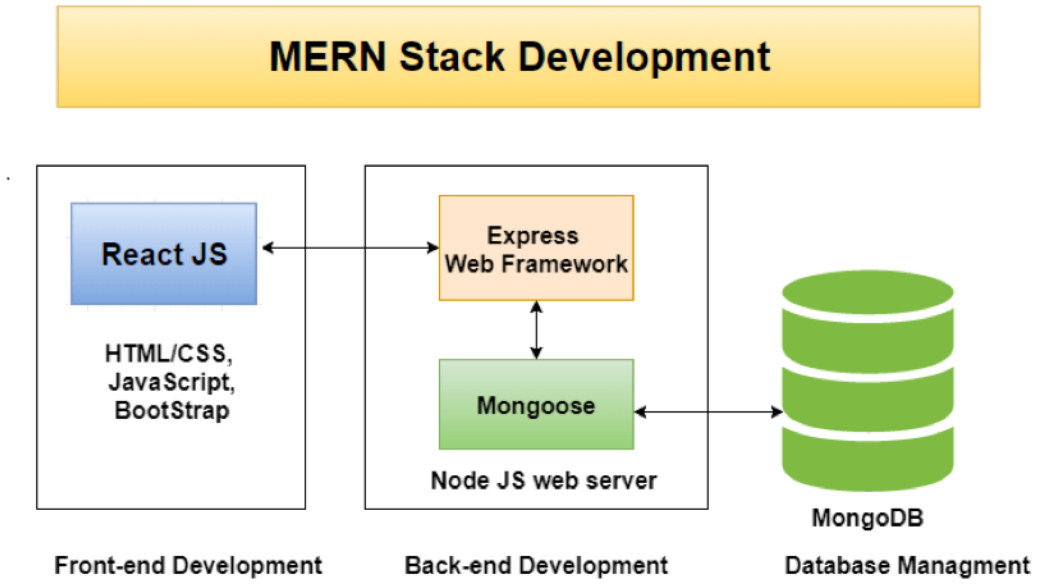
Software Engineering

Digital Assignment-2

Done By: S Vishwajith 23BCE1145

Project Title: Lost And Found Item Tracker

1. Architectural Prototype (screen shots):



```javascript
import express from "express";
import dotenv from "dotenv";
import { connectDB } from "./config/db.js";
import finderRoutes from "./routes/finder.route.js";
import claimerRoutes from "./routes/claimer.route.js";
import itemRoutes from "./routes/item.route.js";

dotenv.config();
const app = express();
app.use(express.json());
const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {
    connectDB();
    console.log(`Server is running on http://localhost:${PORT}`);
});

app.use("/api/finder", finderRoutes);
app.use("/api/claimer", claimerRoutes);
app.use("/api/item", itemRoutes);

app.get("/", (req, res) => {
    res.send("LAFIT API is running...");
});
```

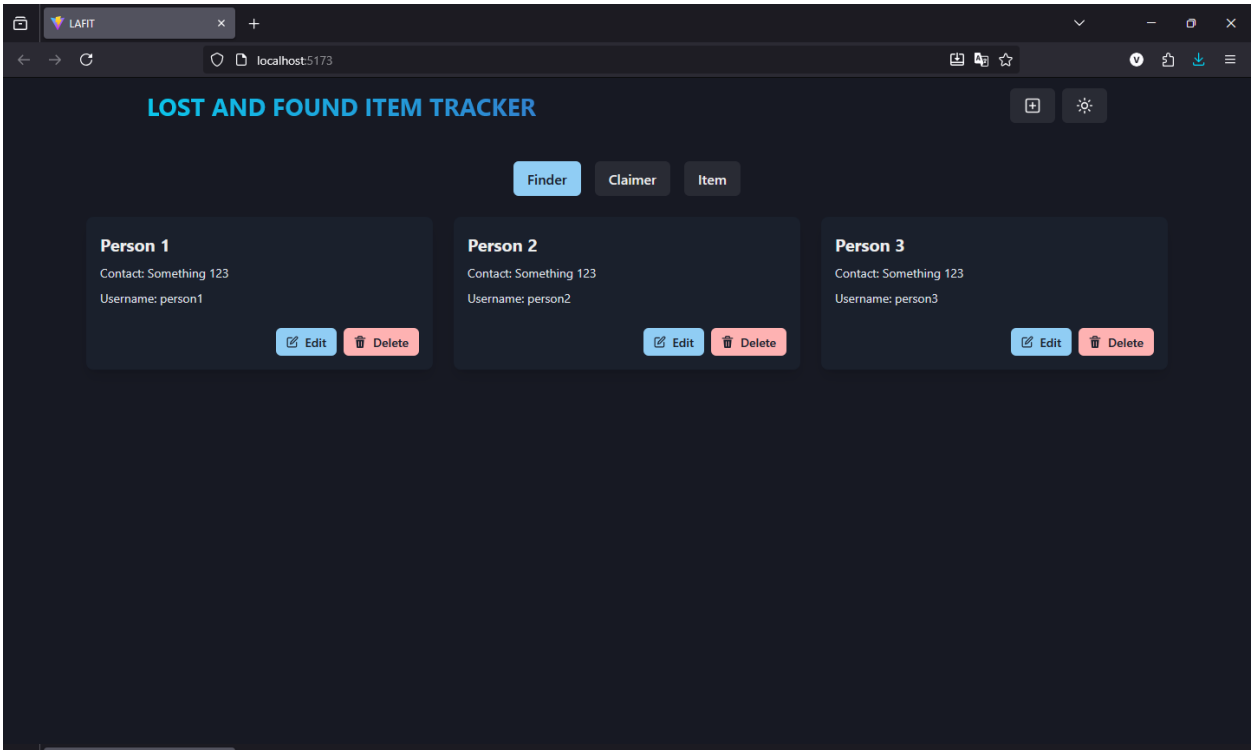
```javascript
import mongoose from "mongoose";

export const connectDB = async() => {
    try {
        const conn = await mongoose.connect(process.env.MONGO_URI);
        console.log(`MongoDB connected: ${conn.connection.host}`);
    } catch (error) {
        console.error(`Error: ${error.message}`);
        process.exit(1);
    }
}
```

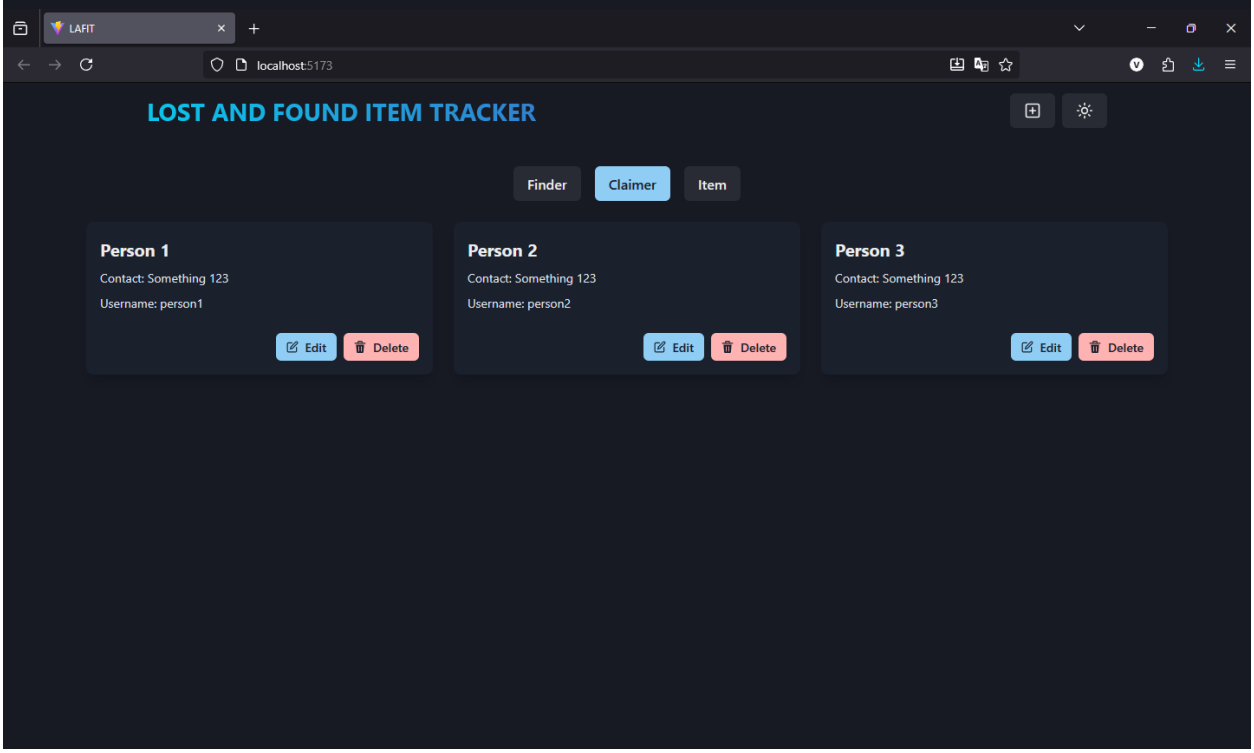
```
import express from "express";
import { createFinder, deleteFinder, getAllFinders, getFinderById, updateFinder } from "../controllers/finder.controller.js";

const router = express.Router();

router.get("/", getAllFinders);
router.post("/", createFinder);
router.get("/:id", getFinderById);
router.put("/:id", updateFinder);
router.delete("/:id", deleteFinder);

export default router;
```

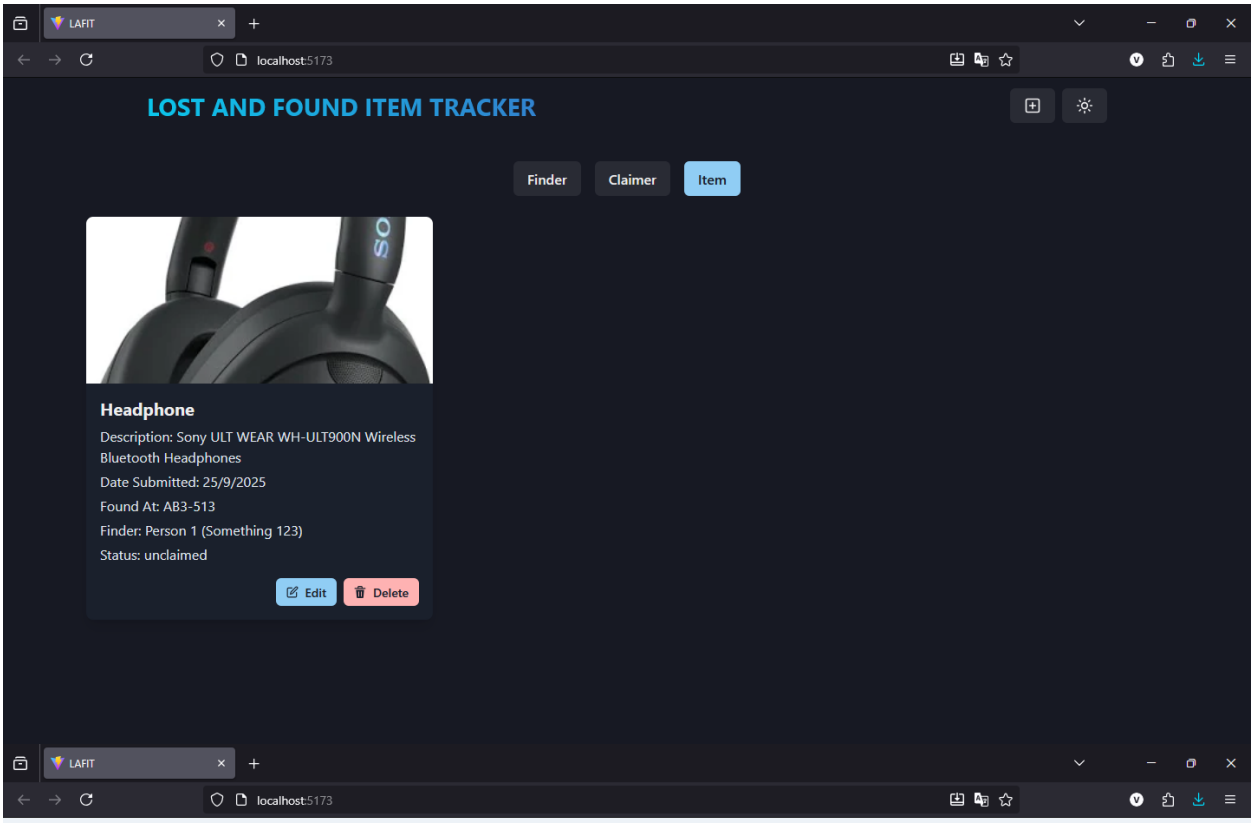
```
import express from 'express';
import { createClaimer, deleteClaimer, getAllClaimers, getClaimerById, updateClaimer } from '../controllers/claimer.controller.js'

const router = express.Router();

router.get("/", getAllClaimers);
router.post("/", createClaimer);
router.get("/:id", getClaimerById);
router.put("/:id", updateClaimer);
router.delete("/:id", deleteClaimer);

export default router;
```

