : messagesLoading ? (          <div className="loading">Loading...</div>        ) : messages && messages.length ? (          <dl>            {messages.map((m) => (              <>                  <dt>{m.author}</dt>                  <dd>{m.text}</dd>              </>            ))}          </dl>        ) : (          "No messages"        )}      </div>    );  }  export default App; |

Server

|  |
| --- |
| const cors = require("cors");  const express = require("express");  const app = express();  app.use(express.json());  app.use(cors());  const messages = [    {      author: "SC",      text: "Rolls complete and a pitch is program. One BRAVO.",    },    {      author: "PAO",      text: "One BRAVO is an abort control model. Altitude is 2 miles.",    },    {      author: "CAPCOU",      text: "All is well at Houston. You are good at 1 minute.",    },  ];  app.get('/messages/:forum', (request, response) => {    if (request.params && request.params.forum === 'nasa') {      return response.send(messages);    }    return response.status(404).send({ error: "Unknown forum" });  });  app.post("/messages/:forum", (request, response) => {    if (request.params && request.params.forum === 'nasa') {      messages.push(request.body);      return response.send("OK");    }    return response.status(404).send({ error: "Unknown forum" });  });  app.listen(5000, () => console.log("Launched on port 5000!")); |

**Exercise 2:**

Demo:

A screenshot of a computer

Description automatically generated

Code:

Front-end:

|  |
| --- |
| import DatingCards from './components/datingcard/DatingCard';  import Header from './components/header/Header';  import SwipeButtons from './components/swipebutton/SwipeButtons';  function App() {      return (          <div className="app">              <Header />              <DatingCards />              <SwipeButtons />          </div>      );  }  export default App; |

DatingCards:

|  |
| --- |
| import axios from 'axios';  import React, { useEffect, useState } from 'react';  import TinderCard from 'react-tinder-card';  import './DatingCards.css';  const DatingCards = () => {      const [people, setPeople] = useState([]);      useEffect(() => {          async function fetchData() {              const req = await axios.get('http://localhost:4000/dating/cards');              setPeople(req.data);          }          fetchData();      }, []);      const swiped = (direction, nameToDelete) => {          console.log('receiving' + nameToDelete);      };      const outOfFrame = (name) => {          console.log(name + 'left the screen!!');      };      return (          <div className="datingCards">              <div className="datingCards\_\_container">                  {people.map((person) => (                      <TinderCard                          className="swipe"                          key={person.name}                          preventSwipe={['up', 'down']}                          onSwipe={(dir) => swiped(dir, person.name)}                          onCardLeftScreen={() => outOfFrame(person.name)}                      >                          <div style={{ backgroundImage: `url(${person.imgUrl})`,                                          backgroundSize: 'cover' }} className="card">                              <h3>{person.name}</h3>                          </div>                      </TinderCard>                  ))}              </div>          </div>      );  };  export default DatingCards; |

|  |
| --- |
| .datingCards\_\_container{      display: flex;      justify-content: center;      margin-top: 10vh;  }  .card{      position: relative;      background-color: white;      width: 600px;      padding: 20px;      max-width: 85vw;      height: 50vh;      box-shadow: 0px 18px 53px 0px rgba(0, 0, 0, 0.3);      border-radius: 20px;      background-size: contain;      background-position: center;      background-repeat: no-repeat;  }  .swipe{      position: absolute;  }  .cardContent{      width: 100%;      height: 100%;  }  .card h3{      position: absolute;      bottom: 0;      margin: 10px;      color: white;  }    \*{    margin: 0;    padding: 0;  } |

Header

|  |
| --- |
| import PersonIcon from '@mui/icons-material/Person';  import ForumIcon from '@mui/icons-material/Forum';  import { IconButton } from '@mui/material';  import './Header.css';  import React from 'react';  const Header = () => {      return (          <div className="header">              <IconButton>                  <PersonIcon fontsize="large" className="header\_\_icon" />              </IconButton>              <img width="20px" height="20px" className="header\_\_logo" src="logo192.png" alt="header" />              <IconButton>                  <ForumIcon fontsize="large" className="header\_\_icon" />              </IconButton>          </div>      );  };  export default Header; |

|  |
| --- |
| .header {      display: flex;      align-items: center;      justify-content: space-between;      z-index: 100;      border-bottom: 1px solid #f9f9f9;  }  .header\_\_logo {      object-fit: contain;      height: 40px;  }  .header\_\_icon {      padding: 20px;      color: black;  } |

