Web application development

(Introduction to Basic React)

Instructor: Tran Vinh Khiem

September 1st, 2022

S3T
Smart Software System Team

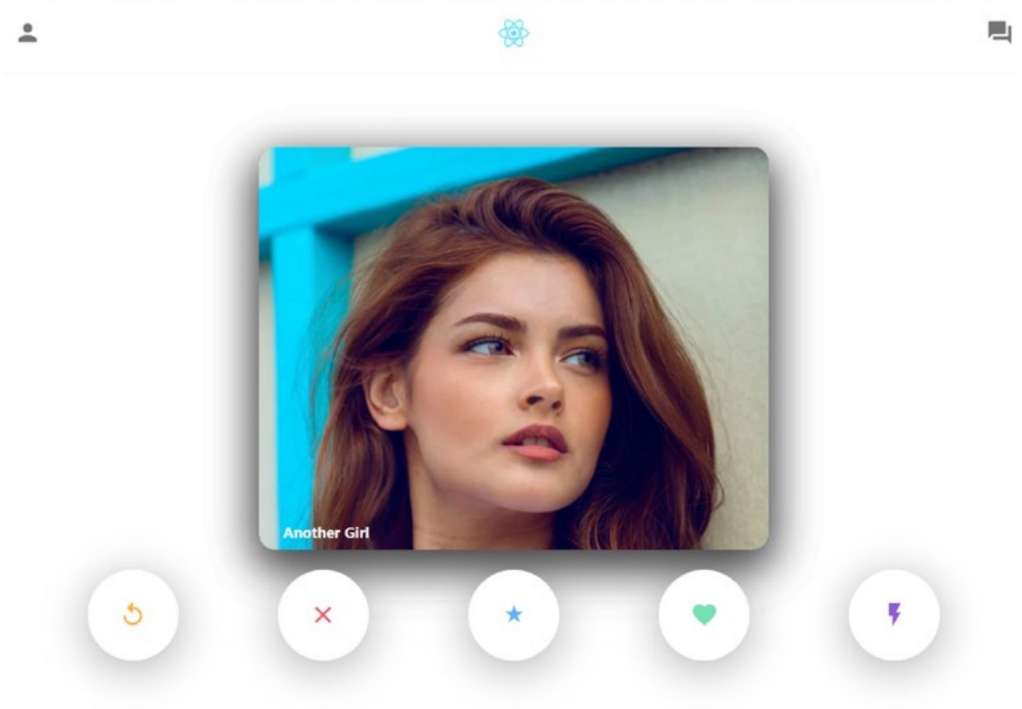

“We love what we do and we do what our clients love & work with great clients all over the world to create thoughtful and purposeful websites.”
— ProWeb365

Basic React – Exercise 0 – State management

Access to this link and do exercise:
https://docs.google.com/document/d/1pN4TKbbrquxEDb9Qtu6RqUWldbzvrlryXQiOMGlWdnQ/

https://docs.google.com/document/d/1NfNBxkK9Yh18qkAaxrlaLr9ds8XTFSlVNvdxQeEMT34/

Basic React – Exercise 1 – Create tinder "phake" app



Basic React – Exercise 1 – Create tinder "pha ke" app – Front End – app.js and Header

```
import './App.css';
import DatingCards from './components/DatingCards';
import Header from './components/Header';
import SwipeButtons from './components/SwipeButtons';

function App() {
  return (
    <div className="app">
      <Header />
      <DatingCards />
      <SwipeButtons />
    </div>
  );
}

export default App;
```

```
import React from 'react'
import './Header.css'
import PersonIcon from '@material-ui/icons/Person'
import IconButton from '@material-ui/core/IconButton'
import ForumIcon from '@material-ui/icons/Forum'

const Header = () => {
    return (
        <div className="header">
            <IconButton>
                <PersonIcon fontSize="large" className="header__icon" />
            </IconButton>
            <img className="header__logo" src="logo192.png" alt="header" />
            <I
                import IconButton
                ze="large" className="header__icon" />
            </IconButton>
        </div>
    )
}

export default Header
```