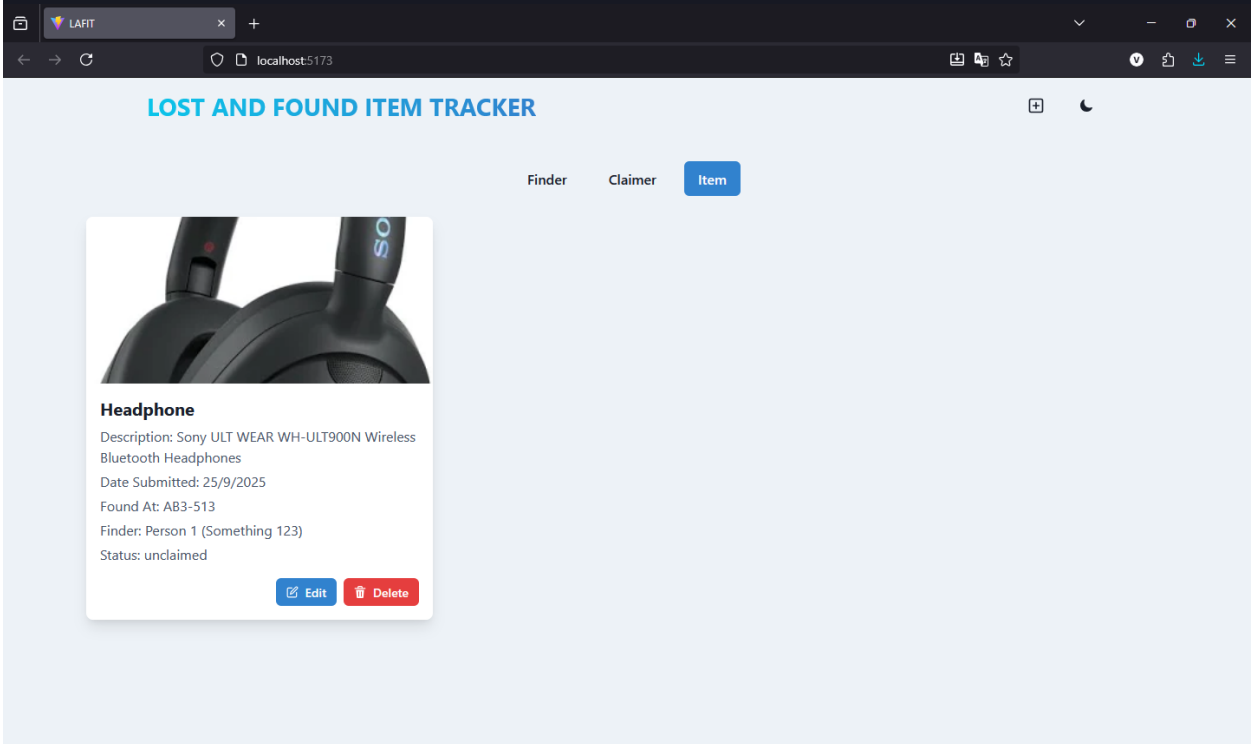
```
import express from 'express';
import { createItem, deleteItem, getAllItems, getItemById, updateItem } from '../controllers/item.controller.js';

const router = express.Router();

router.get("/", getAllItems);
router.post("/", createItem);
router.get("/:id", getItemById);
router.put("/:id", updateItem);
router.delete("/:id", deleteItem);

export default router;
```

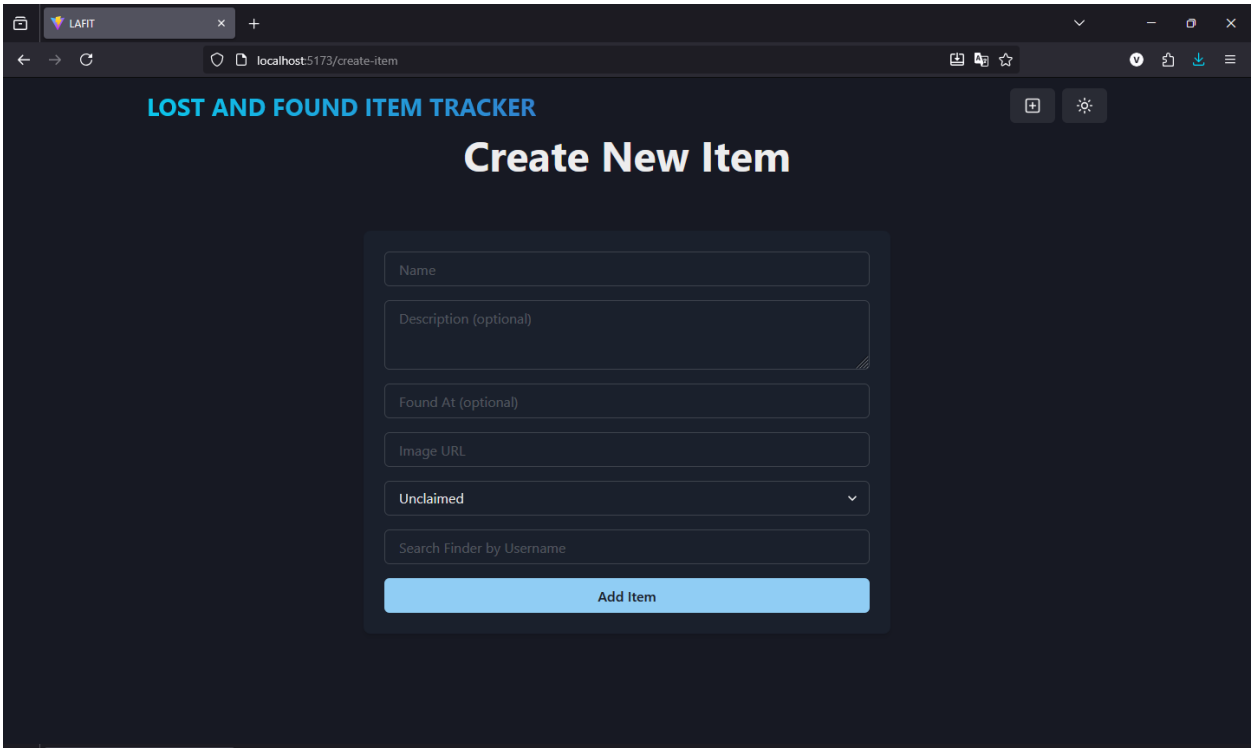
2. User interface prototype (screen shots):

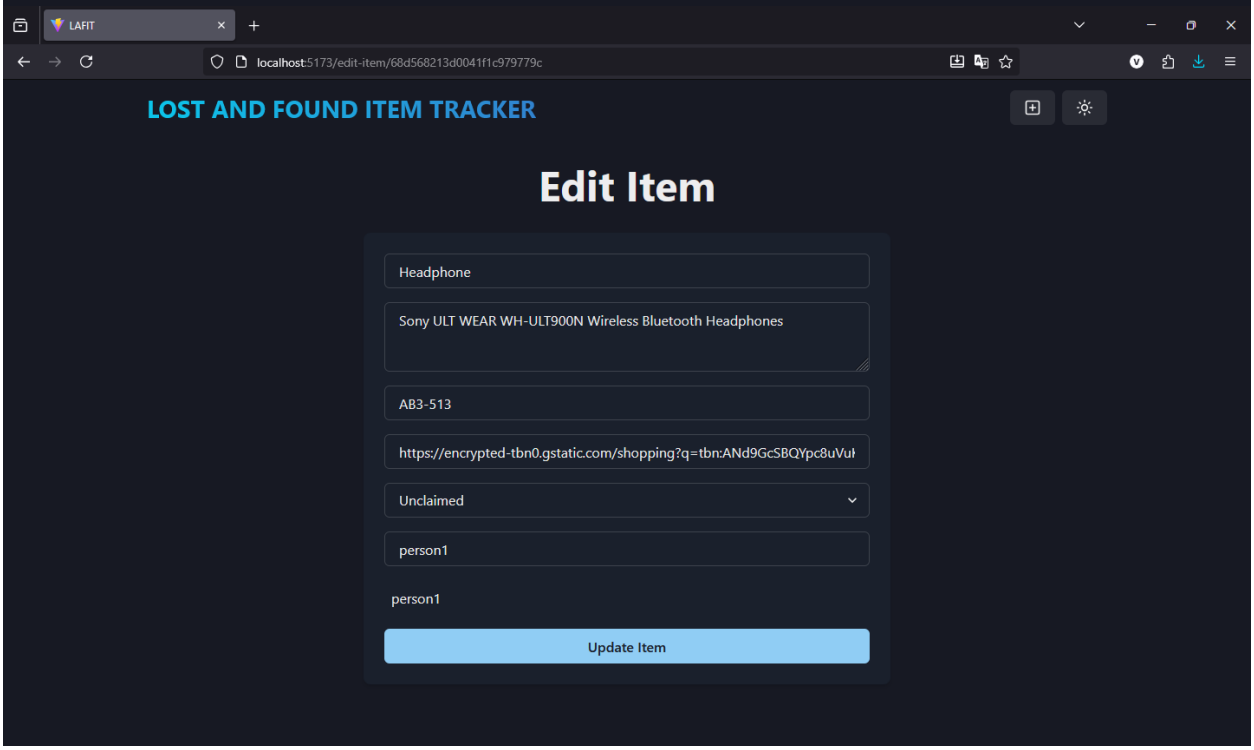





LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Headphone
Description: Sony ULT WEAR WH-ULT900N Wireless Bluetooth Headphones
Date Submitted: 25/9/2025
Found At: AB3-513
Finder: Person 1 (Something 123)
Status: unclaimed
Edit
Delete

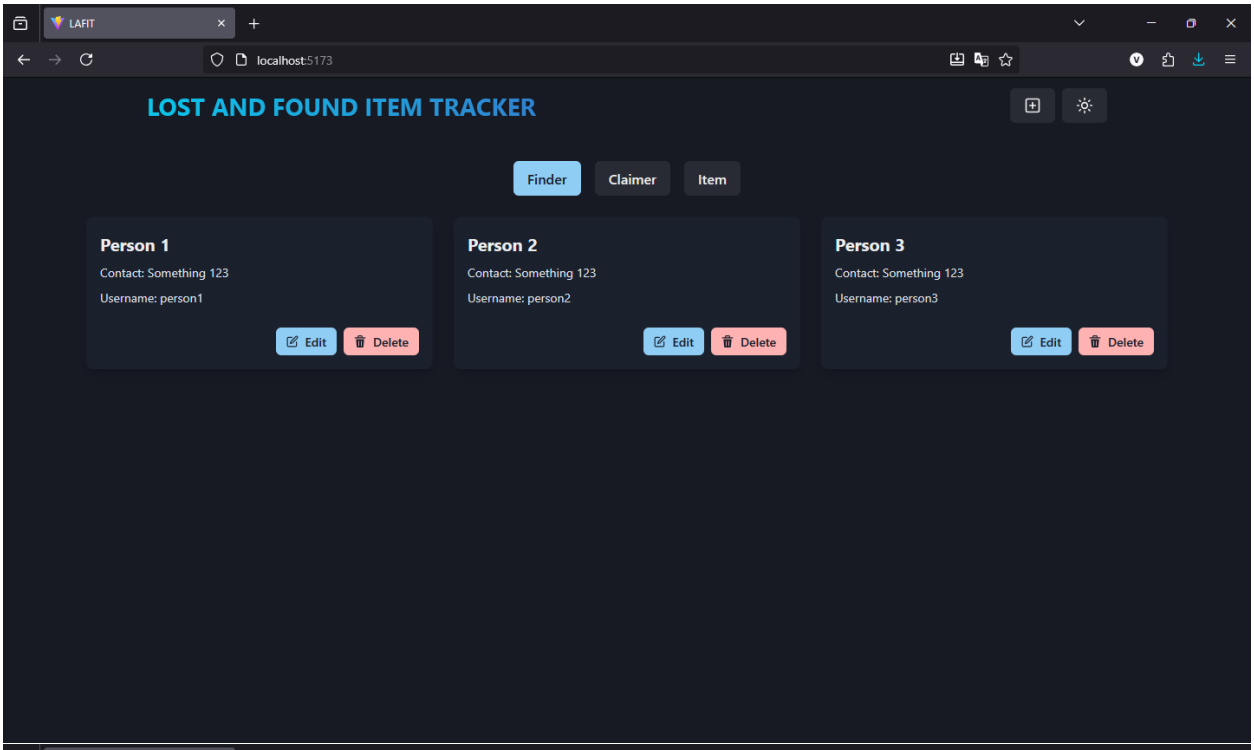

LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Headphone
Description: Sony ULT WEAR WH-ULT900N Wireless Bluetooth Headphones
Date Submitted: 25/9/2025
Found At: AB3-513
Finder: Person 1 (Something 123)
Status: unclaimed
Edit
Delete

