



# **NORTH SOUTH UNIVERSITY**

*Center of Excellence in Higher Education*

## **CSE299: Junior Design**

### **Week: 03 Progress Report**

**Submitted to:** Tanzilah Noor Shabnam  
**Submitted on:** 22/10/2024  
**Team Name:** Machine Minds  
**Project Title:** Flood Guard



**Task completed:** Victim Portal [UI, Backend and Database]

### **Team members**

Name	ID
Sabbir Hossain	2131272042
Md. Misbah Khan	2132086642
Aritra Islam Saswato	2132629642

## Languages and Tools

Programming language: JavaScript, SQL

Framework:

Name	Language
React	JavaScript and typescript
MySQL	SQL

Middleware:

Name	Use	Links
Leaflet	for map location and direction	<a href="https://leafletjs.com/">https://leafletjs.com/</a>

Collaboration and storing code: GitHub

Contribution:

Name	Contribution
Md. Misbah Khan	Login & Signup[UI & firebase], map component for exact location and direction.
Sabbir Hossain	Victim table, Shelter Table, Modal, Badge, Table, Tab reusable component.
Aritra Islam Saswato	Donations Table, Backend code for get Donation, Victim and Shelter table and Database Bug fix.

External API:

Name	Use	Links
Open Cage	converting latitude and longitude to location name	<a href="https://opencagedata.com/api">https://opencagedata.com/api</a>
Open Street Map API v0.6	Accessing and editing map data	<a href="https://wiki.openstreetmap.org/wiki/API_v0.6">https://wiki.openstreetmap.org/wiki/API_v0.6</a>

## Snippets of the Code:

```
1 import React, { useState } from "react";
2 import Modal from "../modal";
3 import MapComponent from "../MapComponent";
4 import DirectionsMap from "../DirectionsMap";
5
6 const AdditionalInfoTable = ({
7   additionalInfo,
8   row,
9   mapComponent: MapComponentProp,
10  directionsComponent: DirectionsComponentProp,
11  tableTitle = "Additional Information",
12  children,
13 }) => {
14   const [isMapOpen, setIsMapOpen] = useState(false);
15   const [isDirectionOpen, setIsDirectionOpen] = useState(false);
16
17   return (
18     <tr>
19       <td colSpan={6}>
20         <div className="p-2 m-2 pb-6 w-3/4 mx-auto rounded-xl bg-secondary">
21           <h4 className="text-center font-museo text-3xl my-6">{tableTitle}</h4>
22           <table className="w-11/12 mx-auto table-auto border-collapse">
23             <tbody>
24               {additionalInfo.map((info) => (
25                 <tr className="border-b" key={info.label}>
26                   <td className="p-1">{info.label}</td>
27                   <td className="text-end">
28                     {info.label === "Exact Location" ? (
29                       <>
30                         <button
31                           className="text-blue-500 underline mr-4"
32                           onClick={() => setIsMapOpen(true)}
33                         >
34                           {info.value[0]}
35                         </button>
36                         <button
37                           className="text-blue-500 underline"
38                           onClick={() => setIsDirectionOpen(true)}
39                         >
40                           {info.value[1]}
41                         </button>
42                       </>
43                     ) : (
44                       info.value
45                     )}
46                   </td>
47                 </tr>
48               )]}
49             </tbody>
50           </table>
51
52           {children}
53
54           <Modal isOpen={isMapOpen} onClose={() => setIsMapOpen(false)}>
55             <h3 className="text-center mb-4">View Location</h3>
56             {MapComponentProp ? (
57               <MapComponentProp
58                 latitude={row.latitude}
59                 longitude={row.longitude}
60               />
61             ) : (
62               <MapComponent latitude={row.latitude} longitude={row.longitude} />
63             )}
64           </Modal>
65
66           <Modal
67             isOpen={isDirectionOpen}
68             onClose={() => setIsDirectionOpen(false)}
69           >
70             <h3 className="text-center mb-4">Get Directions</h3>
71             {DirectionsComponentProp ? (
72               <DirectionsComponentProp
73                 destination={[row.latitude, row.longitude]}
74               />
75             ) : (
76               <DirectionsMap destination={[row.latitude, row.longitude]} />
77             )}
78           </Modal>
79         </div>
80       </td>
81     </tr>
82   );
83 };
84
85 export default AdditionalInfoTable;
86
```

