React

Thursday, September 11, 2025 11:59 AM

1.FITNESS TRACKER

Fitness Tracker.js

```
const FitnessTracker = () => {
  const[logs,setLogs] = useState([]);
  const addLog = (log) => {
    setLogs((prevLogs) => [...prevLogs,log]);
  };
  const resetLogs = () => {
    setLogs([]);
  }

<h1>Track Your Fitness</h1>
     <LogForm addLog = {addLog} resetLogs={resetLogs}/>
     <LogList logs = {logs}/>
```

_____

LogForm.js

```
const LogForm = ({addLog,resetLogs}) => {
  const[excerciseType,setExcerciseType] = useState("");
  const[duration,setDuration] = useState("");
  const[caloriesBurned,setCaloriesBurned] = useState("");
  const[error,setError] = useState("");

  const handleSubmit = (e) =>{
    e.preventDefault();

    if(!excerciseType.trim()){
      setError("Exercise type must not be empty.");
      return;
    }
    if(isNaN(duration) || duration <= 0){
      setError("Duration must be a positive number.");
      return;
    }
    if(isNaN(caloriesBurned) || caloriesBurned <= 0){
      setError("Calories must be a positive number.");
      return;
    }
    addLog({excerciseType,duration,caloriesBurned});
    setExcerciseType("");
    setDuration("");
    setCaloriesBurned("");
    setError("");
  };

  const handleReset =() =>{
    resetLogs();
    setError("");
  };
onSubmit={handleSubmit}>
value={excerciseType}
          onChange={(e) => setExcerciseType(e.target.value)}
value={duration}
          onChange={(e) => setDuration(e.target.value)}
value={caloriesBurned}
          onChange={(e) => setCaloriesBurned(e.target.value)}
</form>
     {error &&(
     <p data-testid="error-message" style={{ color: "red" }}>
       {error}
     </p> )}
```

_____

Loglist.js

```
const LogList = ({logs}) => {
```

2.React - Employee

AddEmployee.js

```
import React, {useState,Fragment } from 'react'

const AddEmployee = ({appEmployeesList,setAppEmployeesList}) => {
  const[name,setName] = useState('');
  const[position,setPosition] = useState('');
  const[salary,setSalary] = useState('');

  const isFormValid = name.trim() !== '' && position.trim() !== '' && salary.trim() !
== '';
  const handleAdd =() =>{
  const newEmployee = {
    id: appEmployeesList.length > 0 ?
appEmployeesList[appEmployeesList.length-1].id+1:0,
    name,
    position,
    salary:parseInt(salary),
  };
  setAppEmployeesList([...appEmployeesList,newEmployee]);
  setName('');
  setPosition('');
  setSalary('');
};
.....
placeholder='Enter Name'
        value = {name}
        onChange={(e) => setName(e.target.value)}

placeholder='Enter Position'
        value = {position}
        onChange={(e) => setPosition(e.target.value)}
placeholder='Enter Salary'
        value = {salary}
        onChange={(e) => setSalary(e.target.value)}
className='x-small w-75 ma-0 px-25'
        disabled = {!isFormValid}
        onClick={handleAdd}
.....
Employee.js
value={salary}
          onChange={(e) => {
            const newSalary = e.target.value;
            setSalary(newSalary);

            const isValid =
              newSalary.trim() !== '' &&
              !isNaN(newSalary) &&
              parseInt(newSalary) >= 0;

            const isDifferent =
              parseInt(newSalary) !== appEmployeesList[idx].salary;

            setEnableSave(isValid && isDifferent);
           }}
.....
onClick={() => {
          let newList = [...appEmployeesList];
          newList[idx].salary = parseInt(salary); // savebutton
          setAppEmployeesList(newList);
          setIsEditing(false);
          setEnableSave(false);
         }}
.....
```

```
    </p> )}
```

Loglist.js

```
const LogList = ({logs}) => {
 return (
   <div data-testid="log-list">
    <h2>Activity Log</h2>
    {logs.length === 0 ? (
      <p>No activities logged yet.</p>
    ):(
    <ul className="pl-0 log-entry-ul">
     {logs.map((log,index) => (
     <li key = {index} data-testid="log-entry
-li mb-10 pa-10">
</strong> {log.excerciseType}
</strong> {log.duration}
</strong> {log.caloriesBurned}
```

3.Country Filter

Search.js

```
function Search({value,onChange})
value ={value}
    onChange={onChange}
```

CountryList.js

```
function CountryList({countries})
<ul>
{countries.map((country) => (
        <li key={country} className="pa-10 pl-20">{country}</li>

     ))}
    </ul>
```