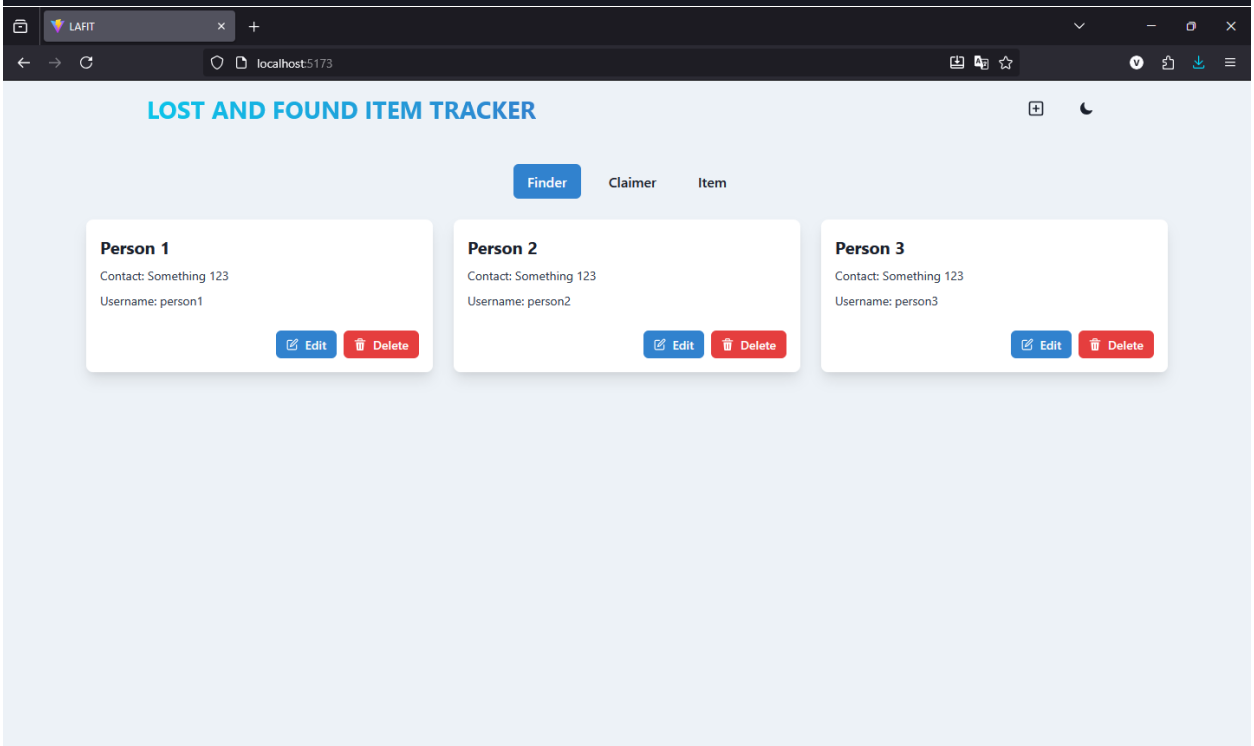

LAFIT
localhost:5173/create-item
LOST AND FOUND ITEM TRACKER
Create New Item
Name
Description (optional)
Found At (optional)
Image URL
Unclaimed
Search Finder by Username
Add Item


LAFIT
localhost:5173/edit-item/68d568213d0041f1c979779c
LOST AND FOUND ITEM TRACKER
Edit Item
Headphone
Sony ULT WEAR WH-ULT900N Wireless Bluetooth Headphones
AB3-513
https://encrypted-tbn0.gstatic.com/shopping?q=tbn:ANd9GcSBQYpc8uVuł
Unclaimed
person1
person1
Update Item

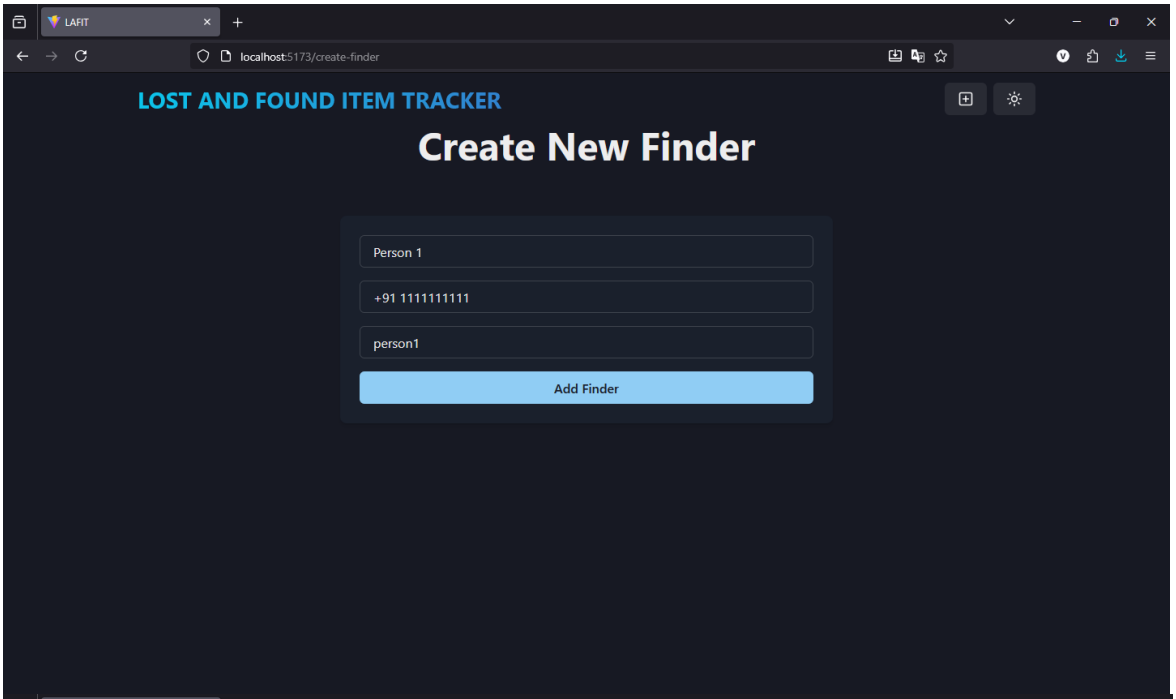
3. Screen shots of developed functionalities:

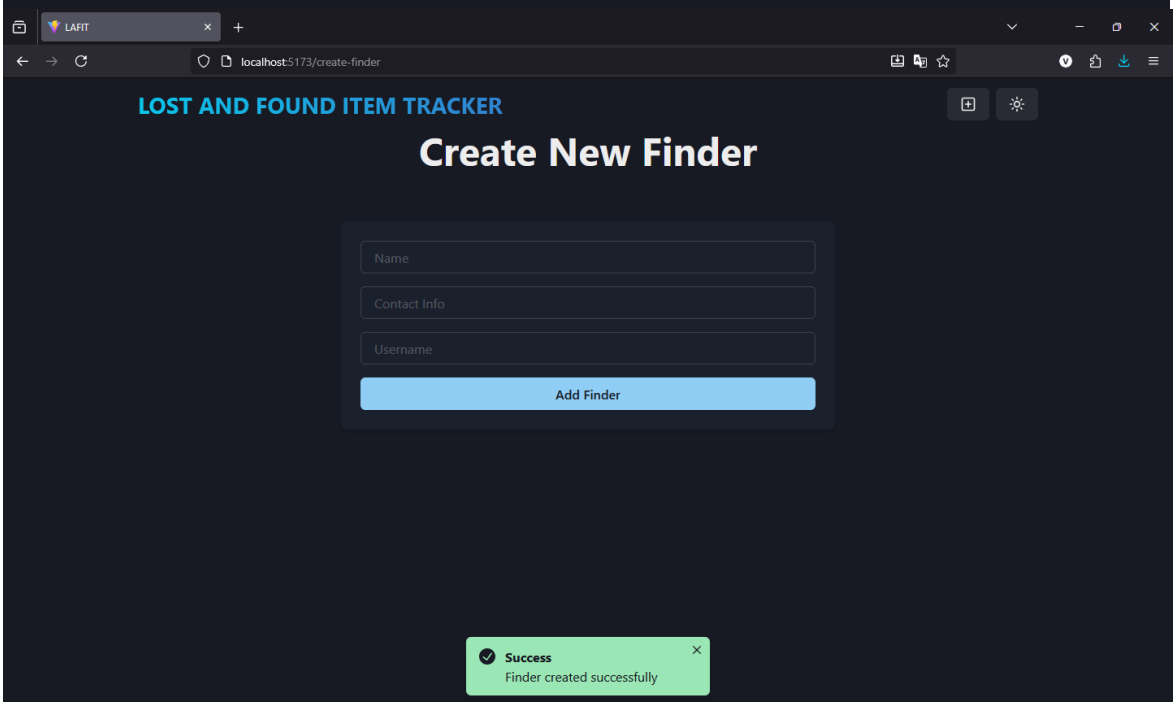
a. Light/Dark Mode toggle:

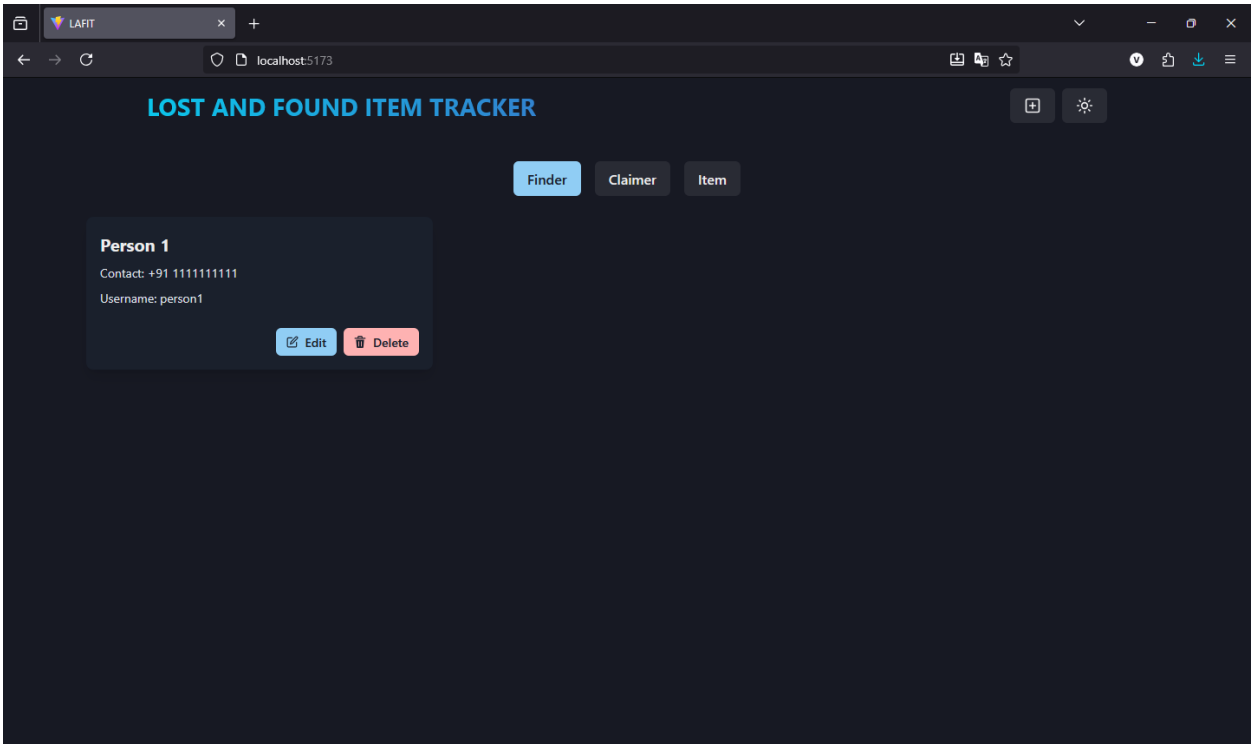


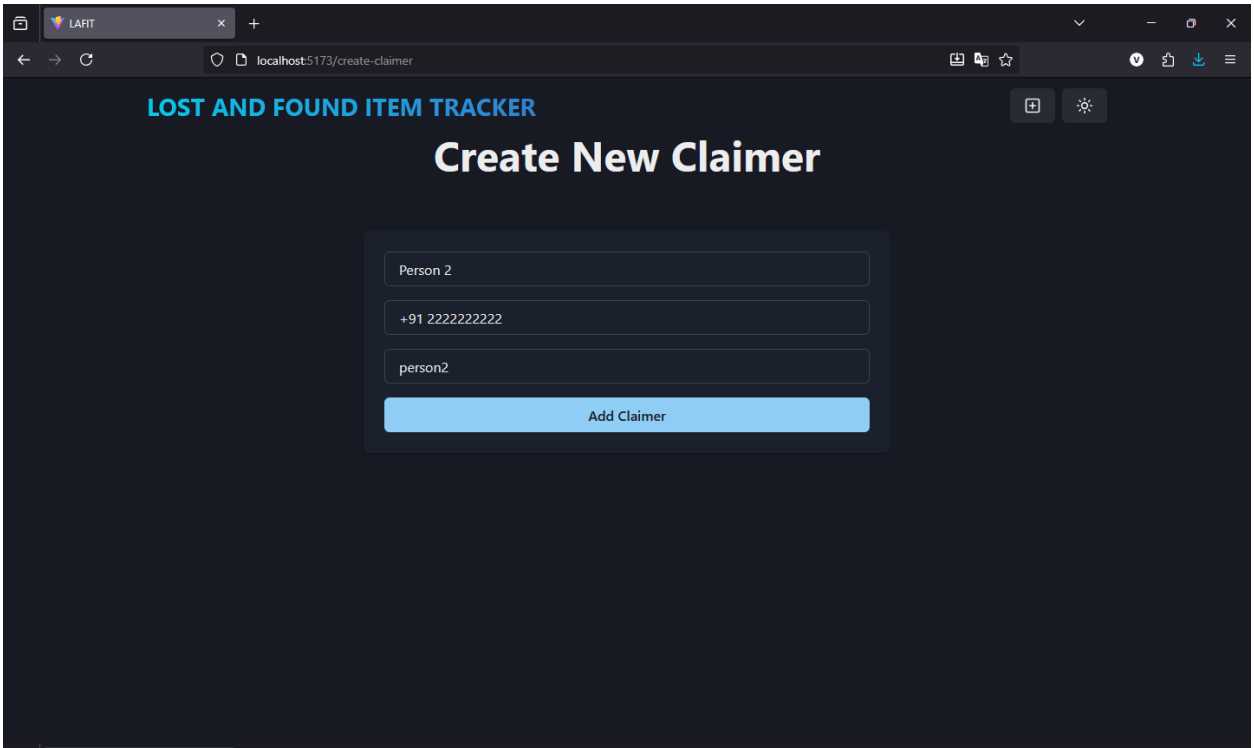


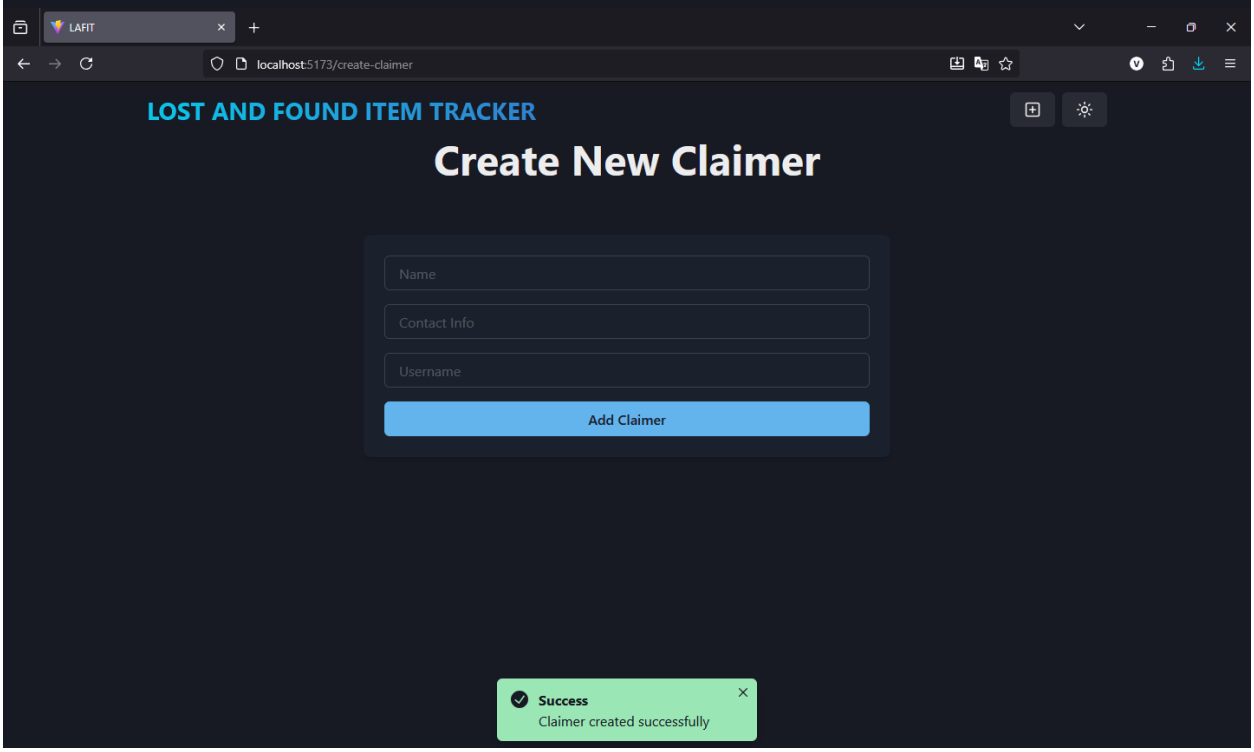
b. Creation of Finders, Claimers, and Items:

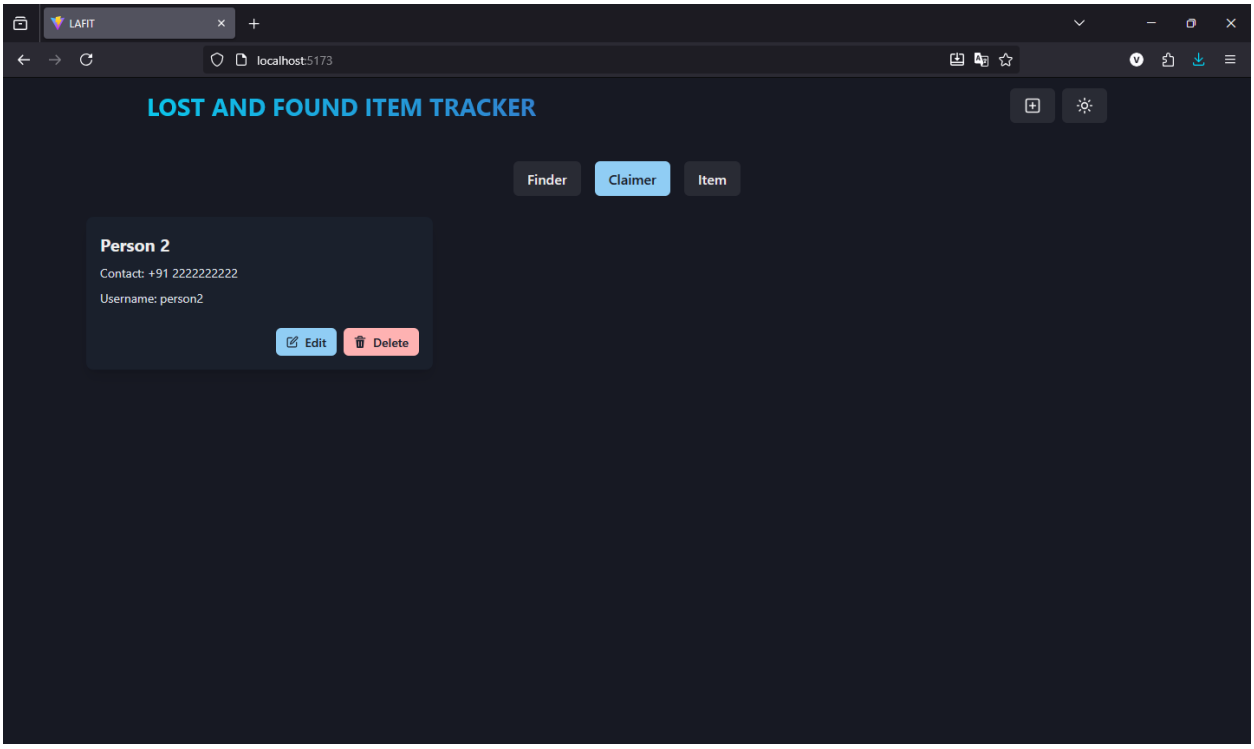


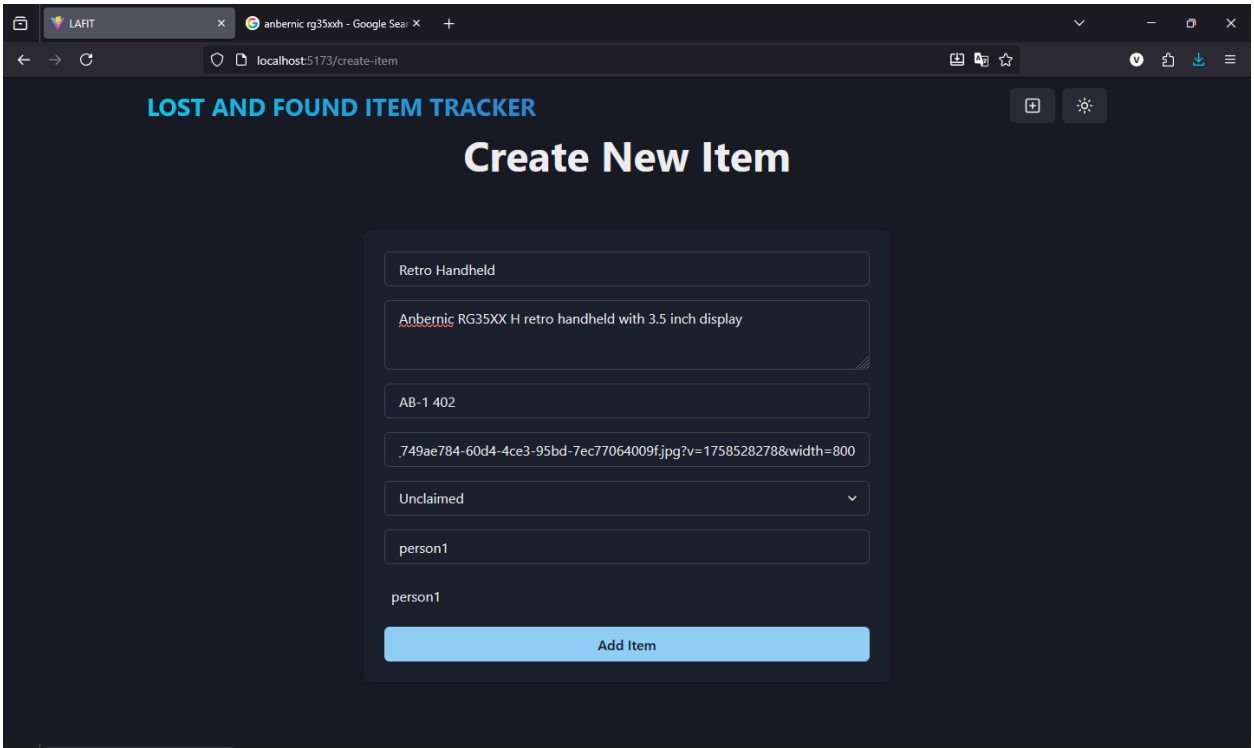


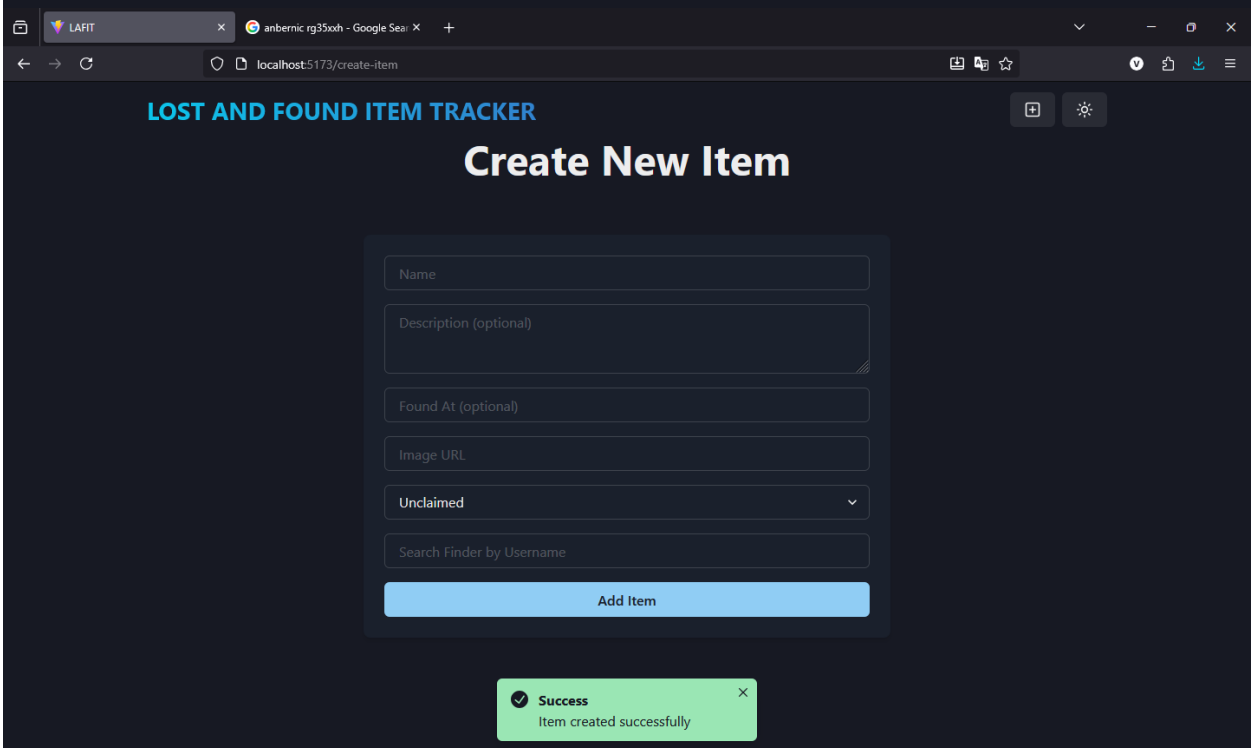

LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Person 1
Contact: +91 1111111111
Username: person1
Edit
Delete

