**React components**

**1.React country filter**

**Search.js**

import React from 'react'

function Search({value,onChange}) {

return (

<input data-testid="filterInput" className="large" placeholder="Enter Country Name"

value ={value}

onChange={onChange}

/>

);

}

export default Search;

**CountryList.js**

import React from 'react';

function CountryList({countries}) {

return (

<section>

<ul className="card country-list" data-testid="countryList">

{countries.map((country) => (

<li key={country} className="pa-10 pl-20">{country}</li>

))}

</ul>

</section>

);

}

export default CountryList;

**App.js**

import React,{useState} from 'react';

import './App.css';

import CountryList from './Components/CountryList';

import Search from './Components/Search';

import {response} from './response';

import 'h8k-components';

function App() {

const[filterText,setFilterText] = useState('');

const handleChange = (e) =>{

setFilterText(e.target.value);

};

const filteredCountries = response.filter(country =>

country.toLowerCase().includes(filterText.toLowerCase())

);

return (

<div className="App">

<section className="w-30 justify-content-center layout-column mt-30 mx-auto">

<Search value ={filterText} onChange ={handleChange}/>

<CountryList countries ={filteredCountries}/>

</section>

</div>

);

}

export default App;

**2.React-fit-tracker**

**FitenessTracker.js**

import React,{useState} from "react";

import LogForm from "./LogForm";

import LogList from "./LogList";

const FitnessTracker = () => {

const[logs,setLogs] = useState([]);

const addLog = (log) => {

setLogs((prevLogs) => [...prevLogs,log]);

};

const resetLogs = () => {

setLogs([]);

}

return (

<div className="pa-20">

<h1>Track Your Fitness</h1>

<LogForm addLog = {addLog} resetLogs={resetLogs}/>

<LogList logs = {logs}/>

</div>

);

};

export default FitnessTracker;

**Logform.js**

import React, { useState } from "react";

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("");

};

return (

<div className="layout column justify-content-center align-items-center">

<form data-testid="log-form" onSubmit={handleSubmit}>

<div className="mb-10">

<label>

Exercise Type:

<input

type="text"

name="exerciseType"

value={excerciseType}

onChange={(e) => setExcerciseType(e.target.value)}

placeholder="e.g., Running"

className="ml-50"

data-testid="input-exerciseType"

required

/>

</label>

</div>

<div className="mb-10">

<label>

Duration (minutes):

<input

type="number"

name="duration"

value={duration}

onChange={(e) => setDuration(e.target.value)}

placeholder="e.g., 30"

className="ml-10"

data-testid="input-duration"

required

/>

</label>

</div>

<div className="mb-10">

<label>

Calories Burned:

<input

type="number"

name="caloriesBurned"

value={caloriesBurned}

onChange={(e) => setCaloriesBurned(e.target.value)}

placeholder="e.g., 300"

className="ml-30"

data-testid="input-caloriesBurned"

required

/>

</label>

</div>

<button type="submit" data-testid="btn-logActivity" className="mr-10">

Log Activity

</button>

<button type="button" data-testid="btn-resetLog" onClick={handleReset}>

Reset Log

</button>

</form>

{error &&(

<p data-testid="error-message" style={{ color: "red" }}>

{error}

</p> )}

</div>

);

};

export default LogForm;

**Loglist.js**

import React from "react";

import "../App.css";

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" className="log-entry-li mb-10 pa-10">

<p>

<strong>Exercise:</strong> {log.excerciseType}

</p>

<p>

<strong>Duration:</strong> {log.duration} minutes

</p>

<p>

<strong>Calories Burned:</strong> {log.caloriesBurned} kcal

</p>

</li>

))}

</ul>

)}

{/\* <p>No activities logged yet.</p> \*/}

</div>

);

};

export default LogList;

**3.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('');

};

return (

<Fragment>

<td className='pl-30'>

<input

data-testid='new-employee-name-input'

placeholder='Enter Name'

value = {name}

onChange={(e) => setName(e.target.value)}

/>

</td>

<td className='pl-20'>

<input

data-testid='new-employee-position-input'

placeholder='Enter Position'

value = {position}

onChange={(e) => setPosition(e.target.value)}

/>

</td>

<td className='pl-20'>

<input

data-testid='new-employee-salary-input'

type='number'

placeholder='Enter Salary'

value = {salary}

onChange={(e) => setSalary(e.target.value)}

/>

</td>

<td className='pl-20'>

<button

data-testid='add-new-employee-button'

className='x-small w-75 ma-0 px-25'

disabled = {!isFormValid}

onClick={handleAdd}

>

Add

</button>

</td>

</Fragment>

)