Basic React – Exercise 1 – Create tinder "pha ke" app – Front End - DatingCard

```jsx
import React, { useState, useEffect } from 'react'
import DatingCard from 'react-tinder-card'
import './DatingCards.css'
import axios from './axios'

const DatingCards = () => {
    const [people, setPeople] = useState([])

    useEffect(() => {
        async function fetchData() {
            const req = await axios.get("/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) => (
                    <DatingCard
                        className="swipe"
                        key={person.name}
                        preventSwipe={['up', 'down']}
                        onSwipe={(dir) => swiped(dir, person.name)}
                        onCardLeftScreen={() => outOfFrame(person.name)}
                    >
                        <div style={{ backgroundImage: `url(${person.imgUrl})` }} className=
                            <h3>{person.name}</h3>
                        </div>
```

```jsx
                    </DatingCard>
                ))}
            </div>
        </div>
    )
}

export default DatingCards
```

Basic React – Exercise 1 – Create tinder "pha ke" app – Front End – Swipe Button

```
import React from 'react'
import './SwipeButtons.css'
import ReplayIcon from '@material-ui/icons/Replay'
import CloseIcon from '@material-ui/icons/Close'
import StarRateIcon from '@material-ui/icons/StarRate'
import FavoriteIcon from '@material-ui/icons/Favorite'
import FlashOnIcon from '@material-ui/icons/FlashOn'
import IconButton from '@material-ui/core/IconButton'

const SwipeButtons = () => {
    return (
        <div className="swipeButtons">
            <IconButton className="swipeButtons__repeat">
                <ReplayIcon fontSize="large" />
            </IconButton>
            <IconButton className="swipeButtons__left">
                <CloseIcon fontSize="large" />
            </IconButton>
            <IconButton className="swipeButtons__star">
                <StarRateIcon fontSize="large" />
            </IconButton>
            <IconButton className="swipeButtons__right">
                <FavoriteIcon fontSize="large" />
            </IconButton>
            <IconButton className="swipeButtons__lightning">
                <FlashOnIcon fontSize="large" />
            </IconButton>
        </div>
    )
}

export default SwipeButtons
```

Basic React – Exercise 1 – Create tinder "phake" app – Front End – Connect to BE

```
import axios from 'axios'

const instance = axios.create({
    baseURL: "YourBackendURL"
})

export default instance
```

Basic React – Exercise 1 – Create tinder "phake" app – Back end

- Set up firebase
- Set up mongoDB
  - npm i express mongoose
  - npm i nodemon

Basic React – Exercise 1 – Create tinder "phake" app – Backend – database card schema

```
import mongoose from 'mongoose'

const cardSchema = mongoose.Schema({
    name: String,
    imgUrl: String
})

export default mongoose.model('cards', cardSchema)
```

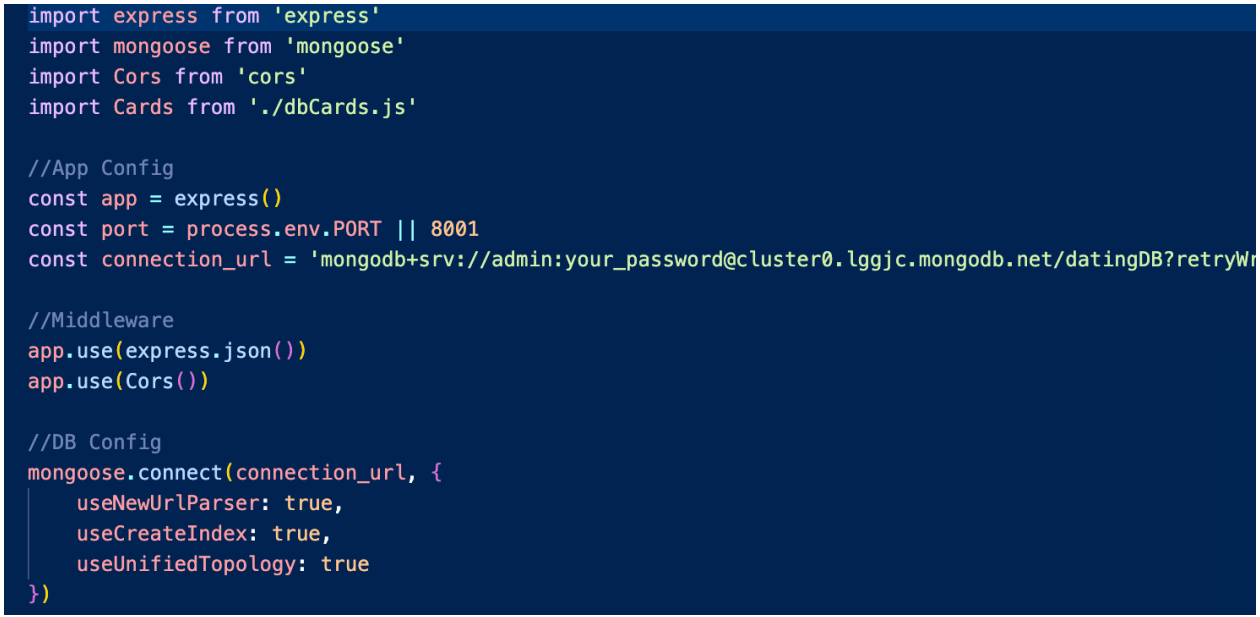
Basic React – Exercise 1 – Make your app more attractive