*Additional Info Table*

```

1 import React, { useEffect, useState } from "react";
2 import { MapContainer, TileLayer } from "react-leaflet";
3 import "leaflet/dist/leaflet.css";
4 import L from "leaflet";
5 import "leaflet-routing-machine";
6 import { ThreeCircles } from "react-loader-spinner";
7
8 const DirectionsMap = ({ destination }) => {
9   const [currentLocation, setCurrentLocation] = useState(null);
10
11   const getUserLocation = () => {
12     return new Promise((resolve, reject) => {
13       if (navigator.geolocation) {
14         navigator.geolocation.getCurrentPosition(
15           (position) => {
16             const { latitude, longitude } = position.coords;
17             setCurrentLocation([latitude, longitude]); // Set state here
18             resolve([latitude, longitude]);
19           },
20           (error) => {
21             reject(error);
22           }
23         );
24       } else {
25         reject(new Error("Geolocation is not supported by this browser."));
26       }
27     });
28   };
29
30   useEffect(() => {
31     getUserLocation().catch((error) => {
32       console.error(error);
33     });
34   }, []);
35   console.log(currentLocation);
36
37   useEffect(() => {
38     if (!currentLocation) return; // Don't initialize map if location is not set
39
40     // Create the map instance
41     const map = L.map("map").setView(currentLocation, 13);
42
43     // Add tile layer
44     L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
45       attribution:
46         '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
47     }).addTo(map);
48
49     // Add routing control
50     L.Routing.control({
51       waypoints: [
52         L.latLng(currentLocation[0], currentLocation[1]),
53         L.latLng(destination[0], destination[1]),
54       ],
55       routeWhileDragging: true,
56     }).addTo(map);
57
58     return () => {
59       map.remove();
60     };
61   }, [currentLocation, destination]);
62
63   if (!currentLocation) {
64     return (
65       <ThreeCircles
66         visible={true}
67         height="100"
68         width="100"
69         color="#4fa94d"
70         ariaLabel="three-circles-loading"
71         wrapperStyle={{}}
72         className="mx-auto w-full"
73       />
74     );
75   }
76
77   return (
78     <div
79       id="map"
80       style={{
81         height: "80vh",
82         width: "100%",
83         color: "red",
84         border: 0,
85       }}
86     ></div>
87   );
88 };
89
90 export default DirectionsMap;
91

```



```
1 import React, { useState } from "react";
2 import { IoIosArrowDown, IoIosArrowUp } from "react-icons/io";
3 import AdditionalInfoTable from "../../../../../components/ui/AdditionalInfoTable";
4 import { formatDate } from "../../../../../utils/dateFormatter";
5
6 const DonationDetails = ({ row }) => {
7   const [open, setOpen] = useState(false);
8
9   const additionalInfo = [
10     { label: "Name", value: row.donor_name },
11     { label: "Donation Type", value: row.donation_type },
12     { label: "Quantity", value: row.quantity },
13     { label: "Date Received", value: formatDate(row.date_received) },
14     { label: "Notes", value: row.notes },
15   ];
16
17   return (
18     <>
19       <tr className="border-b border-opacity-25 border-primary">
20         <td className="pb-1">
21           <button
22             onClick={() => setOpen(!open)}
23             className="p-1 text-xs text-white rounded"
24           >
25             {open ? <IoIosArrowUp /> : <IoIosArrowDown />}
26           </button>
27         </td>
28         <td className="pb-1">{row.donor_name}</td>
29         <td className="pb-1">{row.donation_type}</td>
30         <td className="text-center">{row.quantity}</td>
31         <td className="text-center">{formatDate(row.date_received)}</td>
32       </tr>
33
34       {open && (
35         <AdditionalInfoTable
36           additionalInfo={additionalInfo}
37           row={row}
38         ></AdditionalInfoTable>
39       )}
40     </>
41   );
42 };
43 export default DonationDetails;
44
```