App.js

```
function App() {
   const[filterText,setFilterText] = useState('');

   const handleChange = (e) =>{
      setFilterText(e.target.value);
   };

   const filteredCountries = response.filter(country =>
      country.toLowerCase().includes(filterText.toLowerCase())
   );
inside return(hint)
<Search value ={filterText} onChange ={handleChange}/>
            <CountryList countries ={filteredCountries}/>
```

5.JobApplicationForm.js

```
const JobApplicationForm = ({ formData, setFormData, errors,
onPreview, onReset }) => {
return (<form data-testid="job-application-form" className="layout-
column ml-auto" onSubmit = {onPreview}>

name="name" value={formData.name} onChange={(e) =>
setFormData({ ...formData, name: e.target.value })}
</label>
{errors.name && <p data-testid="error-name" className="error">
{errors.name}</p>}

name="email" value={formData.email} onChange={(e) =>
setFormData({ ...formData, email: e.target.value })}
</label>{errors.email && <p data-testid="error-email"
className="error">{errors.email}</p>}

name="experience" value={formData.experience} onChange={(e) =>
setFormData({ ...formData, experience: e.target.value })}
</label>{errors.experience && <p data-testid="error-experience"
className="error">{errors.experience}</p>}

onClick={onReset}> Reset
          setIsEditing(false);
          setEnableSave(false);
        }}
.....
const App = () => {
  const [appEmployeesList,setAppEmployeesList] =
React.useState([...employeesList]);
.....
<tbody>
     { appEmployeesList.map((employee, idx) => (
<tr key={ employee.id } data-testid={ `row-${idx}` }>
.....
<tr>
<AddEmployee  appEmployeesList={appEmployeesList}
setAppEmployeesList={setAppEmployeesList}/>
</tr>
.....
```

4.HackerShopCheckout

CartIndex.js

```
export default class Cart extends Component {
   handleCouponChange = (e) =>{
      const discountedValue = parseInt(e.targe.value,10);
      this.props.onCouponChange(discountedValue);
   };
   render() {
      const{items,subTotal,discount,totalPrice} = this.props.cart;
……
<tbody>{items.map((cartItem, idx) =>
……
${cartItem.price} (hint-testid="cart-item-price)
……
<span data-testid="cart-subtotal">${subTotal}</span>
<span className="discount" data-testid="cart-discount">${discount}</span>
<span data-testid="cart-total">${totalPrice}</span>
```

CartProduct.js

```
export default class ProductList extends Component {
   handleAddToCart = (product) =>{
      const updatedProduct ={...product,cartQuantity:1};
      this.props.onAddToCart(updatedProduct);
   };
   handleRemoveFromCart = (product) => {
      const updatedProduct ={...product,cartQuantity:0};
      this.props.onRemoveCart(updatedProduct);
   };
inside return(hint)
<button className="x-small outlined" data-testid="btn-item-add" onClick={() =>
this.handleAddToCart(product)}> Add To Cart </button>

<button className="x-small danger" data-testid="btn-item-remove" onClick={()
=> this.handleRemoveFromCart(product)}> Remove </button>
```

6.Date-API

DateButton.js

```
const DateButton = ({onClickButton})

variant="contained" onClick={onClickButton}

const DateDisplay = ({ apiResponse }) => {
let day = '', month = '', year = '';
if (apiResponse && apiResponse.date) {
const [dd, mm, yyyy] = apiResponse.date.split('-');
day = String(Number(dd));
month = String(Number(mm));
year = yyyy;}

<ListItemText data-testid="day">Day: {day}
<ListItemText data-testid="month">Month: {month}
<ListItemText data-testid="year">Year: {year}

DateDisplay.propTypes = {
```

```
className="error">{errors.experience}</p>}

onClick={onReset}> Reset

Preview.js
const Preview = ({ formData, onClear, onSubmit }) => {

Name:</strong> {formData.name}
Email:</strong> {formData.email}
Experience:</strong> {formData.experience}
onClick={onClear}>Clear
onClick={onSubmit}>Submit

SuccessMessage.js
SuccessMessage = ({onHome})

onClick = {onHome}> Home

App.js
const App = () => {
 const [step, setStep] = useState("form"); // form | preview | success
 const [formData, setFormData] = useState({
  name: "",
  email: "",
  experience: "",});
 const [errors, setErrors] = useState({});
 const validate = () => {
  const errs = {};
  if (!formData.name.trim()) {
   errs.name = "Name is required.";   }
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!emailRegex.test(formData.email)) {
   errs.email = "Enter a valid email address."; }
  const exp = Number(formData.experience);
  if (!exp || exp <= 0) { errs.experience = "Experience must be a positive
number.";}
  return errs; };
 const handlePreview = (e) => {
  e.preventDefault();
  const errs = validate();
  if (Object.keys(errs).length > 0) {
   setErrors(errs);
  } else {
   setErrors({});
   setStep("preview");    }  };
 const handleReset = () => {
  setFormData({ name: "", email: "", experience: "" });
  setErrors({});
  setStep("form"); };

<h8k-navbar header={title}></h8k-navbar>
    {step === "form" && (
<JobApplicationForm formData={formData}
setFormData={setFormData}
       errors={errors} onPreview={handlePreview}
onReset={handleReset}/> )}
    {step === "preview" && ( <Preview
       formData={formData}
       onClear={handleReset}
       onSubmit={() => setStep("success")} /> )}
    {step === "success" && <SuccessMessage onHome={handleReset}
/>}
```