Access to this link and do exercise:
https://docs.google.com/document/d/1SeT_qmLkCK9Y1PpEHF_r0zVbdsfUlRfSDzhc_h6Rz6g

After that integrating this to your project.

Remember to responsive and integrate to your project, too.
https://docs.google.com/document/d/1oqvfXfq5dIVftXBu5jHeFGj0J4DEynIUIluzac-jeUI/

Basic React – Exercise 1 – Create tinder "pha ke" app – Backend – Server



```
import express from 'express'
import mongoose from 'mongoose'
import Cors from 'cors'
import Cards from './dbCards.js'

//App Config
const app = express()
const port = process.env.PORT || 8001
const connection_url = 'mongodb+srv://admin:your_password@cluster0.lggjc.mongodb.net/datingDB?retryWr

//Middleware
app.use(express.json())
app.use(Cors())

//DB Config
mongoose.connect(connection_url, {
    useNewUrlParser: true,
    useCreateIndex: true,
    useUnifiedTopology: true
})
```

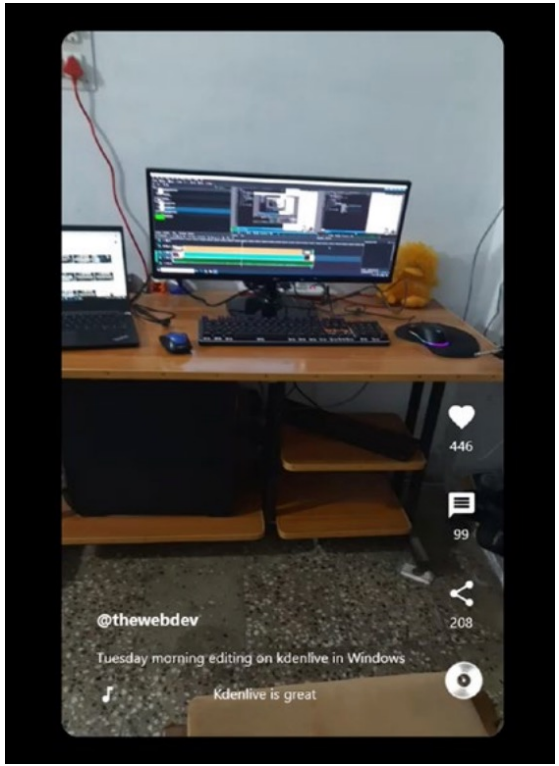
Basic React – Exercise 1 – Create tinder "phake" app – Backend – Server

```
//API Endpoints
app.get("/", (req, res) => res.status(200).send("Hello TheWebDev"))

app.post('/dating/cards', (req, res) => {
    const dbCard = req.body
    Cards.create(dbCard, (err, data) => {
        if(err) {
            res.status(500).send(err)
        } else {
            res.status(201).send(data)
        }
    })
})

app.get('/dating/cards', (req, res) => {
    Cards.find((err, data) => {
        if(err) {
            res.status(500).send(err)
        } else {
            res.status(200).send(data)
        }
    })
})

//Listener
app.listen(port, () => console.log(`Listening on localhost: ${port}`))
```