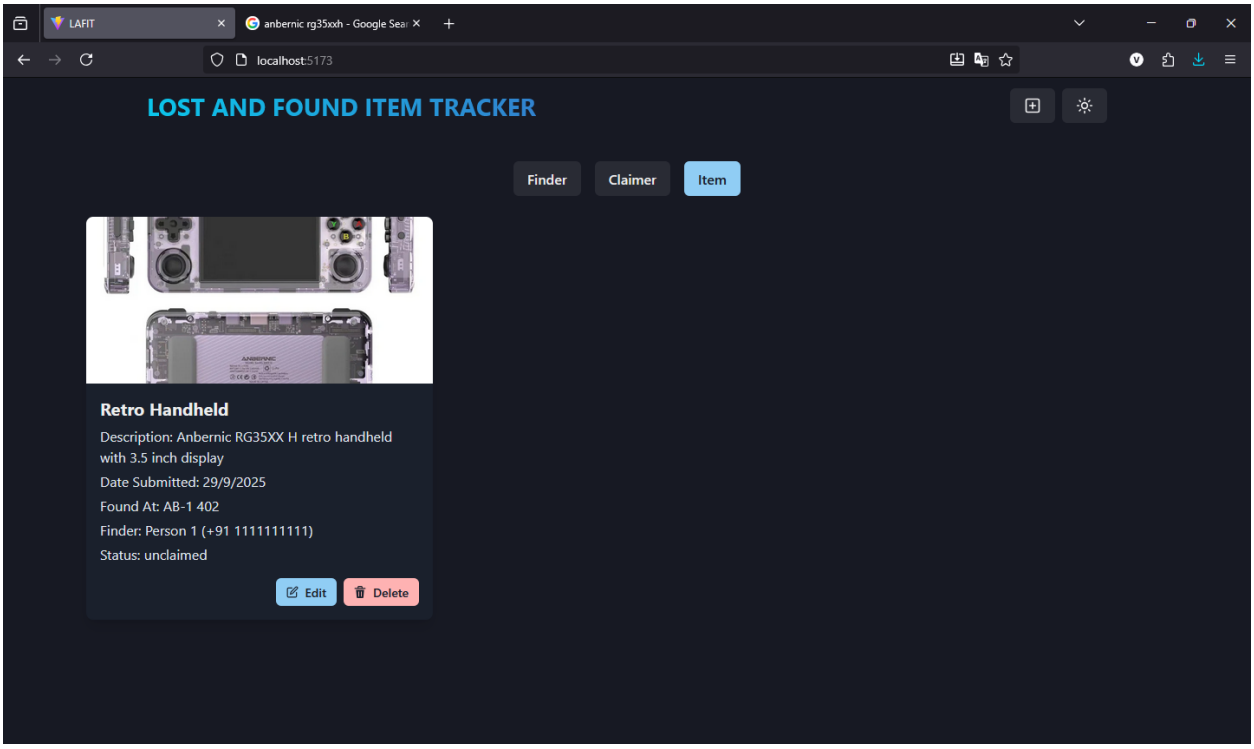

LAFIT
localhost:5173/create-claimer
LOST AND FOUND ITEM TRACKER
Create New Claimer
Person 2
+91 2222222222
person2
Add Claimer


LAFIT
localhost:5173/create-claimer
LOST AND FOUND ITEM TRACKER
Create New Claimer
Name
Contact Info
Username
Add Claimer
Success
Claimer created successfully

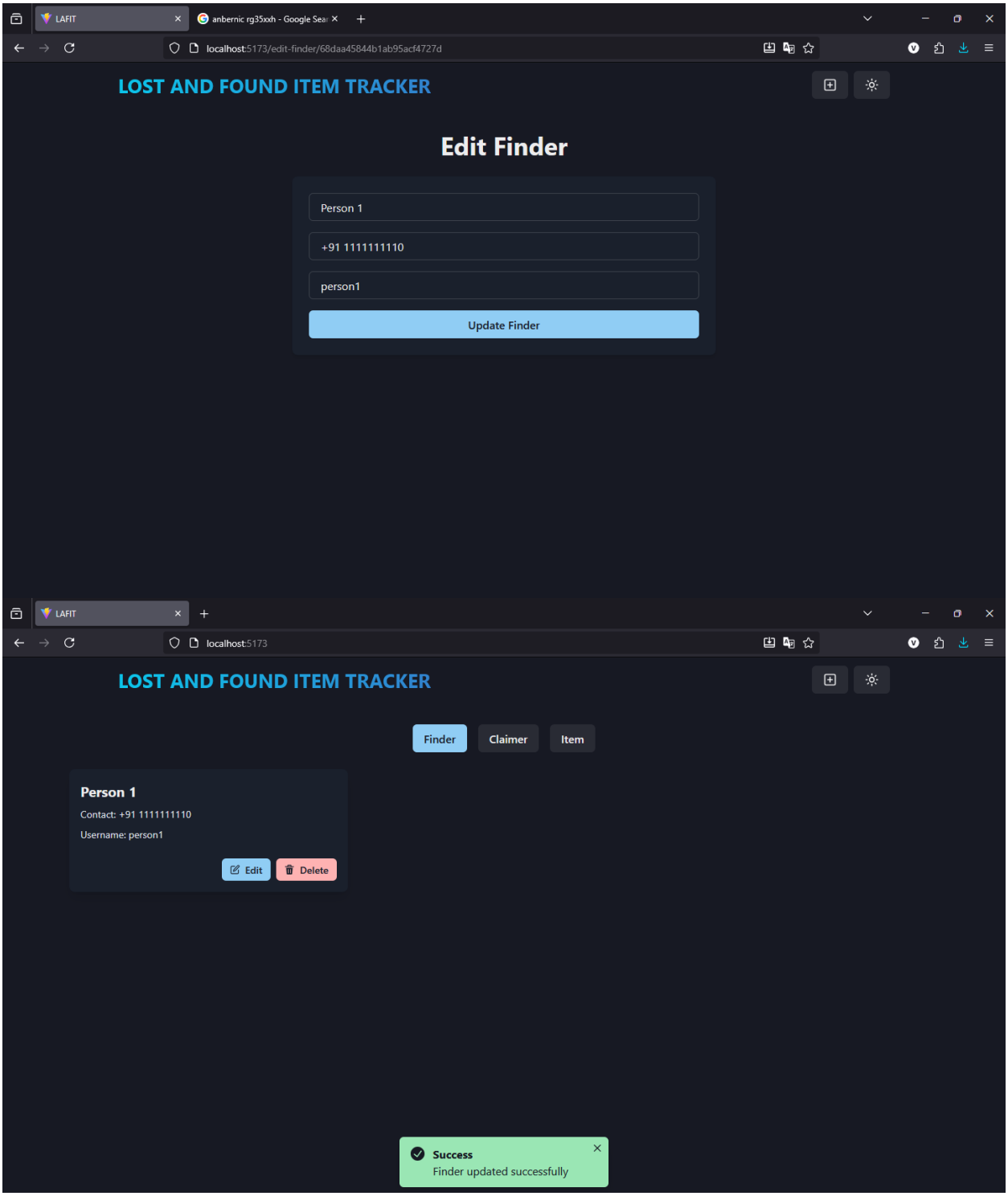

LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Person 2
Contact: +91 2222222222
Username: person2
Edit
Delete

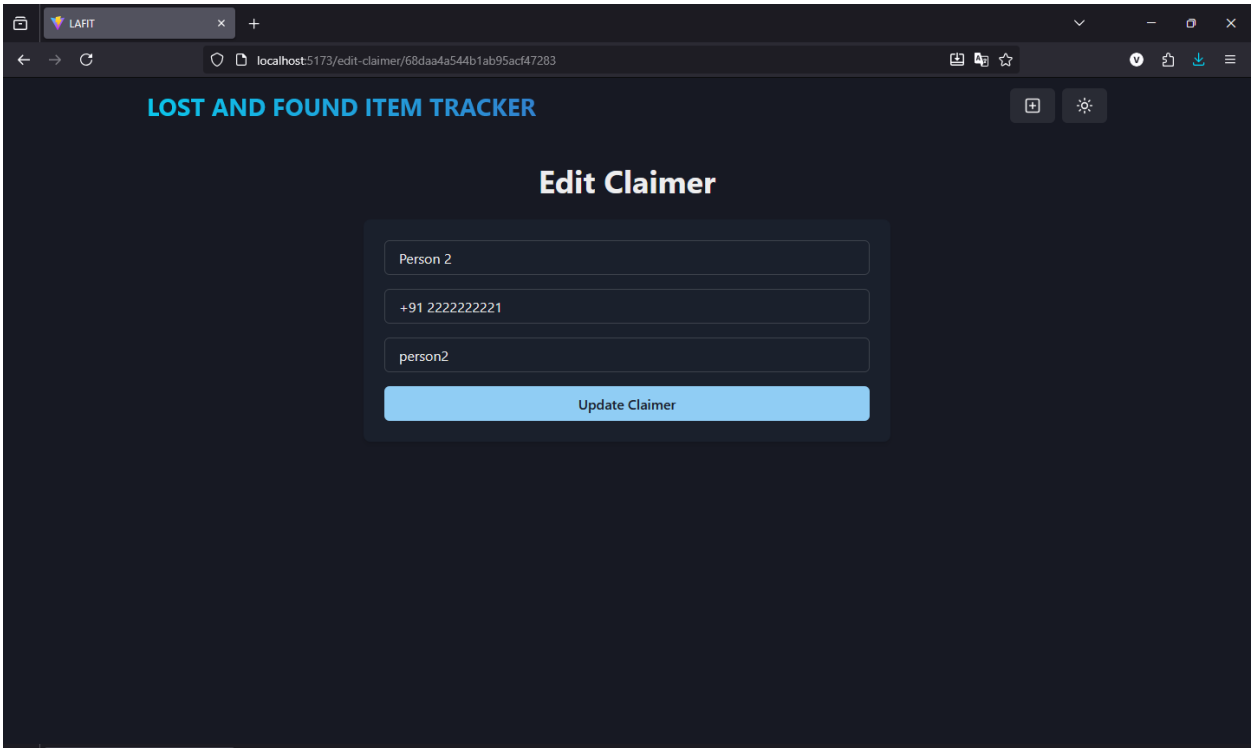

LAFIT
anbernic rg35xxh - Google Sear
localhost:5173/create-item
LOST AND FOUND ITEM TRACKER
Create New Item
Retro Handheld
Anbernic RG35XX H retro handheld with 3.5 inch display
AB-1 402
,749ae784-60d4-4ce3-95bd-7ec77064009f.jpg?v=1758528278&width=800
Unclaimed
person1
person1
Add Item

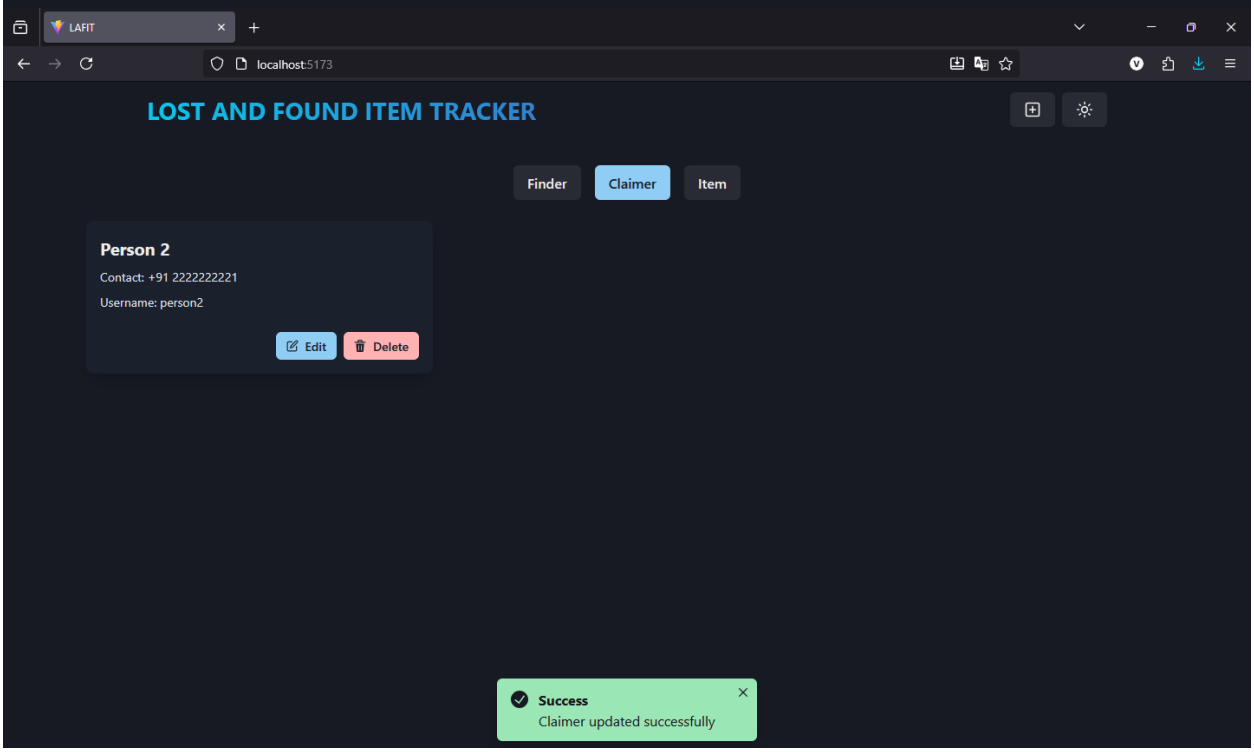

LAFIT
anbernic rg35xxh - Google Sear
localhost:5173/create-item
LOST AND FOUND ITEM TRACKER
Create New Item
Name
Description (optional)
Found At (optional)
Image URL
Unclaimed
Search Finder by Username
Add Item
Success
Item created successfully