8.HackerBank.js

```
------------
const [filteredTxns,setFilteredTxns] = useState(txns);
const [selectedDate, setSelectedDate] = useState("2019-11-29");
useEffect(() => {
if (!selectedDate) {
setFilteredTxns(txns);}
}, [selectedDate, txns]);
<ListItemText data-testid="month">Month: {month}
<ListItemText data-testid="year">Year: {year}

DateDisplay.propTypes = {
apiResponse: PropTypes.shape({
date: PropTypes.string,
time: PropTypes.string })};
----------
DateApi.js
----------
getAPIResponse() {return fetch(DATE_JSON_URL).then((res) => res.json());}
--------
App.js
--------
this.state= {apiResponse: null};

API.getAPIResponse().then((res) => {this.setState({ apiResponse: res }); });

<center><DateButton onClickButton={this.handleButtonClick} /></center>
<DateDisplay apiResponse={this.state.apiResponse}/>
```

7.Team Selection

```
Index.js
import React, { useState } from "react";
import playersList from "../../players.json";

export default function TeamSelection() {
 const [availablePlayers,setAvailablePlayers] = useState(playersList);
 const [selectedPlayers,setSelectedPlayers] = useState([]);
 const [disabledSelect,setDisabledSelect] = useState({});

 const sortBy = (key) => {
  // Empty function - no sorting functionality
  const sorted = [...availablePlayers].sort((a,b) => {
   if(typeof a[key] === "string"){
    return a[key].localeCompare(b[key]);
   }
   return a[key] - b[key];
  });
  setAvailablePlayers(sorted);

 };

 const addPlayer = (index) => {
  const player = availablePlayers[index];
  setSelectedPlayers([...selectedPlayers,player]);
  setDisabledSelect({...disabledSelect,[player.name]:true});
  // Empty function - no add functionality
 };

 const removePlayer = (name) => {
  const updatedSelected = selectedPlayers.filter(p => p.name !== name);
  setSelectedPlayers(updatedSelected);
  setDisabledSelect({...disabledSelect,[name]:false});
  // Empty function - no remove functionality
 };
.....................................................................
Bowl
</th>
<th>Action</th>
</tr>
</thead>
<tbody data-testid="available-players-table-body">
          {availablePlayers.map((player, index) => (
<tr
            data-testid={`available-${player.name.split(" ").join("-")}-row`}
            key={index}
.....................................................................
<button
data-testid={`available-${player.name.split(" ").join("-")}-select`}
              onClick={() => addPlayer(index)}
              disabled={disabledSelect[player.name]}
              aria-label="Select player"
```

```
useEffect(() => {
if (!selectedDate) {
setFilteredTxns(txns);}
}, [selectedDate, txns]);
const handleDateChange = (e) => {
setSelectedDate(e.target.value);};
const handleFilter = () => {
if(!selectedDate)
return;
const filtered = txns.filter((txn) => txn.date === selectedDate);
setFilteredTxns(filtered);}
const sortByAmount = () => {
const sorted = [...filteredTxns].sort((a,b) => a.amount - b.amount);
setFilteredTxns(sorted);}

defaultValue="2019-11-29"
onChange={handleDateChange}

<button className="small"onClick={handleFilter}>Filter</button>

<tbody>
{filteredTxns.map((txn,index) => (
<tr key = {index}>
<td>{txn.date}</td>
<td>{txn.description}</td>
<td>{txn.type === 1 ? "Debit" : "Credit"}</td>
<td>{txn.amount}</td>
<td>{txn.balance}</td>
</tr>))}</tbody>
```