Basic React – Exercise 1 - Finalize app

Follow this guide to set up your project on github:
https://docs.google.com/document/d/1-33Bdhp5etWNsKZHhznGCYcPYFFNKtMiF76-hTkAMNU/

Basic React – Exercise 2 – Create reels app



Basic React – Exercise 2 – Create reels app - FE

```jsx
import React, { useState, useEffect } from 'react'
import './App.css';
import Video from './components/Video';
import axios from './components/axios';

function App() {
  const [videos, setVideos] = useState([])

  useEffect(() => {
    async function fetchData() {
      const res = await axios.get("/v2/posts")
      setVideos(res.data)
      return res
    }
    fetchData()
  }, [])

  return (
    <div className="app">
      <div className="app__videos">
        {videos.map(({ url, channel, description, song, likes, shares, messages }) => (
          <Video
            key={url}
            url={url}
            channel={channel}
            description={description}
            song={song}
            likes={likes}
            shares={shares}
            messages={messages}
          />
        ))}
      </div>
    </div>
  );
}
export default App;
```

```css
html{
    scroll-snap-type: y mandatory;
}

.app{
    height: 100vh;
    background-color: black;
    display: grid;
    place-items: center;
}

.app__videos{
    position:relative;
    height: 800px;
    border-radius: 20px;
    overflow: scroll;
    width: 80%;
    max-width: 500px;
    scroll-snap-type: y mandatory;
}

.app__videos::-webkit-scrollbar{
    display: none;
}

.app__videos{
    -ms-overflow-style: none;
    scrollbar-width: none;
}
```

Basic React – Exercise 2 – Create reels app - FE

```jsx
import React, { useRef, useState } from 'react'
import './Video.css'
import VideoFooter from './VideoFooter'
import VideoSidebar from './VideoSidebar'

const Video = ({ url, channel, description, song, likes, shares, messages }) => {
    const [playing, setPlaying] = useState(false)
    const videoRef = useRef(null)
    const handleVideoPress = () => {
        if(playing){
            videoRef.current.pause()
            setPlaying(false)
        } else {
            videoRef.current.play()
            setPlaying(true)
        }
    }

    return (
        <div className="video">
            <video
                src={url}
                className="video__player"
                loop
                ref={videoRef}
                onClick={handleVideoPress}
            >
            </video>
            <VideoFooter channel={channel} description={description} song={song} />
            <VideoSidebar likes={likes} shares={shares} messages={messages} />
        </div>
    )
}

export default Video
```

```css
.video{
    position: relative;
    background-color: white;
    width: 100%;
    height:100%;
    scroll-snap-align: start;
}

.video__player{
    object-fit: fill;
    width: 100%;
    height: 100%;
}
```

Basic React – Exercise 2 – Create reels app - FE

```jsx
import React from 'react'
import './VideoFooter.css'
import MusicNoteIcon from '@material-ui/icons/MusicNote'
import Ticker from 'react-ticker'

const VideoFooter = ({ channel, description, song }) => {
    return (
        <div className="videoFooter">
            <div className="videoFooter__text">
                <h3>@{channel}</h3>
                <p>{description}</p>
                <div className="videoFooter__ticker">
                    <MusicNoteIcon className="videoFooter__icon" />
                    <Ticker mode="smooth">
                        {({ index }) => (
                            <>
                                <p>{song}</p>
                            </>
                        )}
                    </Ticker>
                </div>
            </div>
            <img className="videoFooter__record" src="your-video-icon" alt="video footer" />
        </div>
    )
}

export default VideoFooter
```

```css
.videoFooter{
    position: relative;
    color: white;
    bottom: 150px;
    margin-left: 40px;
    display: flex;
}

.videoFooter__text{
    flex: 1;
}

.videoFooter__text > h3{
    padding-bottom: 20px;
}

.videoFooter__text > p{
    padding-bottom: 20px;
}

.videoFooter__icon{
    position: absolute;
}

.videoFooter__ticker > .ticker{
    height: fit-content;
    margin-left: 30px;
    width: 60%;
}

.videoFooter__record{
    animation: spinTheRecord infinite 5s linear;
    height: 50px;
    filter: invert(1);
    position: absolute;
    bottom: 0;
    right: 20px;
}
```

