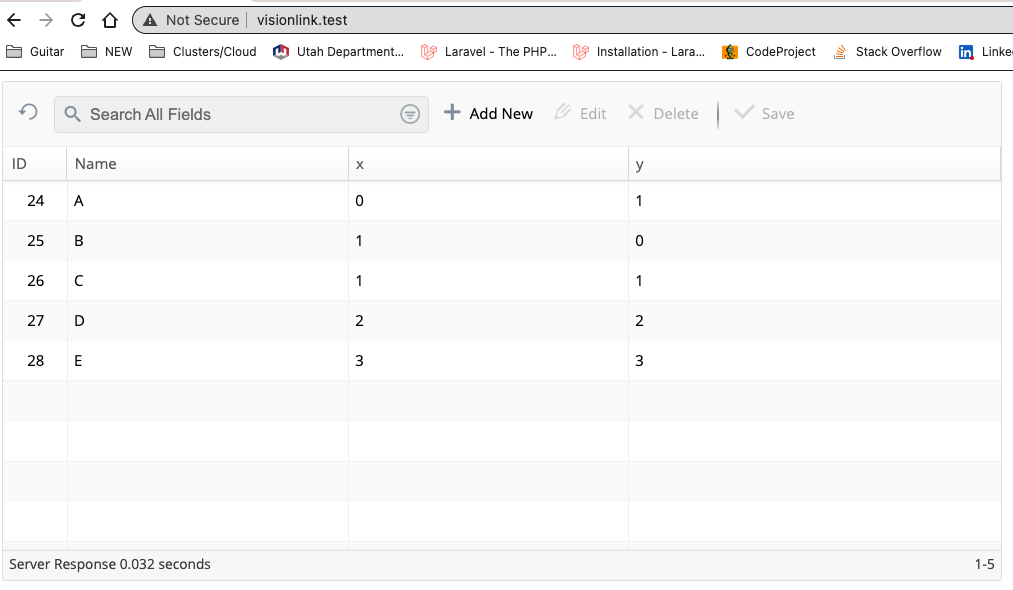
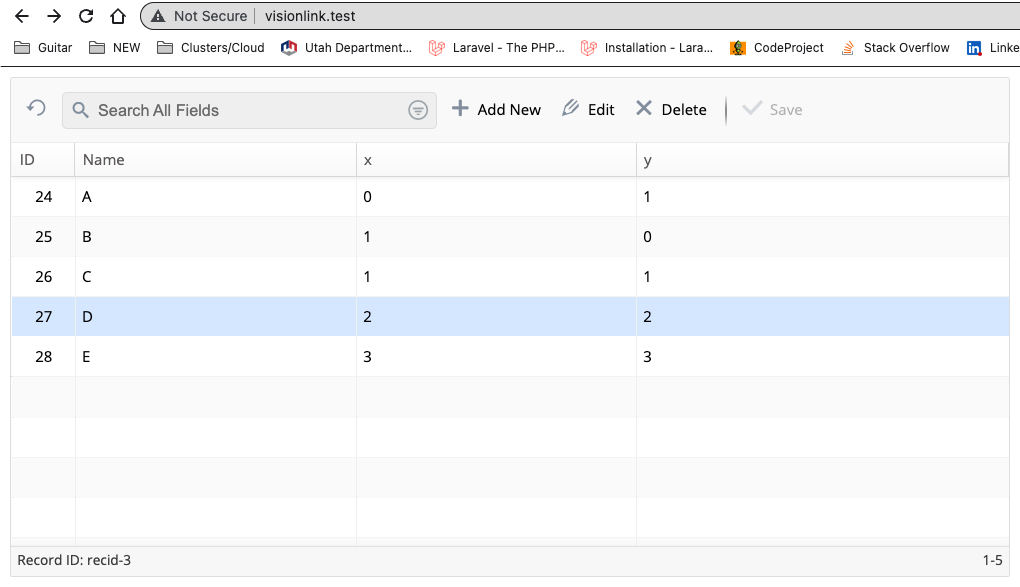
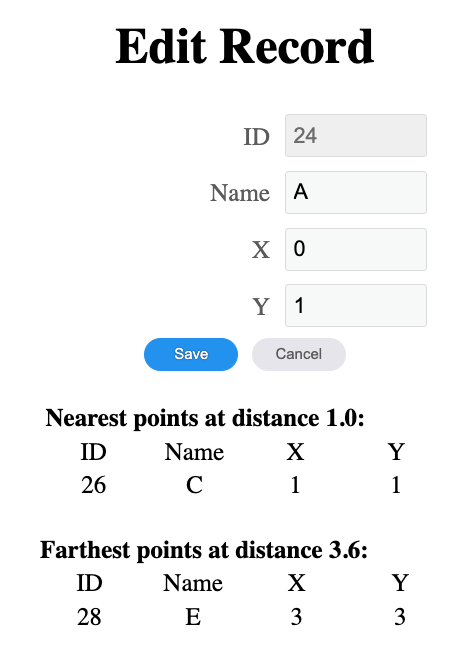