9.Pokemon

pokemonDetails.js

```
const [id, setId] = React.useState('');
const getPokemonById = async (pid) => {
if (pid < 1 || pid > 151) {
alert('Pokemon ID must be between 1 and 151');
return;
}
const res = await
fetch(`https://jsonmock.hackerrank.com/api/pokemon?id=${pid}`);
const data = await res.json();
if (data.data) {
// API returns single object, not array
setPokemon(data.data); } };
const handleNext = () => {
if (pokemon && pokemon.next_evolution) {
const nextId = pokemon.next_evolution[0].num;
getPokemonById(parseInt(nextId));
setId(parseInt(nextId)); } };

const handlePrev = () => {
if (pokemon && pokemon.prev_evolution) {
const prevId = pokemon.prev_evolution[0].num;
getPokemonById(parseInt(prevId));
setId(parseInt(prevId)); } };

return (
<div className="mt-50 layout-column justify-content-center align-
items  center">
{pokemon && <Pokemon pokemon={pokemon} />}
{pokemon && (
<p>
<button
data-testid="pokemon-prev"
onClick={handlePrev}
disabled={!pokemon.prev_evolution}
>
Previous Evolution
</button>
```

```
data-testid={`available-${player.name.split(" ").join("-")}-select`}
                    onClick={() => addPlayer(index)}
                    disabled={disabledSelect[player.name]}
                    aria-label="Select player"
                    className="w-12 sm:w-[68px] px-3 sm:px-6 py-2 sm:py-3 flex
items-center justify-center gap-3 border-b border-[#ECEFF2] hover:bg-gray-50"
>
.....

              <th>Action</th>
</tr>
</thead>
<tbody data-testid="selected-players-table-body">
              {/* Empty - no players selected */}
              {selectedPlayers.map((player, index) => {
                const testIdBase = `selected-${player.name.split(" ").join("-")}`;
                return (
<tr key={index} data-testid={`${testIdBase}-row`} className="border-b border-
[#ECEFF2]">
<td>{player.name}</td>
<td>{player.type}</td>
<td>{player.battingSkill}</td>
<td>{player.bowlingSkill}</td>
<td>
<button
                    data-testid={`${testIdBase}-remove`}
                    onClick={() => removePlayer(player.name)}
>
                    Remove
</button>
</td>
</tr>
              );
              })}
</tbody>
.....
```

10.Image Preview

ImagePreview.js

```
import React ,{useState}from "react";
import HiddenImageDiv from "./HiddenImageDiv";
const ImagePreview = ({images=[]}) => {
const [imageStates, setImageStates] = useState(images);

 const toggleVisibility = (index) => {
  const updated = [...imageStates];
  updated[index].visible = !updated[index].visible;
  setImageStates(updated);
 };

 const showAll = () => {
  setImageStates(imageStates.map((img) => ({ ...img, visible: true })));
 };

 const hideAll = () => {
  setImageStates(imageStates.map((img) => ({ ...img, visible: false })));
 };
.....

      <div
        className="layout-row justify-content-around"
        data-testid="images-div"
>

          {imageStates.map((img, index) =>
            img.visible ? (
<img
              key={index}
              src={img.src}
              height={200}
              width={300}
              alt={img.alt || ""}
```

```
disabled={!pokemon.prev_evolution}
>
Previous Evolution
</button>
<button
data-testid="pokemon-next"
onClick={handleNext}
disabled={!pokemon.next_evolution}
>
Next Evolution
</button>
</p>
)}
<p>
<input
data-testid="id-input"
value={id}
onChange={(e) => setId(e.target.value)}
type="number"
placeholder="Pokemon Id"
/>
<button
data-testid="random-pokemon"
onClick={() => getPokemonById(parseInt(id))}
className="text"
>
Get Pokemon
</button>
</p>
</div>
);
};
```