*DonationDetails*

```

1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import { debounce } from "lodash";
4 import Pagination from "../../components/ui/Pagination";
5 import Table from "../../components/ui/table";
6 import DonationDetails from "../DonationDetails";
7
8 const Donations = () => {
9   const [donations, setDonations] = useState([]);
10  const [totalDonations, setTotalDonations] = useState(0);
11  const [pageNumber, setPageNumber] = useState(0);
12  const [rowsPerPage, setRowsPerPage] = useState(10);
13  const [searchTerm, setSearchTerm] = useState("");
14  const [searchField, setSearchField] = useState("name");
15  const [sortField, setSortField] = useState("name");
16  const [sortOrder, setSortOrder] = useState("asc");
17
18  const headers = [
19    { label: "Donor Name", field: "donor_name", sortable: true },
20    { label: "Donation Type", field: "donation_type", sortable: true },
21    { label: "Quantity", field: "quantity", sortable: true },
22    { label: "Date Received", field: "date_received", sortable: true },
23  ];
24
25  const handleChangePage = (newPage) => {
26    setPageNumber(newPage);
27  };
28
29  const handleChangeRowsPerPage = (event) => {
30    setRowsPerPage(+event.target.value);
31    setPageNumber(0);
32  };
33
34  const handleSearchInput = debounce((event) => {
35    setSearchTerm(event.target.value);
36    setPageNumber(0);
37  }, 300);
38
39  const handleSort = (field) => {
40    const newSortOrder =
41      sortField === field % sortOrder === "asc" ? "desc" : "asc";
42    setSortField(field);
43    setSortOrder(newSortOrder);
44  };
45
46  useEffect(() => {
47    const fetchDonations = async () => {
48      try {
49        const offset = pageNumber * rowsPerPage;
50        const response = await axios.get(
51          "http://localhost:3000/api/donations",
52          {
53            params: {
54              search: searchTerm || "",
55              searchField,
56              sortField,
57              sortOrder,
58              limit: rowsPerPage,
59              offset,
60            },
61          }
62        );
63        setDonations(response.data.donations);
64        setTotalDonations(response.data.totalCount);
65      } catch (error) {
66        console.error("Error fetching donations:", error);
67      }
68    };
69
70    fetchDonations();
71  }, [searchTerm, searchField, sortField, sortOrder, pageNumber, rowsPerPage]);
72
73  const renderVictimRow = (victim) => (
74    <DonationDetails key={victim.id} row={victim} />
75  );
76
77  return (
78    <div className="w-full p-4">
79      <div className="flex flex-col justify-between mb-4">
80        <p className="text-center my-4 text-lg">Search by Name and Item</p>
81        <input
82          type="text"
83          onChange={handleSearchInput}
84          placeholder="Enter query"
85          className="px-4 py-2 border w-1/2 mx-auto bg-secondary rounded-lg"
86        />
87      </div>
88
89      <Table
90        headers={headers}
91        data={donations}
92        sortField={sortField}
93        sortOrder={sortOrder}
94        handleSort={handleSort}
95        renderRow={renderVictimRow}
96      />
97
98      <Pagination
99        pageNumber={pageNumber}
100        rowsPerPage={rowsPerPage}
101        totalItems={totalDonations}
102        onPageChange={handleChangePage}
103        onRowsPerPageChange={handleChangeRowsPerPage}
104      />
105    </div>
106  );
107 };
108 export default Donations;
109

```



```
1 import React from "react";
2 import { MapContainer, TileLayer, Marker, Popup } from "react-leaflet";
3 import "leaflet/dist/leaflet.css";
4 import L from "leaflet";
5 import markerIconUrl from "leaflet/dist/images/marker-icon.png";
6 import markerIconRetinaUrl from "leaflet/dist/images/marker-icon-2x.png";
7 import markerShadowUrl from "leaflet/dist/images/marker-shadow.png";
8
9 // Default marker icon
10 delete L.Icon.Default.prototype._getIconUrl;
11 L.Icon.Default.mergeOptions({
12   iconRetinaUrl: markerIconRetinaUrl,
13   iconUrl: markerIconUrl,
14   shadowUrl: markerShadowUrl,
15 });
16
17 const MapComponent = ({ latitude, longitude }) => {
18   return (
19     <MapContainer
20       center={[latitude, longitude]}
21       zoom={20}
22       style={{ height: "80vh", width: "100%" }}
23     >
24       <TileLayer
25         url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
26         attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
27       />
28       <Marker position={[latitude, longitude]}>
29         <Popup>
30           Location: {latitude}, {longitude}
31         </Popup>
32       </Marker>
33     </MapContainer>
34   );
35 };
36
37 export default MapComponent;
38
```