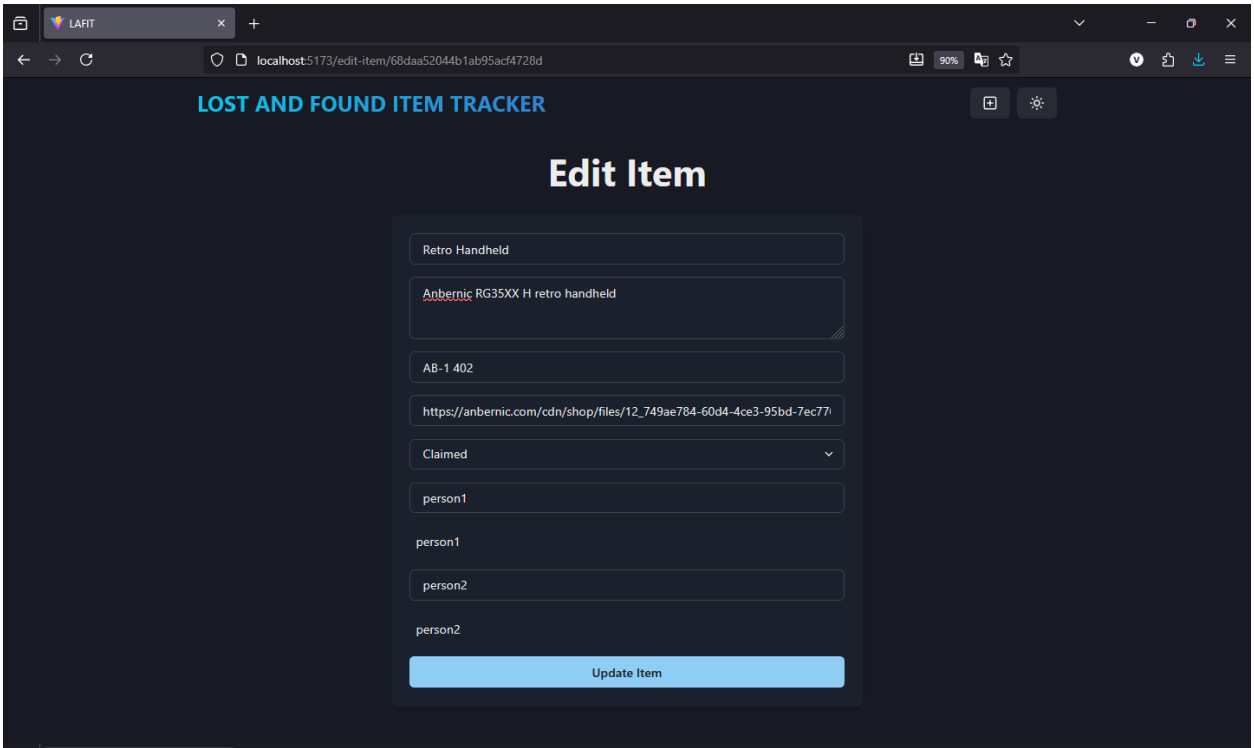

LAFIT
anbernic rg35xxh - Google Sear
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Retro Handheld
Description: Anbernic RG35XX H retro handheld with 3.5 inch display
Date Submitted: 29/9/2025
Found At: AB-1 402
Finder: Person 1 (+91 1111111111)
Status: unclaimed
Edit
Delete

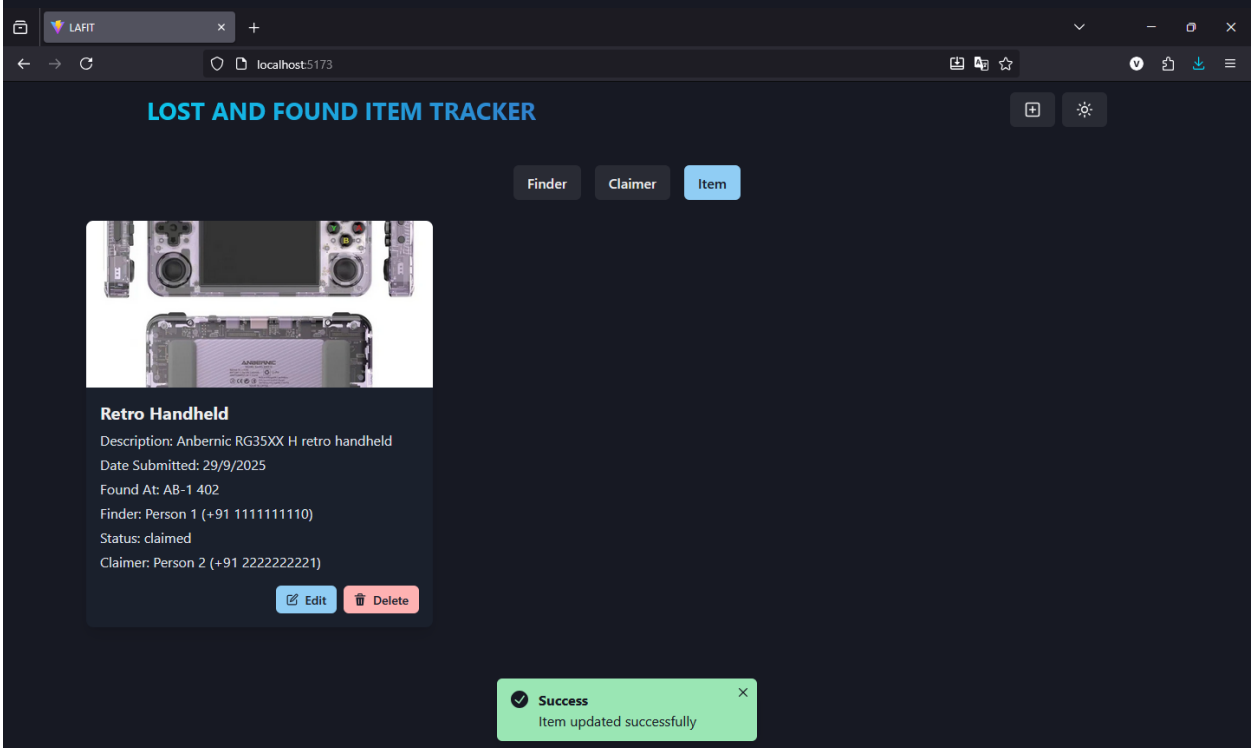
c. Editing of Finder, Claimer, and Item details:




LAFIT
localhost:5173/edit-claimer/68daa4a544b1ab95acf47283
LOST AND FOUND ITEM TRACKER
Edit Claimer
Person 2
+91 2222222221
person2
Update Claimer


LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Person 2
Contact: +91 2222222221
Username: person2
Edit
Delete
Success
Claimer updated successfully


LAFIT
localhost:5173/edit-item/68daa52044b1ab95acf4728d
90%
LOST AND FOUND ITEM TRACKER
Edit Item
Retro Handheld
Anbernic RG35XX H retro handheld
AB-1 402
https://anbernic.com/cdn/shop/files/12_749ae784-60d4-4ce3-95bd-7ec77
Claimed
person1
person1
person2
person2
Update Item

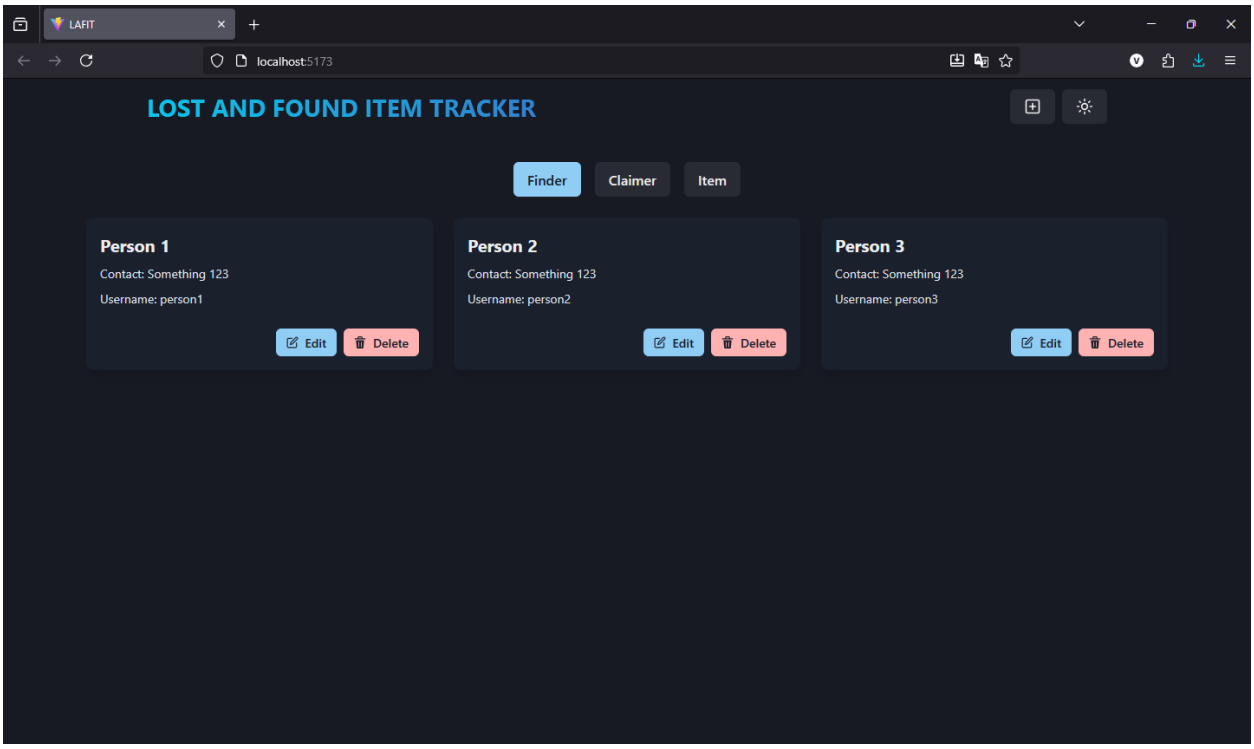

LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Retro Handheld
Description: Anbernic RG35XX H retro handheld
Date Submitted: 29/9/2025
Found At: AB-1 402
Finder: Person 1 (+91 1111111110)
Status: claimed
Claimer: Person 2 (+91 2222222221)
Edit
Delete
Success
Item updated successfully

4. Tools/Technologies used:

- **MongoDB:** A NoSQL database which is used to store data of claimers, finders, and items.
- **ExpressJS:** A JavaScript framework built on top of Node.JS to simplify networking (example, routing).
- **Node.JS:** A JavaScript framework that allows JavaScript code to run without a browser or otherwise used to build the backend.
- **ReactJS:** A JavaScript framework used to build the frontend with UI/UX elements with trackable states.
- **VSCode:** An IDE developed using Electron to edit code belonging to various languages with support for extensions for scalability.

5. SOLID principles application (Screenshots):

a. **Single Responsibility Principle:** Distinction of finders, claimers, and items.



b. **Open/Closed Principle:** The Schema of finders, claimers and items remain constant, but the APIs can be modified.

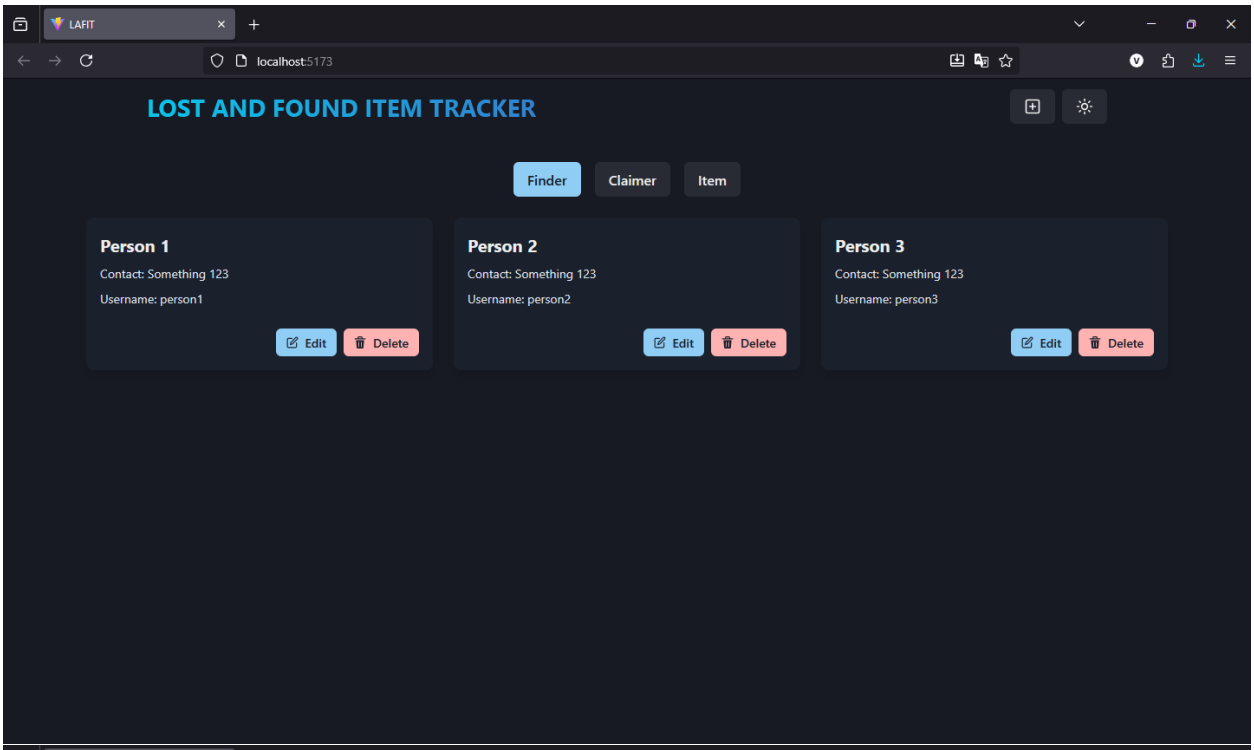
```
import mongoose from "mongoose";

const finderSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true
    },
    contactInfo: {
        type: String,
        required: true
    },
    userName: {
        type: String,
        required: true,
        unique: true
    }
});

const Finder = mongoose.model('Finder', finderSchema);
export default Finder;
```