Basic React – Exercise 2 – Create reels app - FE

```jsx
import React, { useState } from 'react'
import './VideoSidebar.css'
import FavoriteIcon from '@material-ui/icons/Favorite'
import FavoriteBorderIcon from '@material-ui/icons/FavoriteBorder'
import MessageIcon from '@material-ui/icons/Message'
import ShareIcon from '@material-ui/icons/Share'

const VideoSidebar = ({ likes, shares, messages }) => {
    const [liked, setLiked] = useState(false)

    return (
        <div className="videoSidebar">
            <div className="videoSidebar__button">
                { liked ? <FavoriteIcon fontSize="large" onClick={e => setLiked(false)} /> : <FavoriteBorderIcon fontSize="large" onClick={e => setLiked(true)} /> }
                <p>{liked ? +likes + 1 : likes}</p>
            </div>
            <div className="videoSidebar__button">
                <MessageIcon fontSize="large" />
                <p>{messages}</p>
            </div>
            <div className="videoSidebar__button">
                <ShareIcon fontSize="large" />
                <p>{shares}</p>
            </div>
        </div>
    )
}

export default VideoSidebar
```

```css
.videoSidebar{
    position: absolute;
    top: 50%;
    right: 10px;
    color: white;
}

.videoSidebar__button{
    padding: 20px;
    text-align: center;
}
```

Basic React – Exercise 2 – Create reels app - FE

```
import axios from 'axios'

const instance = axios.create({
    baseURL: "your-backend-heroku-url"
})

export default instance
```

Basic React – Exercise 2 – Make your app more attractive

Access to this link and do exercise:
https://docs.google.com/document/d/1SeT_qmLkCK9Y1PpEHF_r0zVbdsfUlRfSDzhc_h6Rz6g

After that integrating this to your project.

Remember to responsive and integrate to your project, too.
https://docs.google.com/document/d/1oqvfXfq5dIVftXBu5jHeFGj0J4DEynIUIluzac-jeUI/

Basic React – Exercise 2 – Create reels app - BE

```
import mongoose from 'mongoose'

const shortVideoSchema = mongoose.Schema({
    url: String,
    channel: String,
    description: String,
    song: String,
    likes: String,
    shares: String,
    messages: String
})

export default mongoose.model('shortVideos', shortVideoSchema)
```

Basic React – Exercise 2 – Create reels app - BE

```
import express from 'express'
import mongoose from 'mongoose'
import Cors from 'cors'
import Videos from './dbModel.js'
//App Config
const app = express()
const port = process.env.PORT || 9000
const connection_url = 'your-mongodb-api-url'

//Middleware
app.use(express.json())
app.use(Cors())

//DB Config
mongoose.connect(connection_url, {
    useNewUrlParser: true,
    useCreateIndex: true,
    useUnifiedTopology: true
})

//API Endpoints
app.get("/", (req, res) => res.status(200).send("He so lo , ho so ly ly"))

app.post('/v2/posts', (req, res) => {
    const dbVideos = req.body
    Videos.create(dbVideos, (err, data) => {
        if(err)
            res.status(500).send(err)
        else
            res.status(201).send(data)
    })
})
```

```
app.get('/v2/posts', (req, res) => {
    Videos.find((err, data) => {
        if(err) {
            res.status(500).send(err)
        } else {
            res.status(200).send(data)
        }
    })
})
//Listener
app.listen(port, () => console.log(`Listening on localhost: ${por
```

Basic React – Exercise 2 – Create reels app – Finalize app

Follow this guide to set up your project on github:
https://docs.google.com/document/d/1-33Bdhp5etWNsKZHhznGCYcPYFFNKtMiF76-hTkAMNU/

Q & A

Thank you for cooperating
Gét gô

“Coming together is a beginning;
Keeping together is progress;
Working together is success.”
- HENRY FORD