*ExactLocationMap*

```

1  import React, { createContext, useEffect, useState } from "react";
2  import {
3    getAuth,
4    createUserWithEmailAndPassword,
5    onAuthStateChanged,
6    signOut,
7    signInWithEmailAndPassword,
8  } from "firebase/auth";
9
10 import app from "../firebase/firebase.config";
11 export const AuthContext = createContext();
12 const auth = getAuth(app);
13
14 export const AuthProvider = ({ children }) => {
15   const [user, setUser] = useState(null);
16   const [loading, setLoading] = useState(null);
17
18   ;
19
20   const createUser = async (email, password) => {
21     setLoading(true);
22     try {
23       const result = await createUserWithEmailAndPassword(
24         auth,
25         email,
26         password
27       );
28       setUser(result.user);
29       console.log(result);
30       console.log(result.user);
31       return result;
32     } finally {
33       setLoading(false);
34     }
35   };
36
37   const login = async (email, password) => {
38     setLoading(true);
39     try {
40       const result = await signInWithEmailAndPassword(auth, email, password);
41       setUser(result.user);
42       console.log(result);
43       console.log(result.user);
44       return result;
45     } finally {
46       setLoading(false);
47     }
48   };
49
50   const logout = async () => {
51     setLoading(true);
52     try {
53       await signOut(auth);
54     } finally {
55       setLoading(false);
56     }
57   };
58
59   useEffect(() => {
60     const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
61       setUser(currentUser);
62       setLoading(false);
63     });
64
65     return () => {
66       unsubscribe();
67     };
68   }, []);
69
70   const authInfo = {
71     user,
72     loading,
73     createUser,
74     login,
75     logout,
76   };
77
78   return (
79     <AuthContext.Provider value={authInfo}>{children}</AuthContext.Provider>
80   );
81 };
82
83 export default AuthProvider;
84