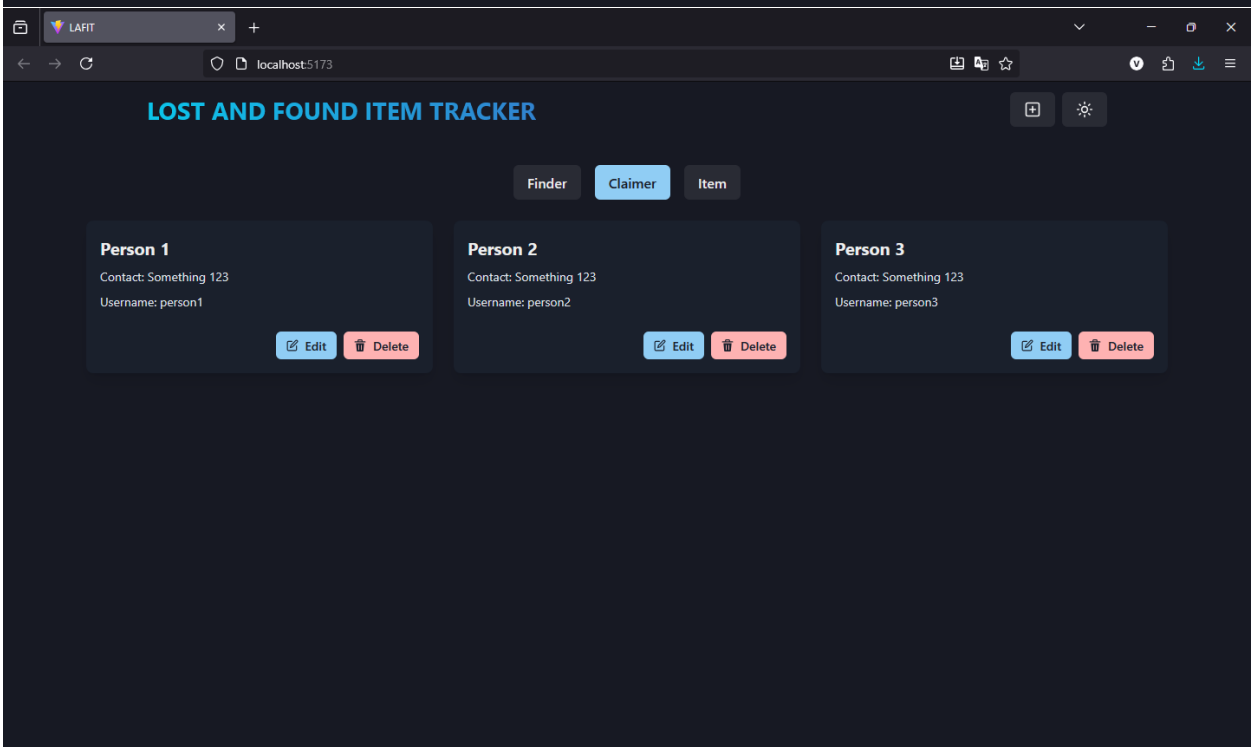
```python
import mongoose from "mongoose";

const claimerSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true
    },
    contactInfo: {
        type: String,
        required: true
    },
    userName: {
        type: String,
        required: true,
        unique: true
    }
});

const Claimer = mongoose.model('Claimer', claimerSchema);
export default Claimer;
```

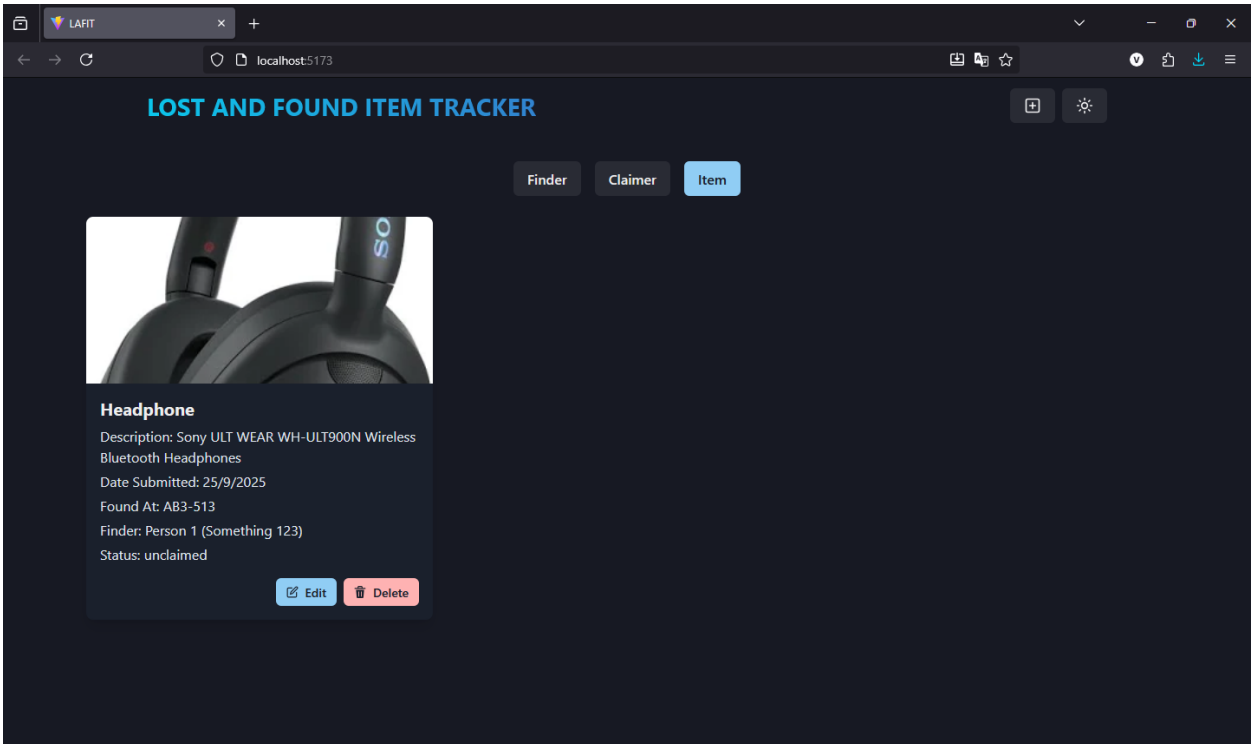
```python
import mongoose from "mongoose";

const itemSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true
    },
    description: {
        type: String
    },
    dateSubmitted: {
        type: Date,
        default: Date.now
    },
    foundAt: {
        type: String
    },
    image: {
        type: String,
        required: true
    },
    status: {
        type: String,
        enum: ['unclaimed', 'claimed'],
        default: 'unclaimed',
    },
    foundBy: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'Finder',
        required: true
    },
    claimedBy: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'Claimer'
    }
}, {
    timestamps: true
});

const Item = mongoose.model('Item', itemSchema);
export default Item;
```

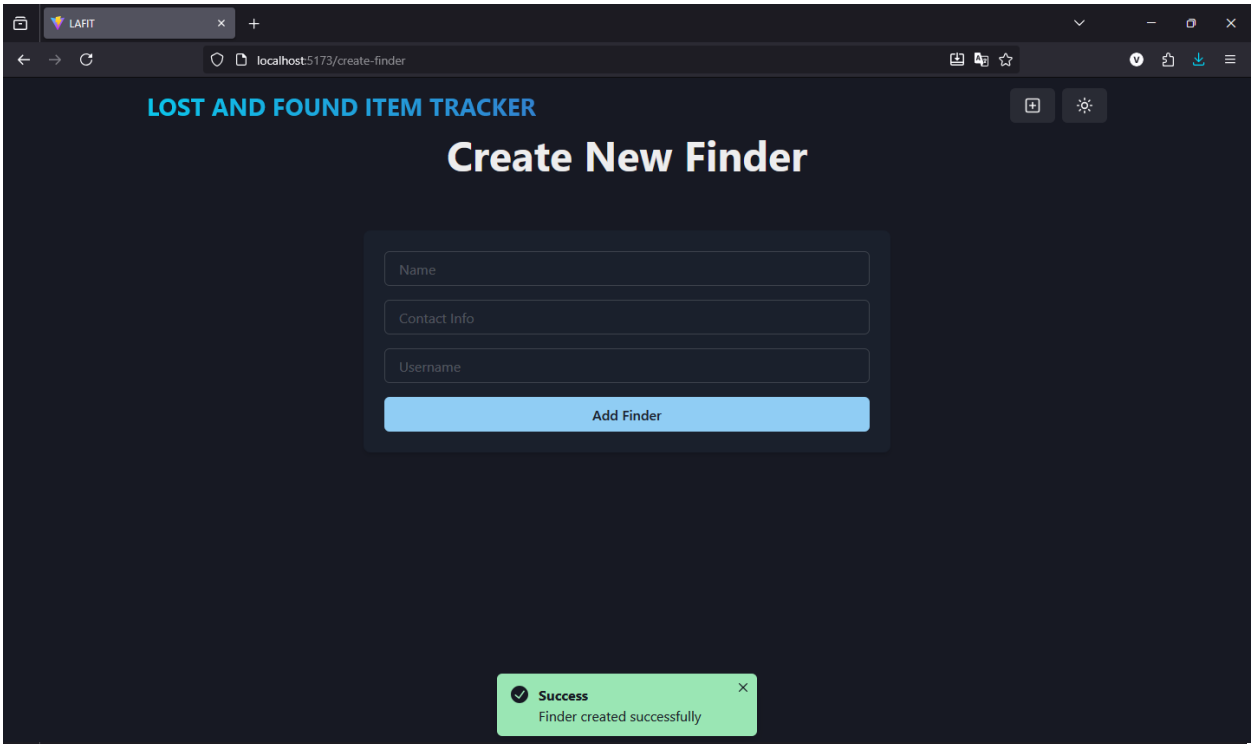
c. **Liskov's Substitution Principle:** Since we don't need to use classes here, we do not need to be concerned about inheritance, child-parent classes, etc.

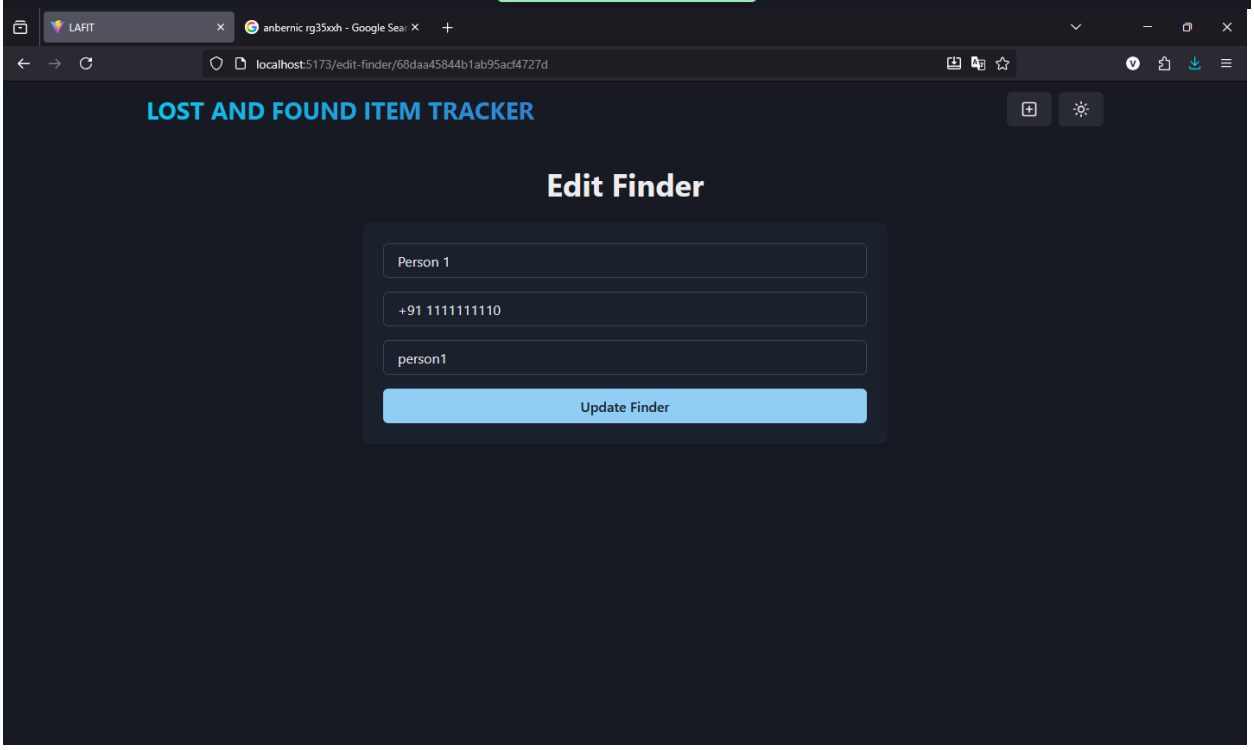
d. **Interface Segregation Principle:** The distinction between the view/delete pages, create pages, and edit pages have been clearly defined between finders, claimers and items.

