K-Fitness App-> A modern responsive React fitness exercises app with the functionality to choose exercise categories and specific muscle groups browse more than a thousand exercises with practical examples pagination exercise details pulling related videos from YouTube displaying similar exercises and much more. Alongside building the App, I have learned a lot of folder and file structure and hooks, a stylish user interface using material UI creation, and API data fetching techniques from unlimited sources using Rapid API.

Developing Languages, Tools, and Techniques Needed:

Vscode 1.74 https://code.visualstudio.com/updates/v1_74
npm/npx https://www.npmjs.com/package/npx
React Native JS https://reactjs.org/
JavaScript https://www.javascript.com/
ES7+ React/Redux/React-Native snippets
https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets
Yarn https://yarnpkg.com/
RapidAPI ExerciseDB https://rapidapi.com/justin-WFnsXH_t6/api/exercisedb
ESLint v2.2.6 https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint
RapidAPI Youtube Search and Download
https://rapidapi.com/h0p3rwe/api/youtube-search-and-download

Prerequisites & Setups:

In Vscode Terminal, install all React App dependencies and packages by running:

```
npx create-react-app ./
```

Create `index.js` in `src` and setup imports and roots rendering:

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <App />
```

Create `App.js` in `src` and setup:

```
import React from 'react';
const App = () => {
    return (
        <div>App</div>
...export default App
```

Install needed dependencies:

```
 yarn add @emotion/react
 yarn add @emotion/react @emotion/styled
 yarn add @mui/icons-material
 yarn add @mui/material
 npm install --save react-horizontal-scroll-menu
 yarn add react-loader-spinner
 npm i react-router-dom
```

Audit all installed dependencies to the latest version:

```
  npm install --legacy-peer-deps
```

Start the web server at localhost:3000:

```
 npm start
```

Web server console error:

"Warning: You are importing createRoot from 'react-dom' which is not supported. You should instead import it from 'react-dom/client'".

DEBUGGING: In `index.js:`

```
import ReactDOM from 'react-dom/client';
```

Synchronous Developing Notes:

NavBar:

Customize NavBar sections styles in `Navbar.js:`

```
   <Stack direction="row"
            justifyContent="space-around"
            sx={{
                gap: { sm: '122px', xs: '40px' },
                mt: { sm: '32px', xs: '20px' },
                justifyContent: 'none'
            }} px = "20px">
            <Link to="/">
                <img src={Logo} alt="logo" style={{
                    width: '48px',
                    height: '48px', margin: '0 20px'
                }} />
...
```

navbar home and exercise customized.PNG

Hero Banner Customization:

Customize hero banner get better design in `HeroBanner.js:`

```
</Typography>
            <Button variant="contained"
                color="error" href="#exercises"
                sx={{
                    backgroundColor: '#ff2625',
                    padding: '10px'
                }}
            >Explore Exercises</Button>
            <Typography
...
```

hero banner well designed.PNG

Search Exercises Customization:

Customize hover search bar and button at footer in `SearchExercises.js:`

```
  <Button className="search-btn"
                    sx={{
```

```
        bgcolor: '#FF2625',
        color: '#fff',
        textTransform: 'none',
        width: { lg: '175px', xs: '80px' },
        fontSize: { lg: '20px', xs: '14px' },
        height: '56px',
        position: "absolute",
        right: '0'
    }}
    onClick={handleSearch}>
...
```

search bar button and text input are customized.PNG

Fetch Data:

Obtain API keys and needed data from RapidAPI, fetch search exercises data in `SearchExercises.js:`

```
const SearchExercises = () => {
    const [search, setSearch] = useState('')
    const handleSearch = async () => {
        if (search) {
            const exercisesData = await fetchData(
                'https://exercisedb.p.rapidapi.com/exercises',
exerciseOptions);
            console.log(exercisesData);
     ...
```

Now test SEARCH button, data fetched at localhost:3000/#exercises:

rapid api search data fetched.PNG

Install scroll menu:

```
 npm install react-horizontal-scrolling-menu
```

Body Parts Components:

Eslint Error. DEBUGGING: To disable Eslint error from popping out, install above Vscode Eslint extension and right click to Disable Eslint at workplace.

Eslint Compiled Error: XXX is missing in props validation react/prop-types. DEBUGGING: At certain file, do `import PropTypes from 'prop-types';` Then right before the `export default FILENAME,` do `FILENAME.propTypes = {XXX: PropTypes.XXX'S DATA TYPE[REFRESH CONSOLE ELEMENT TO SEE].isRequired,...}`

Implement Body parts exercises components as buttons in `BodyPart.js:`