Swipebutton:

|  |
| --- |
| import React from "react";  import "./SwipeButtons.css";  import ReplayTwoToneIcon from '@mui/icons-material/ReplayTwoTone';  import CloseTwoToneIcon from '@mui/icons-material/CloseTwoTone';  import StarIcon from '@mui/icons-material/Star';  import FavoriteIcon from '@mui/icons-material/Favorite';  import FlashOnTwoToneIcon from '@mui/icons-material/FlashOnTwoTone';  import { IconButton } from '@mui/material';  const SwipeButtons = () => {      return (          <div className="swipeButtons">              <IconButton>                  <ReplayTwoToneIcon className="swipeButtons\_\_repeat" />              </IconButton>              <IconButton>                  <CloseTwoToneIcon className="swipeButtons\_\_left" />              </IconButton>              <IconButton>                  <StarIcon className="swipeButtons\_\_star" />              </IconButton>              <IconButton>                  <FavoriteIcon className="swipeButtons\_\_right" />              </IconButton>              <IconButton>                  <FlashOnTwoToneIcon className="swipeButtons\_\_lightning" />              </IconButton>          </div>      );  };  export default SwipeButtons; |

|  |
| --- |
| .swipeButtons {      position: fixed;      bottom: 10vh;      display: flex;      width: 100%;      justify-content: space-evenly;  }  .swipeButtons .MuiIconButton-root {      background-color: white;      box-shadow: 0px 10px 53px 0px rgba(0, 0, 0, 0.3) !important;  }  .swipeButtons\_\_repeat {      padding: 1vw !important;      color: #f5b748 !important;  }  .swipeButtons\_\_left {      padding: 1vw !important;      color: #ec5e6f !important;  }  .swipeButtons\_\_star {      padding: 1vw !important;      color: #62b4f9 !important;  }  .swipeButtons\_\_right {      padding: 1vw !important;      color: #76e2b3 !important;  }  .swipeButtons\_\_lightning {      padding: 1vw !important;      color: #915dd1 !important;  } |

|  |
| --- |
| import axios from 'axios'  const baseURL = "http://localhost:4000/dating/cards";  const instance = axios.create({      baseURL: baseURL  })  export default instance; |

Backend

Card model

|  |
| --- |
| const mongoose = require('mongoose');  const cardSchema = mongoose.Schema({      name: String,      imgUrl: String  })  module.exports = mongoose.model('Card', cardSchema) |

Server

|  |
| --- |
| const mongoose = require('mongoose');  const Card = require('./CardsModel.js');  require('dotenv').config();  const cors = require('cors');  const express = require('express');  const app = express()  const port = process.env.PORT  const connection\_url = 'mongodb+srv://20521472:nguyenvantruongkhoa20521472@cluster0.cz6lnjt.mongodb.net/?retryWrites=true&w=majority'  app.use(express.json())  app.use(cors())  try {      mongoose.connect(connection\_url);      console.log('database successfully connected');  } catch (error) {      throw error;  }  app.get("/", (req, res) => res.status(200).send("Hello TheWebDev"))  app.post("/dating/cards", (req, res) => {      const dbCards = req.body      console.log(dbCards);      Card.create(dbCards, (err, data) => {          if (err) {              res.status(500).send(err)          } else {              res.status(201).send(data)          }      })  })  app.get("/dating/cards", (req, res) => {      Card.find((err, data) => {          if (err) {              res.status(500).send(err)          } else {              res.status(201).send(data)          }      })  })  app.listen(port, () => console.log(`Listening on localhost: http://localhost:${port}`)) |