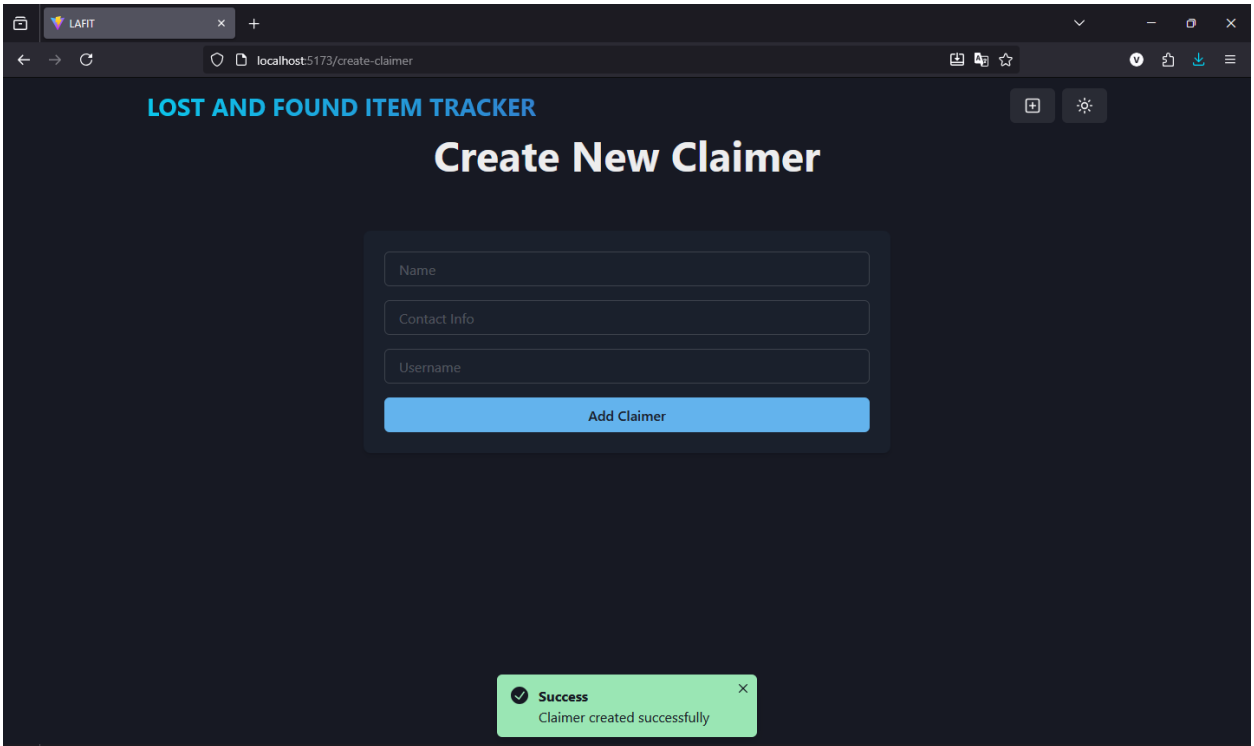


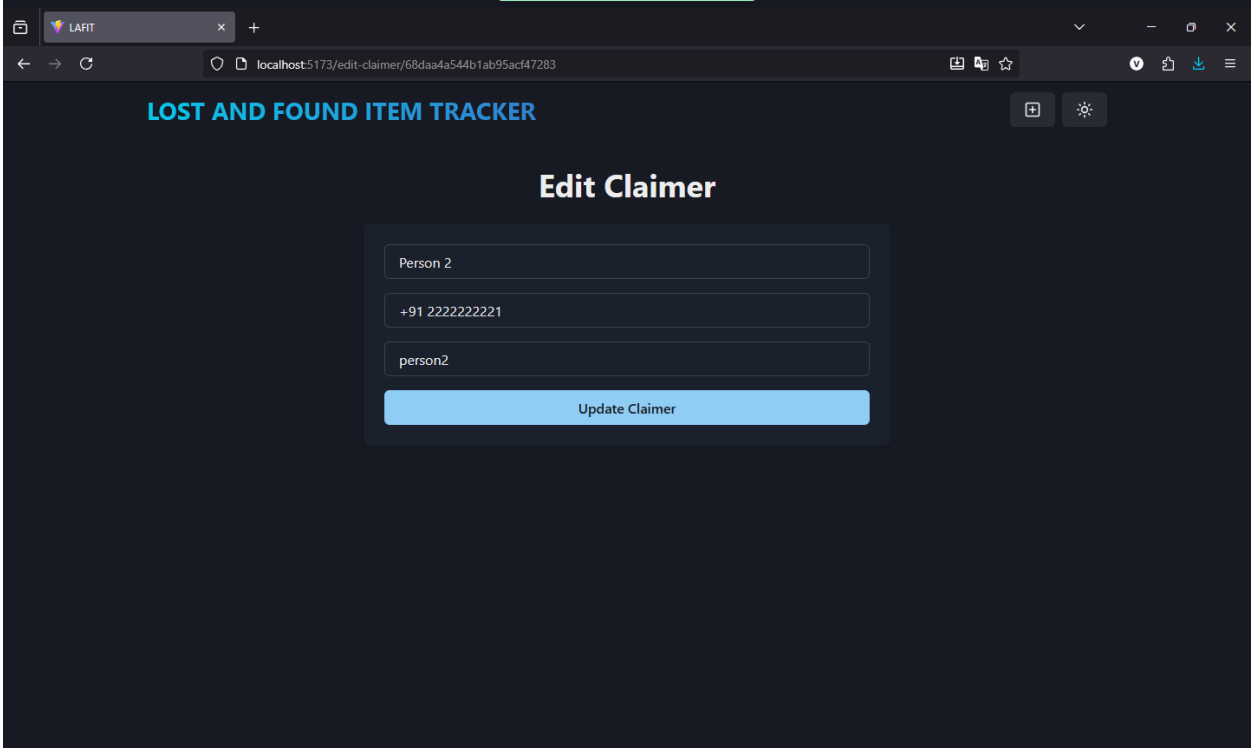


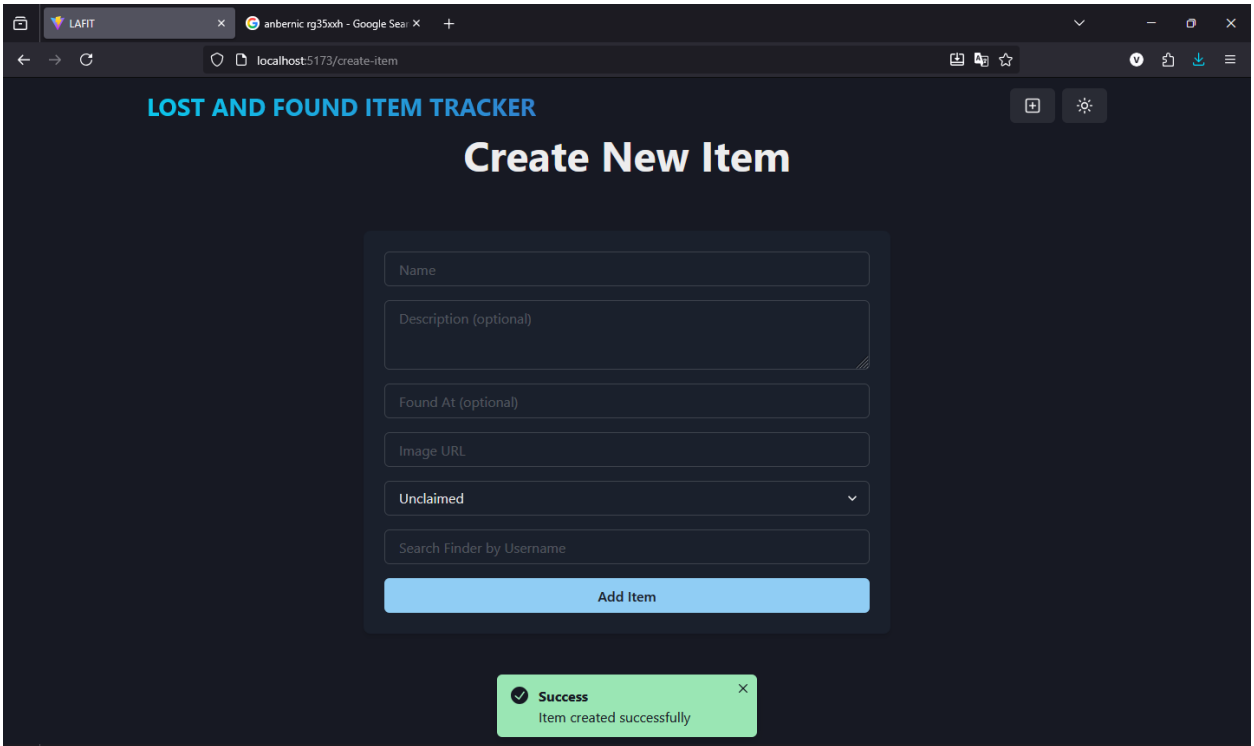

LAFIT
localhost:5173
LOST AND FOUND ITEM TRACKER
Finder
Claimer
Item
Headphone
Description: Sony ULT WEAR WH-ULT900N Wireless Bluetooth Headphones
Date Submitted: 25/9/2025
Found At: AB3-513
Finder: Person 1 (Something 123)
Status: unclaimed
Edit
Delete

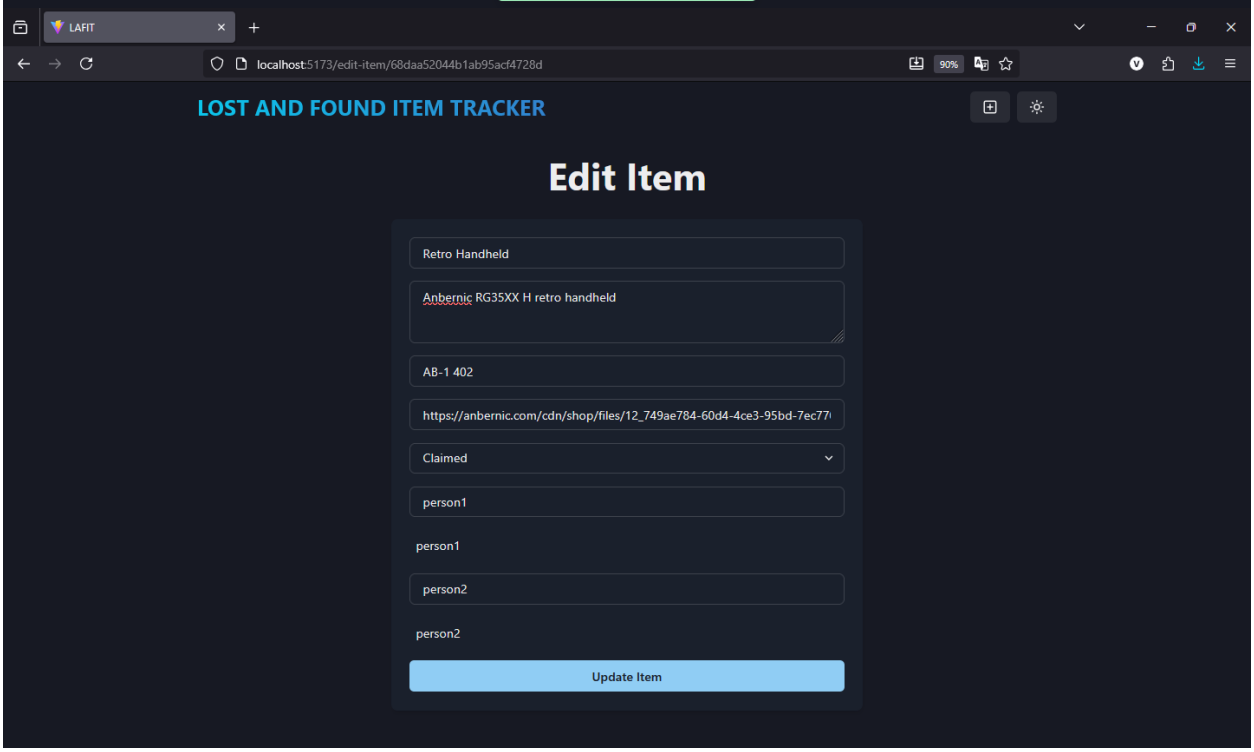

LAFIT
localhost:5173/create-finder
LOST AND FOUND ITEM TRACKER
Create New Finder
Name
Contact Info
Username
Add Finder
Success
Finder created successfully


LAFIT
anbernic rg35xxh - Google Sear
localhost:5173/edit-finder/68daa45844b1ab95acf4727d
LOST AND FOUND ITEM TRACKER
Edit Finder
Person 1
+91 1111111110
person1
Update Finder


LAFIT
localhost:5173/create-claimer
LOST AND FOUND ITEM TRACKER
Create New Claimer
Name
Contact Info
Username
Add Claimer
Success
Claimer created successfully


LAFIT
localhost:5173/edit-claimer/68daa4a544b1ab95acf47283
LOST AND FOUND ITEM TRACKER
Edit Claimer
Person 2
+91 2222222221
person2
Update Claimer


LAFIT
anbernic rg35xxh - Google Sear
localhost:5173/create-item
LOST AND FOUND ITEM TRACKER
Create New Item
Name
Description (optional)
Found At (optional)
Image URL
Unclaimed
Search Finder by Username
Add Item
Success
Item created successfully


LAFIT
localhost:5173/edit-item/68daa52044b1ab95acf4728d
90%
LOST AND FOUND ITEM TRACKER
Edit Item
Retro Handheld
Anbernic RG35XX H retro handheld
AB-1 402
https://anbernic.com/cdn/shop/files/12_749ae784-60d4-4ce3-95bd-7ec77
Claimed
person1
person1
person2
person2
Update Item

e. **Dependency Inversion Principle:** This is impossible to implement in a language like JavaScript, as it is prototype based rather than being class based like Java.