# Software Design Task 4 Database Implementation

## Features
- User Authentication: Users can register and log in with email and password.
- Profile Management: Users can update their profile details, including address, skills, and availability.
- Event Creation: Admins can create events with details like event name, location, required skills, and urgency.
- Volunteer Matching: Matches volunteers to events based on skills and availability.
- Notifications: Provides notifications for upcoming events and reminders.
- Volunteer History: Tracks and displays volunteer participation history.

## Technologies Used
- Node.js: JavaScript runtime used for building the backend server.
- Express: Web framework for Node.js to create and manage routes.
- Jest & Supertest: Testing libraries for unit and integration testing.
- Other Packages:
- body-parser: Parses incoming request bodies in a middleware.
- cors: Allows cross-origin requests for development.

## server.js
- It uses Jest and Supertest to perform unit and integration tests, ensuring that each API route behaves as expected. The tests cover scenarios like successful and failed user registration, login, profile updates, event creation, and volunteer matching, verifying the robustness and reliability of the backend code.

### Prerequisites
- Node.js (v14 or higher)
- MongoDB

### Cloning the Repository

1. Clone the repository:

```bash
git clone https://github.com/ahussein0/task4.git
cd task4test
```

## Install Dependencies
- Make sure you have Node.js and npm installed. Install the necessary packages by running:

```
npm install
```

```
npm install express
```

This will install all dependencies listed in package.json, including:
  - express

  - body-parser
  - cors
  - jest (for testing)
  - supertest (for testing)

## Launching the server locally

- To launch the server locally, open the terminal in your project directory and run:

``` node server.js ```

- This will start the server, and it will be accessible at http://localhost:3001

##MongoDB Integration

-The application uses MongoDB as the database to store and manage user, event, and volunteer data.

-Each user's information, including username, email, password, and timestamps, is stored in a MongoDB collection, which allows for easy retrieval and updates.

-MongoDB also stores event details and volunteer participation, making it simple to match volunteers to events based on stored skills and availability.

-To get started with MongoDB, ensure MongoDB is installed and running on your system, or connect to a MongoDB cloud instance. Update your MongoDB connection string in the configuration file as needed for your environment.