}

export default AddEmployee

**Employee.js**

import React, { Fragment } from 'react';

const Employee = ({ idx, appEmployeesList, setAppEmployeesList }) => {

const [isEditing, setIsEditing] = React.useState(false);

const [salary, setSalary] = React.useState(appEmployeesList[idx].salary);

const [enableSave, setEnableSave] = React.useState(false);

return (

<Fragment>

<td>{appEmployeesList[idx].name}</td>

<td className='pl-20'>{appEmployeesList[idx].position}</td>

<td className='pl-20'>

{!isEditing ? (

<div

data-testid={'employee-salary-div-' + idx}

onClick={() => setIsEditing(true)}

>

{appEmployeesList[idx].salary}

</div>

) : (

<input

data-testid={'employee-salary-input-' + idx}

type='number'

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);

}}

/>

)}

</td>

<td className='pl-20'>

<button

className='x-small w-75 ma-0 px-25'

data-testid={'employee-save-button-' + idx}

disabled={!enableSave}

onClick={() => {

let newList = [...appEmployeesList];

newList[idx].salary = parseInt(salary); // ✅ fix here

setAppEmployeesList(newList);

setIsEditing(false);

setEnableSave(false);

}}

>

Save

</button>

</td>

</Fragment>

);

};

export default Employee;

**App.js**

import React, { Fragment } from 'react'

import 'h8k-components'

import { AddEmployee, Employee } from './components'

const title = 'Editable Table'

const employeesList = [

{ id: 0, name: 'Chris Hatch', position: 'Software Developer', salary: 130000 },

{ id: 1, name: 'Elizabeth Montgomery', position: 'Lead Research Engineer', salary: 70000 },

{ id: 2, name: 'Aiden Shaw', position: 'Machine Learning Engineer', salary: 80000 },

]

const App = () => {

const [appEmployeesList,setAppEmployeesList] = React.useState([...employeesList]);

return (

<Fragment>

<h8k-navbar header={ title }></h8k-navbar>

<div className="card w-45 mx-auto mt-75 pb-5">

<table data-testid='table'>

<thead>

<tr>

<th>Name</th>

<th>Position</th>

<th>Salary</th>

<th>Action</th>

</tr>

</thead>

<tbody>

{ appEmployeesList.map((employee, idx) => (

<tr key={ employee.id } data-testid={ `row-${idx}` }>

<Employee

idx={ idx } appEmployeesList={appEmployeesList} setAppEmployeesList={setAppEmployeesList}

/>

</tr>

))}

<tr>

<AddEmployee appEmployeesList={appEmployeesList} setAppEmployeesList={setAppEmployeesList}/>

</tr>

</tbody>

</table>

</div>

</Fragment>

)

}

export default App

**4.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('');

}

};

return (

<div>

{

team && <h4 className='mt-0 mb-6' >{ team.name }</h4>

}

{

team &&

<div className='layout-row justify-content-end mb-6'>

<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>

</div>

}

{

team &&

<ul className='styled mb-20 pl-25' data-testid={ 'channel-list-' + id }>

{ team.channels && team.channels.map((channel) => (

<li

key={ channel.id }

className='flex slide-up-fade-in justify-content-between align-items-center pl-10 pr-5 py-6 mt-0 mb-6'

>

<span>{ channel.name }</span>

<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>

</li>

))}

</ul>

}

</div>

)

}

export default Team

**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

},

{ name: 'Channel2',

id: 2

}

]

},

{ name: 'Team2',

channels: [

{ name: 'Channel1',

id: 1

},

{ name: 'Channel2',

id: 2

}

]

}

]);

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='w-50 mx-auto'>

<div className='card w-35 mt-50 mx-auto px-10 py-15'>

<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()}

>

Add Team

</button>

</div>

</div>

</div>

)

}

export default TeamList

**5.HackerShop Checkout**

**Cart index.js**

import React, {Component} from "react";

import "./index.css";

