Customer API Documentation:

REQUIREMENTS

S.N.	Title	Version
1	PHP	8.2.9
2	React	18.3.1
3	MySQL	8.0.30
4	phpMyAdmin	5.2.1

Composer packages:

Title	Uses
"spatie/ignition": "^1.15",	Show error in a proper way
"league/container": "^4.2",	Main container for the app(Tested but used Illuminate Container)
"vlucas/phpdotenv": "^5.6",	access env variables
"illuminate/database": "^11.34"	Database toolkits for working with database
"illuminate/routing": "^11.34"	Provides routing for endpoints
"illuminate/events": "^11.34"	Provides an event dispatcher for handling
"illuminate/http": "^11.34"	Handles HTTP requests and responses, including routing and middleware.
"illuminate/validation": "^11.34"	Offers powerful data validation rules and error management for your application (Not used, custom class adds)
"illuminate/view": "^11.34"	Renders dynamic templates using Laravel's Blade templating engine or other view engines(Not usable for api)
"laravel/serializable-closure": "^2.0"	Enables secure serialization and unserialization of PHP closures (not used)

Frontend packages used:

Title	Uses
"@tanstack/react-table": "^8.20.5"	creates :datables
"axios": "^1.7.9"	promise-based HTTP client for making requests to APIs
"dotenv": "^16.4.7"	access env variables
"react": "^18.3.1"	JavaScript library for building user interfaces
"react-dom": "^18.3.1"	Rendering DOM React components
"react-icons": "^5.4.0"	Provides icons for application
"react-router-dom": "^7.0.2"	Routing library for React that enables navigate and dynamic URL

Url for Application (After configuration)

Frontend: http://localhost:5173/
Backend: http://localhost:8000/

Database configuration:

- Navigate to phpmyadmin for database
  http://localhost/phpmyadmin

- Create database with name **customer-app**
- Import database with **customer-app.sql** file provided
- You are done with database configuration

How to run Backend API:

- Navigate to backend folder
- Open terminal on the same directory and run following commands:

```
composer install
php -S localhost:8000 -t public/
```

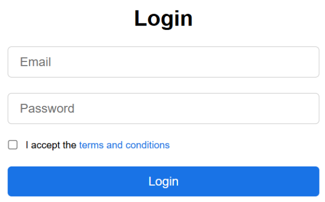
- You are done

How to run Frontend APP:

- Navigate to frontend folder
- Open terminal on the same directory and run following commands:

```
yarn install
yarn dev
```

- Navigate to url http://localhost:5173/
- You will see login page as:

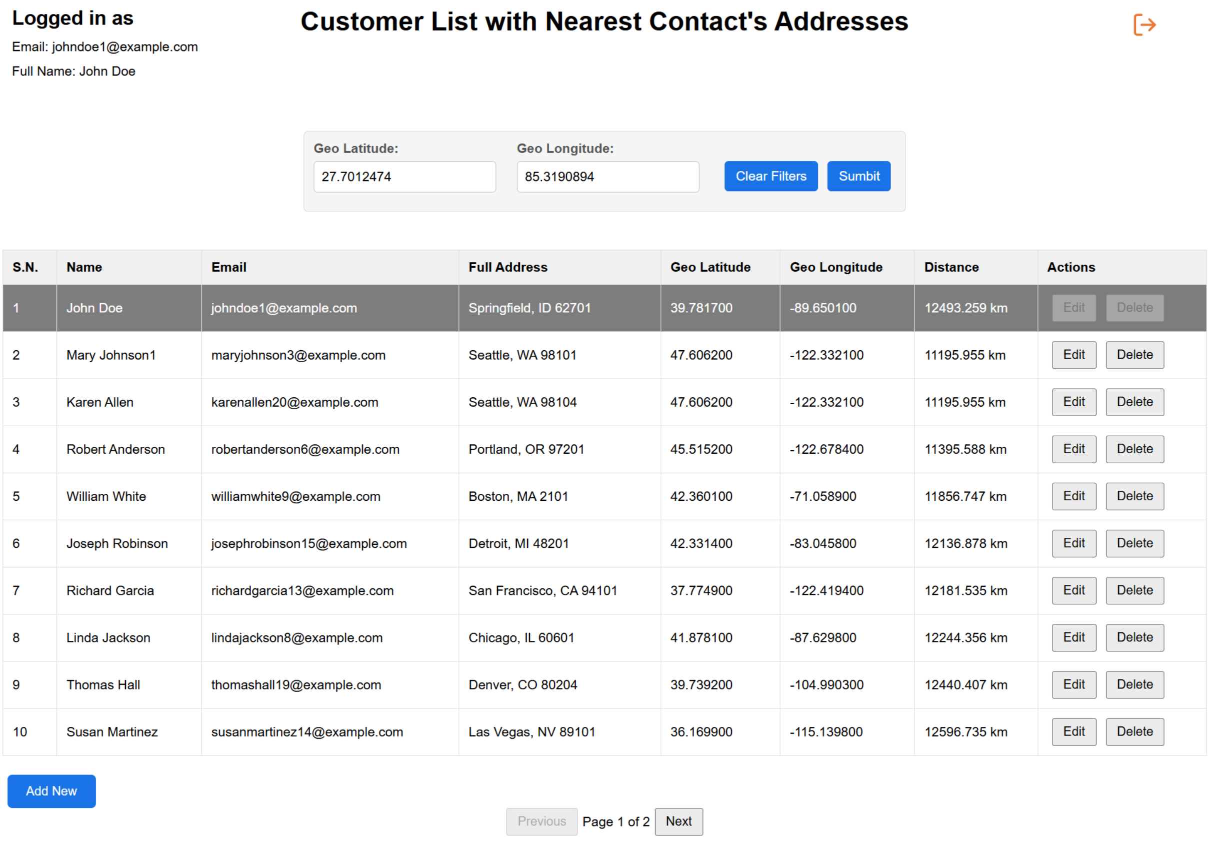


Login

- Next step is used any email provided on a database
  Use credentials as:
  - email: johndoe1@example.com
    password: password