Pokemon.js

```
export const Pokemon = ({ pokemon }) => {
if (!pokemon) return null;
return (
<div className="card outlined px-50 py-30" style={{ width: '900px' }}>
<div className="card-header"
<strong>ID: </strong>
<span data-testid="pokemon-id">{pokemon.num}</span>
</p>
<p>
<strong>Name: </strong>
<span data-testid="pokemon-name">{pokemon.name}</span>
</p>
<p>
<strong>Type: </strong>
</p>
<ul data-testid="pokemon-types">
{pokemon.type.map((t) => (
<li key={t} data-testid={`pokemon-type-${t}`}>
{t}
</li>
))}
</ul>
<p>
<strong>Height: </strong>
<span data-testid="pokemon-height">{pokemon.height}</span>
</p>
<p>
<strong>Weight: </strong>
<span data-testid="pokemon-weight">{pokemon.weight}</span>
</p>
<p>
<strong data-testid="pokemon-weeknesses">Weaknesses:</strong>
</p>
<ol>
{pokemon.weaknesses.map((w) => (
<li key={w}>{w}</li>
```

```
                src={img.src}
                height={200}
                width={300}
                alt={img.alt || ""}
                onClick={() => toggleVisibility(index)}
              />
            ) : (
<HiddenImageDiv key={index} onClick={() => toggleVisibility(index)} />
            )
          )}

      </div>
</section>
<section className="card-actions justify-content-center">
<button data-testid="show-all-btn" onClick={showAll}>Show all</button>
<button className="danger" data-testid="hide-all-btn" onClick={hideAll}>
        Hide all
</button>
</section>
```

.....

11.Team and Channels List Application

Team.js

```
import React, { useState } from 'react'

import './Team.css'

const Team = ({ team, id ,onAddChannel,onRemoveChannel}) => {
 const [channelName,setChannelName] = useState('');
 const isChannelNameValid =()=>{
  return(
   channelName.trim().length > 0 &&
   !team.channels.some(channel => channel.name === channelName.trim())
  );
 };

 const handleAddChannel = () =>{
  if(isChannelNameValid()){
   onAddChannel(id,channelName.trim());
   setChannelName('');
  }
 };
```

.....

```
<input
        placeholder='Enter Channel Name'
        className="channel-name-input w-45 px-13"
        data-testid={ 'channel-name-input-' + id }
        value = {channelName}
        onChange={e => setChannelName(e.target.value)}
       />
```

.....

```
<button
        className='channel-name-btn x-small w-35 h-30 pa-6 ma-0 ml-6'
        data-testid={ 'add-channel-btn-' + id }
        onClick={handleAddChannel}
        disabled ={!isChannelNameValid()}
>
        Add Channel
</button>
```

.....

```
<button
          data-testid={ 'remove-channel-button-' + id + channel.id }
          className='icon-only x-small danger ma-0 pa-0'
          onClick={() => onRemoveChannel(id,channel.id)}
>
<i className="material-icons">delete</i>
</button>
```

.....

TeamList.js

```
import React, { useState } from 'react'
import Team from './Team'
```

```
</p>
<ol>
{pokemon.weaknesses.map((w) => (
<li key={w}>{w}</li>
))}
</ol>
```

TeamList.js

```
import React, { useState } from 'react'
import Team from './Team'

import './TeamList.css'

const TeamList = () => {

  const [teams,setTeams] = useState([
    { name: 'Team1',
      channels: [
        { name: 'Channel1',
          id: 1
        },
.....................................................................
const[teamName,setTeamName] = useState('');
  const[channelIdCounter,setChannelIdCounter] = useState(0);
  const isTeamNameValid = () =>{
    return(
      teamName.trim().length > 0 && !teams.some(team => team.name ===
teamName.trim())
    );
  };

const handleAddTeam = () => {
    if (isTeamNameValid()) {
      setTeams([...teams, { name: teamName.trim(), channels: [] }]);
      setTeamName('');
    }
  };

  const handleAddChannel = (teamIndex,channelName) =>{
    const newChannel = {
      id:channelIdCounter,
      name:channelName
    };
    const updatedTeams =[...teams];
    updatedTeams[teamIndex].channels.push(newChannel);
    setTeams(updatedTeams);
    setChannelIdCounter(channelIdCounter+1);
  };
  const handleRemoveChannel = (teamIndex,channelId) => {
    const updatedTeams = [...teams];
    updatedTeams[teamIndex].channels =
updatedTeams[teamIndex].channels.filter(
      channel => channel.id !== channelId
    );
    setTeams(updatedTeams)
  }
  return (
.....................................................................

       <div className='layout-column' data-testid='team-list'>
         { teams.map((team, id) => (
<Team
            key={ id }
            id={ id }
            team={ team }
            onAddChannel={handleAddChannel}
            onRemoveChannel={handleRemoveChannel}
          />
         ))}
</div>
.....................................................................
<div className='layout-row'>
<input
          placeholder='Enter Team Name'
          className='team-list-input w-75'
          data-testid='team-name-input'
          value = {teamName}
```

```
        onChange={e => setTeamName(e.target.value)}
       />
<button
        className='team-list-btn x-small w-35 h-30 pa-6 ma-0 ml-6'
        data-testid='add-team-btn'
        onClick ={handleAddTeam}
        disabled = {!isTeamNameValid()}
>
.....
```