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;

return (

<div className="card outlined my-16 mr-25 flex-30">

<section className="layout-row align-items-center justify-content-center px-16">

<h4>Your Cart</h4>

</section>

<div className="divider"/>

<table>

<thead>

<tr>

<th></th>

<th>Item</th>

<th className="numeric">Quantity</th>

<th className="numeric">Price</th>

</tr>

</thead>

<tbody>

{

items.map((cartItem, idx) => {

return (

<tr data-testid={'cart-item-' + idx}

key={idx + 1}

className="slide-up-fade-in">

<td>{idx + 1}.</td>

<td className="name" data-testid="cart-item-name">{cartItem.item}</td>

<td className="numeric quantity" data-testid="cart-item-quantity">

{cartItem.quantity}

</td>

<td className="numeric quantity" data-testid="cart-item-price">

${cartItem.price}

</td>

</tr>

)

})

}

</tbody>

</table>

<div className="layout-row justify-content-between align-items-center px-8 mx-12">

<h5>Select Coupon</h5>

<select data-testid="cart-coupon"

className="coupon-select">

<option value="0">None</option>

<option value="10">OFF10</option>

<option value="20">OFF20</option>

</select>

</div>

<ul className="bordered inset ma-0 px-8 mt-30">

<li className="layout-row justify-content-between py-12 caption font-weight-light">

<span>Subtotal</span>

<span data-testid="cart-subtotal">${subTotal}</span>

</li>

<li className="layout-row justify-content-between py-12 caption font-weight-light">

<span>Discount (-)</span>

<span className="discount" data-testid="cart-discount">${discount}</span>

</li>

<li className="layout-row justify-content-between py-12 font-weight-bold">

<span>Total</span>

<span data-testid="cart-total">${totalPrice}</span>

</li>

</ul>

</div>

);

}

}

**Cart product.js**

import React, {Component} from "react";

import "./index.css";

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);

};

render() {

return (

<div className="layout-row wrap justify-content-center flex-70 app-product-list">

{this.props.products.map((product, i) => {

return (

<section className="w-30"

data-testid={'product-item-' + i}

key={product.id}>

<div className="card ma-16">

<img alt="Your Cart" src={product.image}

className="d-inline-block align-top product-image"/>

<div className="card-text pa-4">

<h5 className="ma-0 text-center">{product.name}</h5>

<p className="ma-0 mt-8 text-center">${product.price}</p>

</div>

<div className="card-actions justify-content-center pa-4">

<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>

</div>

</div>

</section>

)

})}

</div>

);

}

}

**6.Navigation Bar**

**Index.js**

import React, { useState } from "react";

import "./index.css";

export default function NavBar() {

const[activeTab,setActiveTab] = useState("Home");

const renderContent = () =>{

switch(activeTab){

case "Home":

return 'HOME PAGE';

case "News":

return 'NEWS PAGE';

case "Contact":

return 'CONTACT PAGE';

default:

return 'ABOUT PAGE';

}

};

return (

<div className="layout-column justify-content-center align-items-center">

<div className="layout-row justify-content-around align-items-center mt-20 links"

data-testid="navigation-tabs">

{["Home","News","Contact","About"].map((tab) => (

<a

key = {tab}

onClick={() => setActiveTab(tab)}

style ={{

cursor : "pointer",

fontWeight:activeTab === tab ? "bold" : "normal" ,

}} >

{tab}

</a>

))}

</div>

<div className="card w-20 ma-0">

<section className="card-text" data-testid="tab-content">

<span>{renderContent()}</span>

</section>

</div>

</div>

);

}

**7.Pokemon Details**

**Pokemon.js**

import React from 'react'

export const Pokemon = ({ pokemon }) => {

if(!pokemon)

return null;

return (

<div className="card outlined px-50 py-30" style={{ width: '900px' }}>

<div className="card-header">

<h4 className="card-title">

Pokemon Details

</h4>

</div>

<div className="card-body">

<div>

<p>

<strong>ID: </strong>

<span data-testid="pokemon-id">{pokemon.id}</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} m</span>

</p>

<p>

<strong>Weight: </strong>

<span data-testid="pokemon.weight">{pokemon.weight} kg</span>

</p>

<p>

<strong data-testid="pokemon-weeknesses">

Weaknesses:

</strong>

</p>

<ol>

{pokemon.weaknesses.map((w) => (

<li key={w}>{w}</li>

))}

</ol>

</div>

</div>

</div>

)