  - email: janesmith2@example.com
    password: password

- If you are logged in you will redirected to **customer-list** page



Logged in as

Email: johndoe1@example.com

Full Name: John Doe

Customer List with Nearest Contact's Addresses

Geo Latitude: 27.7012474

Geo Longitude: 85.3190894

S.N.	Name	Email	Full Address	Geo Latitude	Geo Longitude	Distance	Actions
1	John Doe	johndoe1@example.com	Springfield, ID 62701	39.781700	-89.650100	12493.259 km	Edit Delete
2	Mary Johnson1	maryjohnson3@example.com	Seattle, WA 98101	47.606200	-122.332100	11195.955 km	Edit Delete
3	Karen Allen	karenallen20@example.com	Seattle, WA 98104	47.606200	-122.332100	11195.955 km	Edit Delete
4	Robert Anderson	robertanderson6@example.com	Portland, OR 97201	45.515200	-122.678400	11395.588 km	Edit Delete
5	William White	williamwhite9@example.com	Boston, MA 2101	42.360100	-71.058900	11856.747 km	Edit Delete
6	Joseph Robinson	josephrobinson15@example.com	Detroit, MI 48201	42.331400	-83.045800	12136.878 km	Edit Delete
7	Richard Garcia	richardgarcia13@example.com	San Francisco, CA 94101	37.774900	-122.419400	12181.535 km	Edit Delete
8	Linda Jackson	lindajackson8@example.com	Chicago, IL 60601	41.878100	-87.629800	12244.356 km	Edit Delete
9	Thomas Hall	thomashall19@example.com	Denver, CO 80204	39.739200	-104.990300	12440.407 km	Edit Delete
10	Susan Martinez	susanmartinez14@example.com	Las Vegas, NV 89101	36.169900	-115.139800	12596.735 km	Edit Delete

Note: First record will always be for the current login user.

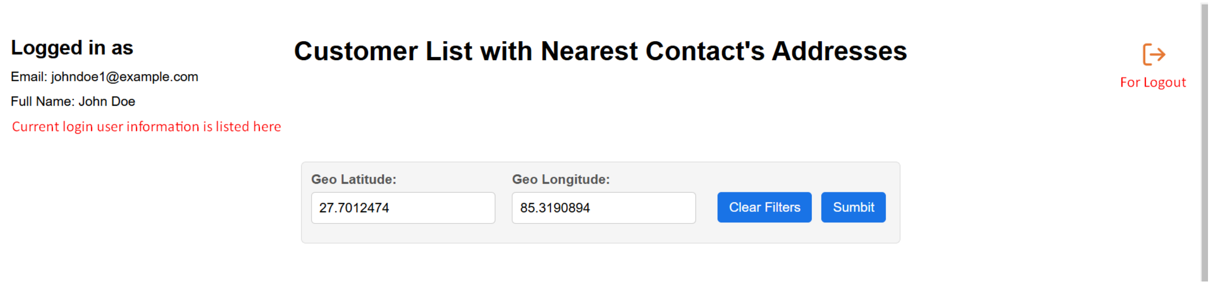
- Similarly the next users on a list will be the users with shortest distance calculated using geo locations (longitude and latitude)
- You can perform edit/delete or add new customers according to the field shown.

S.N.	Fields	Used for
1	S.N.	For showing serial number
2	Email	Customer email
3	Full Address	joins(city,state,zipcode) for customer address
4	Geo Latitude	Address geo location provided on database
5	Geo Longitude	Address geo latitude provided on database
6	Distance	Calculated using geo distance formula and append to fields
7	Actions	Provides actions feature for Edit and Delete button

- After clicking **Delete** button, user will be deleted from database too and page will be reloaded.
- After clicking on **Edit** and **Add New** new modal pop up with fields for edit and to create
- If any validation fails, form won't submitted, error is not shown for now.(later on) Though backend validation is done. You can get payloads for error.
- Top 10 records are shown in table

<u>Use of search box:</u>

- Default geo latitude and longitude has been provided, as you provide new value, records will be filter accordingly to geo locations



Logged in as

Email: johndoe1@example.com

Full Name: John Doe

Current login user information is listed here

Customer List with Nearest Contact's Addresses

For Logout

Geo Latitude: 27.7012474

Geo Longitude: 85.3190894

- After logout session will be deleted from backend application and redirect to login page

(If any confusion regarding documentation, please feel free to contact me)

Thank You
Documentation for Customer API

Personal Information
Pramesh Dhakal
Backend Laravel/PHP Developer
Email: impirameshdhakal@gmail.com
Phone: +977 9863701484

Date submitted: 2024-12-12