```



```

1  "use client";
2  import React, { useContext, useState } from "react";
3  import { Label } from "../../components/ui/label";
4  import { Input } from "../../components/ui/input";
5  import { Button } from "../../components/ui/button";
6  import { LabelInputContainer } from "../../components/ui/LabelInputContainer";
7  import { BottomGradient } from "../../components/ui/BottomGradient";
8  import { cn } from "@/lib/utils";
9  import { InputFields } from "../../inputFields";
10 import { AuthContext } from "../../context/AuthProvider";
11 import logger from "../../utils/logger";
12 import { toast } from "sonner";
13 import {
14   Select,
15   SelectContent,
16   SelectGroup,
17   SelectItem,
18   SelectLabel,
19   SelectTrigger,
20   SelectValue,
21 } from "../../components/ui/select";
22 import { useNavigate } from "react-router-dom";
23
24 const Login = () => {
25   const { login } = useContext(AuthContext);
26   const [role, setRole] = useState("");
27
28   const navigate = useNavigate();
29
30   const handleSelectChange = (value) => {
31     setRole("");
32     setRole(value);
33   };
34
35   const handleLogin = (event) => {
36     event.preventDefault();
37     const form = event.target;
38     const email = form.email.value;
39     const password = form.password.value;
40
41     if (!role) {
42       toast.error("Please select a role.");
43       return;
44     }
45
46     login(email, password)
47       .then((result) => {
48         const user = result.user;
49         console.log(user);
50         toast.success("User logged in successfully");
51         if (user && role === "rescuer") {
52           navigate("/rescue-portal");
53         } else if (user && role === "victim") {
54           navigate("/victim-portal");
55         } else {
56           navigate("/login");
57         }
58       })
59       .catch((err) => {
60         console.error(err);
61         toast.error(err);
62         logger("Error during login: ${err.message}", "error", {
63           color: "ff0000",
64           backgroundColor: "ff8d7d",
65         });
66       });
67   };
68
69   return (
70     <div className="max-w-md w-full mx-auto rounded-none md:rounded-2xl my-12 p-4 md:p-8 shadow-input bg-primary bg-opacity-15">
71       <h2 className="font-light text-3xl text-center font-museo my-6 pb-6 text-primary">
72         Welcome to flood guard
73       </h2>
74       <p className="text-center text-primary my-6 pb-6 text-xl font-light">
75         Log in to your account{" "}
76       </p>
77       <form className="my-8" onSubmit={handleLogin}>
78         <div className="flex flex-col space-y-4">
79           <InputFields
80             .filter((id) => id === "email" || id === "password")
81             .map((id, label, type, placeholder) => {
82               <LabelInputContainer key={id}>
83                 <Label htmlFor={id}>{label}</Label>
84                 <input id={id} placeholder={placeholder} type={type} />
85               </LabelInputContainer>
86             })
87         </div>
88         <LabelInputContainer className="mb-4">
89           <Label htmlFor="role">Select Role</Label>
90           <Select onValueChange={handleSelectChange}>
91             <SelectTrigger className="w-full">
92               <SelectValue
93                 placeholder="Select a role"
94                 className="placeholder:text-primary"
95               />
96             </SelectTrigger>
97             <SelectContent>
98               <SelectGroup>
99                 <SelectLabel>--Select Role--</SelectLabel>
100                <SelectItem value="victim">Victim</SelectItem>
101                <SelectItem value="rescuer">Rescuer</SelectItem>
102              </SelectGroup>
103            </SelectContent>
104          </Select>
105        </LabelInputContainer>
106        <Button
107          variant="submit"
108          className="w-full text-lg font-light text-primary"
109          type="submit"
110        >
111          Log in
112        </ButtonGradient />
113      </Button>
114
115      <div className="bg-gradient-to-r from-transparent via-primary dark:via-primary to-transparent my-10 h-[1px] w-full" />
116
117      <div className="flex flex-col space-y-4">
118        <div>
119          <p className="text-center text-primary font-light">
120            Doesn't have an account?{" "}
121          <a
122            href="/signup"
123            className={cn(
124              "text-primary font-light underline",
125              "text-primary font-light underline"
126            )}
127          >
128            Sign up
129          </a>
130        </p>
131        </div>
132      </div>
133    </form>
134  </div>
135  );
136 }
137
138 export default Login;
139