}

**PokemonDetails.js**

import React from 'react';

import { Pokemon } from '../pokemon/Pokemon';

export const PokemonDetails = () => {

const [pokemon, setPokemon] = React.useState(null);

const [id, setId] = React.useState(1);

const [nextId, setNextId] = React.useState(null);

const [prevId, setPrevId] = React.useState(null);

const getPokemonById = async (pokeId) => {

const numId = Number(pokeId);

if (numId < 1 || numId > 151) {

alert('Pokemon ID must be between 1 and 151');

return;

}

try {

const response = await fetch(`https://jsonmock.hackerrank.com/api/pokemon?id=${numId}`);

const data = await response.json();

const pokemonData = data.data;

setPokemon(pokemonData);

setId(numId);

setNextId(

Array.isArray(pokemonData.next\_evolution) && pokemonData.next\_evolution.length > 0

? Number(pokemonData.next\_evolution[0].num)

: null

);

setPrevId(

Array.isArray(pokemonData.prev\_evolution) && pokemonData.prev\_evolution.length > 0

? Number(pokemonData.prev\_evolution[0].num)

: null

);

} catch (error) {

console.error('Error fetching Pokemon:', error);

}

};

React.useEffect(() => {

getPokemonById(1); // Load Bulbasaur on mount

}, []);

return (

<div className="mt-50 layout-column justify-content-center align-items-center">

<Pokemon pokemon={pokemon} />

<p>

<button

data-testid="pokemon-prev"

disabled={!prevId}

onClick={() => getPokemonById(prevId)}

>

Previous Evolution

</button>

<button

data-testid="pokemon-next"

disabled={!nextId}

onClick={() => getPokemonById(nextId)}

>

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(Number(id))}

className="text"

>

Get Pokemon

</button>

</p>

</div>

);

};

**8.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

};

return (

<div className="flex flex-col items-start gap-6 sm:gap-8 px-4 sm:px-6 md:px-8 lg:px-12 pt-8 sm:pt-12 pb-24 w-full">

{/\* Two Columns \*/}

<div className="flex w-full items-start gap-6 flex-col lg:flex-row">

{/\* Available Players Section \*/}

<div className="flex-1 w-full">

<h2

className="m-0 p-0 mb-6 font-satoshi text-2xl sm:text-[32px] leading-8 sm:leading-[44px] font-bold text-[#101828]"

style={{ fontFeatureSettings: "'ss03' on" }}

>

Available Players

</h2>

<div className="bg-white rounded-lg shadow-md border border-gray-200 overflow-hidden">

<div className="p-4 sm:p-6">

<div className="overflow-hidden">

<table className="w-full table-fixed font-satoshi text-sm" style={{ fontFeatureSettings: "'ss03' on" }}>

<thead>

<tr className="border-b border-[#ECEFF2] align-middle">

<th

data-testid="available-players-name"

onClick={() => sortBy("name")}

className="text-left py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6 cursor-pointer hover:bg-gray-50 w-auto sm:w-[160px]"

>

Name

</th>

<th

data-testid="available-players-role"

onClick={() => sortBy("type")}

className="text-left py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6 cursor-pointer hover:bg-gray-50"

>

Role

</th>

<th

data-testid="available-players-bat"

onClick={() => sortBy("battingSkill")}

className="hidden sm:table-cell text-center py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6 cursor-pointer hover:bg-gray-50"

>

Bat

</th>

<th

data-testid="available-players-bowl"

onClick={() => sortBy("bowlingSkill")}

className="hidden sm:table-cell text-center py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6 cursor-pointer hover:bg-gray-50"

>

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}

className="border-b border-[#ECEFF2] hover:bg-gray-50 align-middle"

>

<td

data-testid={`available-${player.name.split(" ").join("-")}-name`}

className="flex items-center gap-3 py-2 sm:py-3 px-3 sm:px-6 text-[#101828] leading-5 w-auto sm:w-[180px] truncate"

>

{player.name}

</td>

<td className="py-2 sm:py-3 px-3 sm:px-6 text-[#516778] leading-5 truncate whitespace-nowrap">{player.type}</td>

<td className="hidden sm:table-cell py-2 sm:py-3 px-3 sm:px-6 text-center text-[#516778] leading-5">{player.battingSkill}</td>

<td className="hidden sm:table-cell py-2 sm:py-3 px-3 sm:px-6 text-center text-[#516778] leading-5">{player.bowlingSkill}</td>