```
  type="button"
        alignItems="center"
        justifyContent="center"
        className="bodyPart-card"
        sx={bodyPart === item ? { borderTop: '4px solid #FF2625',
background: '#fff', borderBottomLeftRadius: '20px', width: '270px',
height: '282px', cursor: 'pointer', gap: '47px' } : { background:
'#fff', borderBottomLeftRadius: '20px', width: '270px', height:
'282px', cursor: 'pointer', gap: '47px' }}
        onClick={() => {
            setBodyPart(item);
            window.scrollTo({ top: 1800, left: 100, behavior: 'smooth'
});
...
```

body parts components are well customized.PNG

Exercise Cards Components:

Fetch exercise GIF cards binding data in `ExerciseCard.js`:

```
const ExerciseCard = ({ exercise }) => {
    return (
        <Link className="exercise-card"
to={`/exercise/${exercise.id}`}>
            <img src={exercise.gifUrl} alt={exercise.name}
loading="lazy" />
        </Link>
```

Now search for “legs” and all legs results GIF exercises cards showed:

legs cards gif results showed.PNG

Customize the cards buttons:

```
  <Button sx={{
          ml: '21px', color: '#fff', background: '#ffa9a9',
       fontSize: '14px', borderRadius: '20px', textTransform:
'capitalize'
                }}>
          {exercise.bodyPart}
   </Button>
```

cards with two buttons showed.PNG

Now fetch each card’s title:

```
  </Stack>
     <Typography ml="21px" color="#000" fontWeight="bold" mt="11px"
         pb="10px" textTransform="capitalize" fontSize="24px">
        {exercise.name}
     </Typography>
```

cards titles showed.PNG

Now implement the showing results slices pages in `Exercises.js`:

```
  const [currentPage, setCurrentPage] = useState(1);
    const exercisesPerPage = 9;
    const indexOfLastExercise = currentPage * exercisesPerPage;
    const indexOfFirstExercise = indexOfLastExercise -
exercisesPerPage;
```

```
    const currentExercises = exercises.slice(indexOfFirstExercise,
indexOfLastExercise);
    const paginate = (e, value) => {
...
```

page slices showed.PNG

Home page is finalized.

Exercise Details:

Fetch exercise cards details in `Detail.js:`

```
  <Stack sx={{ gap: { lg: '35px', xs: '20px' } }}>
                <Typography variant="h3">
                    {name}
                </Typography>
                <Typography variant="h6">
                    Exercises keep you strong. {name} {` `}
                    is one of the best exercises to target your
{target}. It will help you
                    improve your mood and gain energy.
                </Typography>
...
```

exercise cards details gif fetched.PNG

Fetch more related videos of one specific video in `ExerciseVideos.js:`

```
  key={index}
                      className="exercise-video"

href={`https://www.youtube.com/watch?v=${item.video.videoId}`}
                        target="_blank"
                        rel="noreferrer"
                    >
                        <img src={item.video.thumbnails[0].url}
alt={item.video.title} />
                        <Box>
                            <Typography variant="h5" color="#000">
                                {item.video.title}
                            </Typography>
                            <Typography variant="h6" color="#000">
                                {item.video.channelName}
                            </Typography>
                        </Box>
...
```

more related videos on youtube of one specific video.PNG

Similar Exercises:

To fetch the similar videos attached to a specific video, in `SimilarExercises:`

```
 <Box sx={{ mt: { lg: '100px', xs: '0' } }}>
      <Typography variant="h3" mb={5}>Exercises that target the same
muscle group</Typography>
      <Stack direction="row" sx={{ p: '2', position: 'relative' }}>
        {targetMuscleExercises.length ?
          <HorizontalScrollbar data={targetMuscleExercises} />
          : <Loader />
        }
...
```

similar exercises videos fetched.PNG

Also fetch exercises with the same equipments:

```
 </Stack>
      <Typography variant="h3" mb={5}>Exercises that target the same
equipments</Typography>
      <Stack direction="row" sx={{ p: '2', position: 'relative' }}>
        {equipmentExercises.length ?
          <HorizontalScrollbar data={equipmentExercises} />
          : <Loader />
...
```

exercises with same equipment shows as well.PNG

Footer Customization:

In `Footer.js:`

```
  <Stack gap="40px" alignItems="center" px="40px" pt="24px">
                <img src = {Logo} alt = "logo"/>
                <Typography variant = "h5" pb ="40px" mt="20px">
                    Made with loves by Krystal Zhang
                </Typography>
...
```

footer done.PNG