```



```
1 import { Card, CardContent, CardHeader, CardTitle } from "@components/ui/card";
2 import { Tabs, TabsContent, TabsList, TabsTrigger } from "@components/ui/tabs";
3 import Victims from "../VictimList/Victims";
4 import Shelters from "../ShelterList/Shelters";
5 import Donations from "../DonorList/Donors";
6 import { getGridColsClass } from "../../utils/gridColsCalculator";
7
8 const tabsData = [
9   {
10     label: "Victim List",
11     value: "victimList",
12     title: "List of Victims",
13     content: <Victims />,
14   },
15   {
16     label: "Shelter List",
17     value: "shelterList",
18     title: "List of Shelter",
19     content: <Shelters />,
20   },
21   {
22     label: "Donation List",
23     value: "donationList",
24     title: "List of Donations",
25     content: <Donations />,
26   },
27 ];
28
29 const RescueHome = () => {
30   const gridColsClass = getGridColsClass(tabsData.length);
31   return (
32     <Tabs defaultValue="victimList" className="container my-10">
33       <TabsList className={`grid w-full mx-auto my-4 ${gridColsClass}`}>
34         {tabsData.map((tab) => (
35           <TabsTrigger key={tab.value} className="text-lg" value={tab.value}>
36             {tab.label}
37           </TabsTrigger>
38         ))}
39       </TabsList>
40       {tabsData.map((tab) => (
41         <TabsContent key={tab.value} value={tab.value}>
42           <Card>
43             <CardHeader className="mx-auto w-full">
44               <CardTitle className="font-museo font-light text-3xl">
45                 {tab.title}
46               </CardTitle>
47             </CardHeader>
48             <CardContent>{tab.content}</CardContent>
49           </Card>
50         </TabsContent>
51       ))}
52     </Tabs>
53   );
54 };
55
56 export default RescueHome;
57
```

```

1 import React, { useState } from "react";
2 import { IoIosArrowDown, IoIosArrowUp } from "react-icons/io";
3 import AdditionalInfoTable from "../../../../../components/ui/AdditionalInfoTable";
4
5 // ShelterDetails Component to display each row with collapsible data
6 const ShelterDetails = ({ row }) => {
7   const [open, setOpen] = useState(false);
8
9   const additionalInfo = [
10     { label: "Address Upazila", value: row.address_upazila },
11     { label: "Address District", value: row.address_district },
12     {
13       label: "Exact Location",
14       value: ["View Location", "Get Directions"], // For handling the map actions
15     },
16     { label: "Required Food", value: row.required_food },
17     { label: "Required Medicine", value: row.required_medicine },
18     { label: "Medical Support", value: row.medical_support },
19   ];
20
21   return (
22     <>
23       <tr className="border-b border-opacity-25 border-primary">
24         <td className="pb-1">
25           <button
26             onClick={() => setOpen(!open)}
27             className="p-1 text-xs text-white rounded"
28           >
29             {open ? <IoIosArrowUp /> : <IoIosArrowDown />}
30           </button>
31         </td>
32         <td className="pb-1">{row.name}</td>
33         <td className="pb-1">{row.address_area}</td>
34         <td className="text-center">{row.number_of_children}</td>
35         <td className="text-center">{row.number_of_women}</td>
36         <td className="text-center">{row.number_of_mothers}</td>
37       </tr>
38
39       {open && (
40         <AdditionalInfoTable
41           additionalInfo={additionalInfo}
42           row={row}
43         ></AdditionalInfoTable>
44       )}
45     </>
46   );
47 };
48 export default ShelterDetails;
49

```

```

1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import { debounce } from "lodash";
4 import Pagination from "../../components/ui/Pagination";
5 import Table from "../../components/ui/Table";
6 import ShelterDetails from "../ShelterDetails";
7
8 const Shelters = () => {
9   const [shelters, setShelters] = useState([]);
10   const [totalShelters, setTotalShelters] = useState(0);
11   const [pageNumber, setPageNumber] = useState(0);
12   const [rowsPerPage, setRowsPerPage] = useState(10);
13   const [searchTerm, setSearchTerm] = useState("");
14   const [searchField, setSearchField] = useState("name");
15   const [sortField, setSortField] = useState("name");
16   const [sortOrder, setSortOrder] = useState("asc");
17
18   const headers = [
19     { label: "Name", field: "name", sortable: true },
20     { label: "Area", field: "address_area", sortable: true },
21     {
22       label: "Number of Children",
23       field: "number_of_children",
24       sortable: true,
25     },
26     { label: "Number of Women", field: "number_of_women", sortable: true },
27     { label: "Number of Mothers", field: "number_of_mothers", sortable: true },
28   ];
29
30   const handleChangePage = (newPage) => {
31     setPageNumber(newPage);
32   };
33
34   const handleChangeRowsPerPage = (event) => {
35     setRowsPerPage(event.target.value);
36     setPageNumber(0);
37   };
38
39   useEffect(() => {
40     const fetchShelters = async () => {
41       try {
42         const offset = pageNumber * rowsPerPage;
43
44         const response = await axios.get("http://localhost:3000/api/shelters", {
45           params: {
46             search: searchTerm || "",
47             searchField,
48             sortField,
49             sortOrder,
50             limit: rowsPerPage,
51             offset,
52           },
53         });
54         setShelters(response.data.shelters);
55         setTotalShelters(response.data.totalCount);
56       } catch (error) {
57         console.error("Error fetching shelters:", error);
58       }
59     };
60
61     fetchShelters();
62   }, [searchTerm, searchField, sortField, sortOrder, pageNumber, rowsPerPage]);
63
64   const handleSearchInput = debounce((event) => {
65     setSearchTerm(event.target.value);
66     setPageNumber(0);
67   }, 300);
68
69   const handleSort = (field) => {
70     const newSortOrder =
71       sortField === field && sortOrder === "asc" ? "desc" : "asc";
72     setSortField(field);
73     setSortOrder(newSortOrder);
74   };
75
76   const renderVictimRow = (victim) => {
77     <ShelterDetails key={victim.id} row={victim} />
78   };
79
80   return (
81     <div className="w-full p-4">
82       <div className="flex flex-col justify-between mb-4">
83         <p className="text-center my-4 text-lg">Search by Name and Area</p>
84         <input
85           type="text"
86           onChange={handleSearchInput}
87           placeholder="Enter query"
88           className="px-4 py-2 border w-1/2 mx-auto bg-secondary rounded-lg"
89         />
90       </div>
91       <Table
92         headers={headers}
93         data={shelters}
94         sortField={sortField}
95         sortOrder={sortOrder}
96         handleSort={handleSort}
97         renderRow={renderVictimRow}
98       />
99       <Pagination
100         pageNumber={pageNumber}
101         rowsPerPage={rowsPerPage}
102         totalItems={totalShelters}
103         onPageChange={handleChangePage}
104         onRowsPerPageChange={handleChangeRowsPerPage}
105       />
106     </div>
107   );
108 };
109
110 export default Shelters;
111