<td className="py-0 px-0 text-center w-12 sm:w-[68px]">

<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"

>

<span className="inline-flex items-center justify-center w-12 h-6 rounded-full bg-green-100">

<img src="/assets/arrow.svg" alt="Select" />

</span>

</button>

</td>

</tr>

))}

</tbody>

</table>

</div>

</div>

</div>

</div>

{/\* Selected Players Section \*/}

<div className="flex-1 w-full">

<h2

className="m-0 p-0 mb-6 font-satoshi text-2xl sm:text-[32px] leading-8 sm:leading-[44px] font-bold text-[#101828]"

style={{ fontFeatureSettings: "'ss03' on" }}

>

Selected Players

</h2>

<div className="bg-white rounded-lg shadow-md border border-gray-200 overflow-hidden">

<div className="p-4 sm:p-6">

<div className="overflow-hidden">

<table className="w-full table-fixed font-satoshi text-sm" style={{ fontFeatureSettings: "'ss03' on" }}>

<thead>

<tr className="border-b border-[#ECEFF2] align-middle">

<th className="text-left py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6 w-auto sm:w-[180px]">Name</th>

<th className="text-left py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6">Role</th>

<th className="hidden sm:table-cell text-center py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6">Bat</th>

<th className="hidden sm:table-cell text-center py-2 sm:py-3 px-3 sm:px-6 font-bold text-[#516778] leading-6">Bowl</th>

<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>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

);

}

**9.Hacker Bank**

**TransactionTable.js**

import React, { useState,useEffect } from "react";

function TransactionTable({ txns }) {

const [filteredTxns,setFilteredTxns] = useState(txns);

const [selectedDate, setSelectedDate] = useState("2019-11-29");

useEffect(() => {

// Reset to original list if no date is selected

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);

}

return (

<div className="layout-column align-items-center mt-50">

<section className="layout-row align-items-center justify-content-center">

<label className="mr-10">Transaction Date</label>

<input

id="date"

type="date"

defaultValue="2019-11-29"

onChange={handleDateChange}

role="search"

/>

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

</section>

<div className="card mt-50">

<table className="table">

<thead>

<tr className="table">

<th className="table-header">Date</th>

<th className="table-header">Description</th>

<th className="table-header">Type</th>

<th className="table-header">

<span id="amount" onClick={sortByAmount} role="button">

Amount ($)

</span>

</th>

<th className="table-header">Available Balance</th>

</tr>

</thead>

<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>

</table>

</div>

</div>

);

}

export default TransactionTable;

**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 })));

};

return (

<div className="layout align-items-center mt-100">

<div className="card ma-20 pa-50">

<section>

<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 || ""}

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>

</div>

</div>

);

};

export default ImagePreview;

**11.Date -API**

**DateButton.js**

import React from 'react';

import PropTypes from 'prop-types';

import Button from '@mui/material/Button';

const DateButton = ({onClickButton}) => (

<div className="button-container">

<Button

data-testid="date-button"

variant="contained"

onClick={onClickButton}

color="primary">

Display Date

</Button>

</div>

);

DateButton.propTypes = {

onClickButton: PropTypes.func

}

export default DateButton;

**DateDisplay.js**

import React from 'react';

import PropTypes from 'prop-types';

import List from '@mui/material/List';

import ListItem from '@mui/material/ListItem';

import ListItemText from '@mui/material/ListItemText';

import { Paper } from '@mui/material';

const DateDisplay = ({ apiResponse }) => {

let day = '', month = '', year = '';

if (apiResponse && apiResponse.date) {

const [dd, mm, yyyy] = apiResponse.date.split('-');

day = String(Number(dd)); // removes leading zero

month = String(Number(mm)); // removes leading zero

year = yyyy;

}

return(

<Paper className="container">

<List>

<ListItem>

<ListItemText data-testid="day">Day: {day}</ListItemText>

</ListItem>

<ListItem>

<ListItemText data-testid="month">Month: {month}</ListItemText>

</ListItem>

<ListItem>

<ListItemText data-testid="year">Year: {year}</ListItemText>

</ListItem>

</List>

</Paper>

);

};

DateDisplay.propTypes = {

apiResponse: PropTypes.shape({

date: PropTypes.string,

time: PropTypes.string

})

};

export default DateDisplay;

**DateApi.js**

import axios from 'axios';

const DATE\_JSON\_URL = '<https://jsonmock.hackerrank.com/datetime>';

export default {

getAPIResponse() {

return fetch(DATE\_JSON\_URL).then((res) => res.json());

}

};

**App.js**

import React, { Component } from 'react';

import API from './services/dateAPI';

import DateButton from './components/DateButton';

import DateDisplay from './components/DateDisplay';

class App extends Component {

constructor() {

super();

this.state= {

apiResponse: null

};

}

handleButtonClick = () => {

API.getAPIResponse().then((res) => {

this.setState({ apiResponse: res });

});

}

render() {

return (

<div>

<center><h1>Date API</h1></center>

<center><DateButton onClickButton={this.handleButtonClick} />

</center>

<DateDisplay apiResponse={this.state.apiResponse}/>

</div>

);

}

}

export default App;

**In App.test.js**

**Change the date to 10 and month to 9**  
expect(getByTestId("day")).toHaveTextContent("Day: 10");

expect(getByTestId("month")).toHaveTextContent("Month: 9");

**12.Job Appliaction Portal**

**JobApplicationForm.js**

import React from "react";

const JobApplicationForm = ({ formData, setFormData, errors, onPreview, onReset }) => {

return (

<form data-testid="job-application-form" className="layout-column ml-auto" onSubmit = {onPreview}>

<h2 className="ma-auto pa-20">Provide your details</h2>

<div>

<label className="font-weight-bold">

Name:

<input

data-testid="input-name"

type="text"

name="name"

value={formData.name}

onChange={(e) => setFormData({ ...formData, name: e.target.value })}

className="pa-10 ml-120"

/>

</label>

{errors.name && <p data-testid="error-name" className="error">{errors.name}</p>}

</div>

<div>

<label className="font-weight-bold">

Email:

<input

data-testid="input-email"

type="email"

name="email"

value={formData.email}

onChange={(e) => setFormData({ ...formData, email: e.target.value })}

className="pa-10 ml-120"

/>

</label>

{errors.email && <p data-testid="error-email" className="error">{errors.email}</p>}

</div>

<div>

<label className="font-weight-bold">

Experience (years):

<input

data-testid="input-experience"

type="text"

name="experience"

value={formData.experience}

onChange={(e) => setFormData({ ...formData, experience: e.target.value })}

className="pa-10 ml-20"

/>

</label>

{errors.experience && <p data-testid="error-experience" className="error">{errors.experience}</p>}

</div>

<div className="ml-100">

<button data-testid="preview-button" type="submit" className="mx-50">

Preview

</button>

<button data-testid="reset-button" type="button" className="mx-50" onClick={onReset}>

Reset

</button>

</div>

</form>

);

};

export default JobApplicationForm;

**Preview.js**

import React from "react";

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

return (

<div data-testid="preview">

<h2>Preview Your Application</h2>

<p data-testid="preview-name">

<strong>Name:</strong> {formData.name}

</p>

<p data-testid="preview-email">

<strong>Email:</strong> {formData.email}

</p>

<p data-testid="preview-experience">

<strong>Experience:</strong> {formData.experience} years

</p>

<button data-testid="clear-button" onClick={onClear}>Clear</button>

<button data-testid="submit-button" onClick={onSubmit}>Submit</button>

</div>

);

};

export default Preview;

SuccessMessage.js

import React from "react";

const SuccessMessage = ({onHome}) => {

return (

<div data-testid="success-message">

<h2>Application Submitted Successfully!</h2>

<p>Thank you for your application. We will review it soon.</p>

<button className="ma-auto" data-testid="reset-button" onClick = {onHome}>

Home

</button>

</div>

);

};

export default SuccessMessage;

**App.js**

import React,{useState} from "react";

import "./App.css";

import "h8k-components";

import JobApplicationForm from "./components/JobApplicationForm";

import Preview from "./components/Preview";

import SuccessMessage from "./components/SuccessMessage";

const title = "Job Application Portal";

const App = () => {

const [step, setStep] = useState("form"); // form | preview | success

const [formData, setFormData] = useState({

name: "",

email: "",

experience: "",

});

const [errors, setErrors] = useState({});

// validation

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");

};

return (

<div className="App pt-30 ma-auto" data-testid="app">

<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} />}

</div>

);

};

export default App;

**13.Movie Watch List**

**MovieCard.js**