```

## Signup

```

1 import { IoIosArrowDown, IoIosArrowUp } from "react-icons/io";
2 import AdditionalInfoTable from "../../components/ui/AdditionalInfoTable";
3 import Badge from "../../components/ui/badge";
4 import { useState } from "react";
5 import { formatDate } from "../../utils/dateFormatter";
6
7 const VictimDetails = ({ row }) => {
8   const [open, setOpen] = useState(false);
9
10   const additionalInfo = [
11     { label: "NID", value: row.NID || "N/A" },
12     { label: "Address Upazila", value: row.address_upazila || "N/A" },
13     { label: "Address District", value: row.address_district || "N/A" },
14     {
15       label: "Exact Location",
16       value: ["View Location", "Get Directions"],
17     },
18     { label: "Mobile", value: row.mobile || "N/A" },
19     { label: "Email", value: row.email || "N/A" },
20     { label: "Gender", value: row.gender || "N/A" },
21     { label: "Age", value: row.age || "N/A" },
22     { label: "Number of Family Members", value: row.number_of_family_members || "N/A" },
23     { label: "Health Status", value: row.health_status || "N/A" },
24     { label: "Rescue Time", value: formatDate(row.rescue_time) || "N/A" },
25     { label: "Resources Needed", value: row.resources_needed || "N/A" },
26   ];
27
28   return (
29     <div>
30       { " "}
31       <tr className="border-b border-opacity-25 border-primary">
32         { " "}
33         <td className="pb-1">
34           { " "}
35           <button
36             onClick={() => setOpen(!open)}
37             className="p-1 text-xs text-white rounded"
38           >
39             {open ? <IoIosArrowUp /> : <IoIosArrowDown />}{ " "}
40           </button>{ " "}
41         </td>
42         <td className="pb-1">{row.name}</td>{ " "}
43         <td className="pb-1">{row.address_area}</td>{ " "}
44         <td className="text-center">
45           { " "}
46           <Badge
47             variant={
48               row.danger_level === "High"
49                 ? "destructive"
50                 : row.danger_level === "Medium"
51                   ? "warning"
52                   : row.danger_level === "Low"
53                     ? "success"
54                     : "default"
55             }
56           >
57             {row.danger_level || "N/A"}{ " "}
58           </Badge>{ " "}
59         </td>{ " "}
60         <td className="text-center">
61           { " "}
62           <Badge
63             variant={
64               row.rescue_status === "Pending"
65                 ? "destructive"
66                 : row.rescue_status === "In Progress"
67                   ? "warning"
68                   : row.rescue_status === "Completed"
69                     ? "success"
70                     : "default"
71             }
72           >
73             {row.rescue_status || "N/A"}{ " "}
74           </Badge>{ " "}
75         </td>{ " "}
76         <td className="text-center">
77           { " "}
78           <Badge
79             variant={
80               row.health_status === "Critical"
81                 ? "destructive"
82                 : row.health_status === "Injured"
83                   ? "warning"
84                   : row.health_status
85                     ? "success"
86                     : "default"
87             }
88           >
89             {row.health_status || "N/A"}{ " "}
90           </Badge>{ " "}
91         </td>{ " "}
92       </tr>{ " "}
93       {open && (
94         <AdditionalInfoTable
95           additionalInfo={additionalInfo}
96           row={row}
97         ></AdditionalInfoTable>
98       )}{ " "}
99     </div>
100   );
101 };
102
103 export default VictimDetails;
104

```

```

1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import { debounce } from "lodash";
4 import Pagination from "../../components/ui/Pagination";
5 import Table from "../../components/ui/table";
6 import VictimDetails from "../VictimDetails";
7
8 const Victims = () => {
9   const [victims, setVictims] = useState([]);
10   const [totalVictims, setTotalVictims] = useState(0);
11   const [pageNumber, setPageNumber] = useState(0);
12   const [rowsPerPage, setRowsPerPage] = useState(10);
13   const [searchTerm, setSearchTerm] = useState("");
14   const [searchField, setSearchField] = useState("name");
15   const [sortField, setSortField] = useState("name");
16   const [sortOrder, setSortOrder] = useState("asc");
17
18   const headers = [
19     { label: "Name", field: "name" },
20     { label: "Area", field: "address_area" },
21     { label: "Danger Level", field: "danger_level" },
22     { label: "Rescue Status", field: "rescue_status" },
23     { label: "Health Status", field: "health_status" },
24   ];
25
26   useEffect(() => {
27     const fetchVictims = async () => {
28       try {
29         const offset = pageNumber * rowsPerPage;
30         const response = await axios.get("http://localhost:3000/api/victims", {
31           params: {
32             search: searchTerm || "",
33             searchField,
34             sortField,
35             sortOrder,
36             limit: rowsPerPage,
37             offset,
38           },
39         });
40         setVictims(response.data.victims);
41         setTotalVictims(response.data.totalCount);
42       } catch (error) {
43         console.error("Error fetching victims:", error);
44       }
45     };
46
47     fetchVictims();
48   }, [searchTerm, searchField, sortField, sortOrder, pageNumber, rowsPerPage]);
49
50   const handleSearchInput = debounce((event) => {
51     setSearchTerm(event.target.value);
52     setPageNumber(0);
53   }, 300);
54
55   const handleSort = (field) => {
56     const newSortOrder =
57       sortField === field ? (sortOrder === "asc" ? "desc" : "asc") : "asc";
58     setSortField(field);
59     setSortOrder(newSortOrder);
60   };
61
62   const handleChangePage = (newPage) => {
63     setPageNumber(newPage);
64   };
65
66   const handleChangeRowsPerPage = (event) => {
67     setRowsPerPage(+event.target.value);
68     setPageNumber(0);
69   };
70
71   const renderVictimRow = (victim) => {
72     <VictimDetails key={victim.id} row={victim} />
73   };
74
75   return (
76     <div className="w-full p-4">
77       <div className="flex flex-col justify-between mb-4">
78         <p className="text-center my-4 text-lg">
79           Search by Name, Danger Level, Rescue Status or Health Status
80         </p>
81         <input
82           type="text"
83           onChange={handleSearchInput}
84           placeholder="Enter query"
85           className="px-4 py-2 border w-1/2 mx-auto bg-secondary rounded-lg"
86         />
87       </div>
88
89       <Table
90         headers={headers}
91         data={victims}
92         sortField={sortField}
93         sortOrder={sortOrder}
94         handleSort={handleSort}
95         renderRow={renderVictimRow}
96       />
97
98       <Pagination
99         pageNumber={pageNumber}
100         rowsPerPage={rowsPerPage}
101         totalItems={totalVictims}
102         onPageChange={handleChangePage}
103         onRowsPerPageChange={handleChangeRowsPerPage}
104       />
105     </div>
106   );
107 };
108
109 export default Victims;
110

```