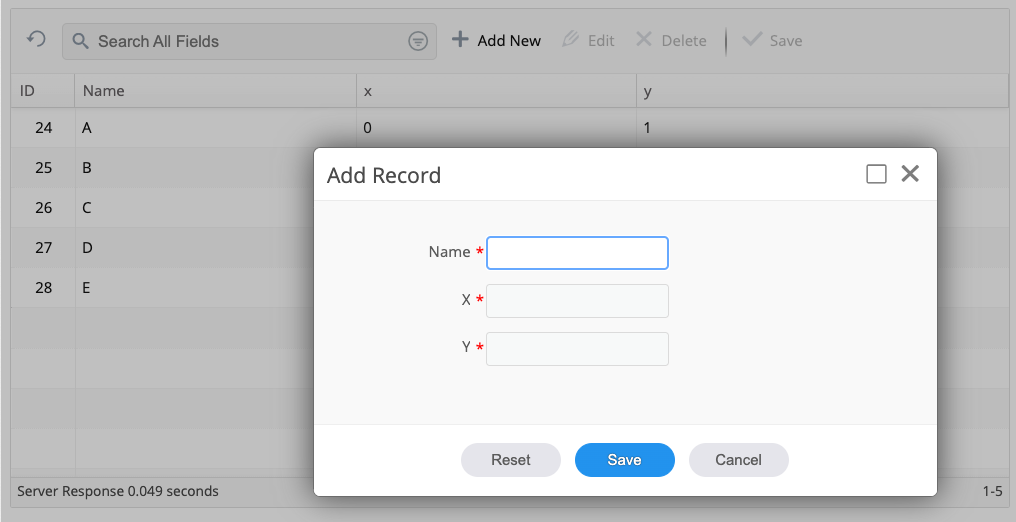
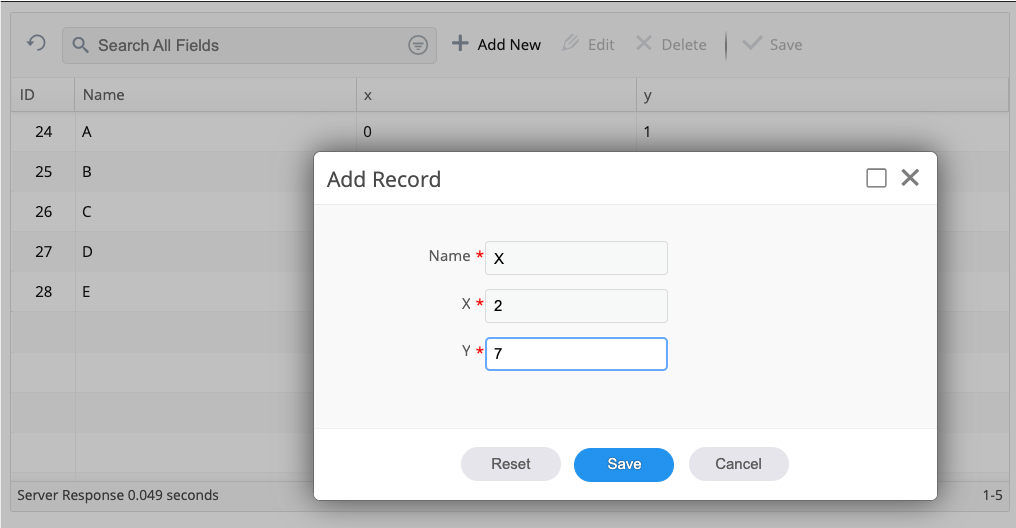
# Screen Shots

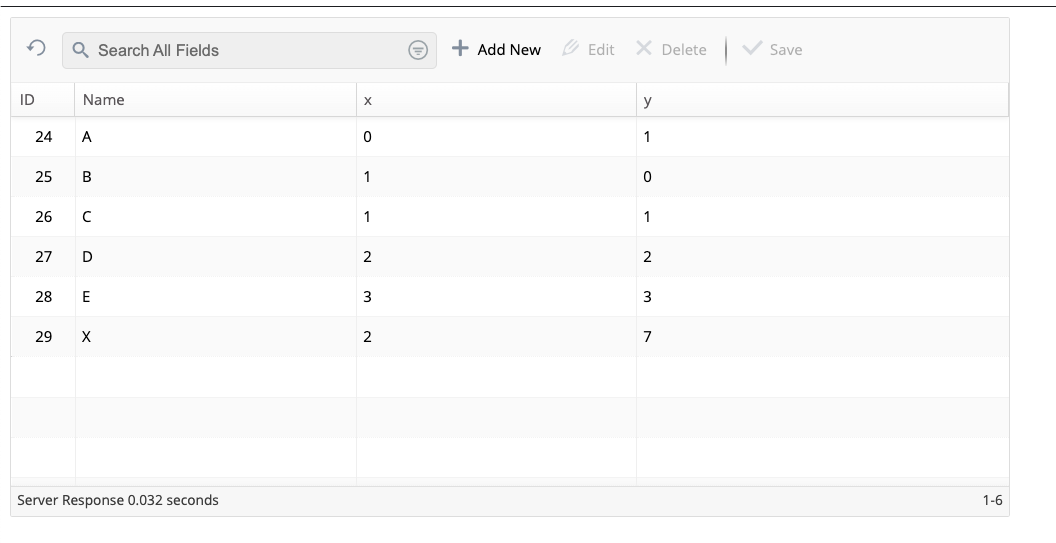


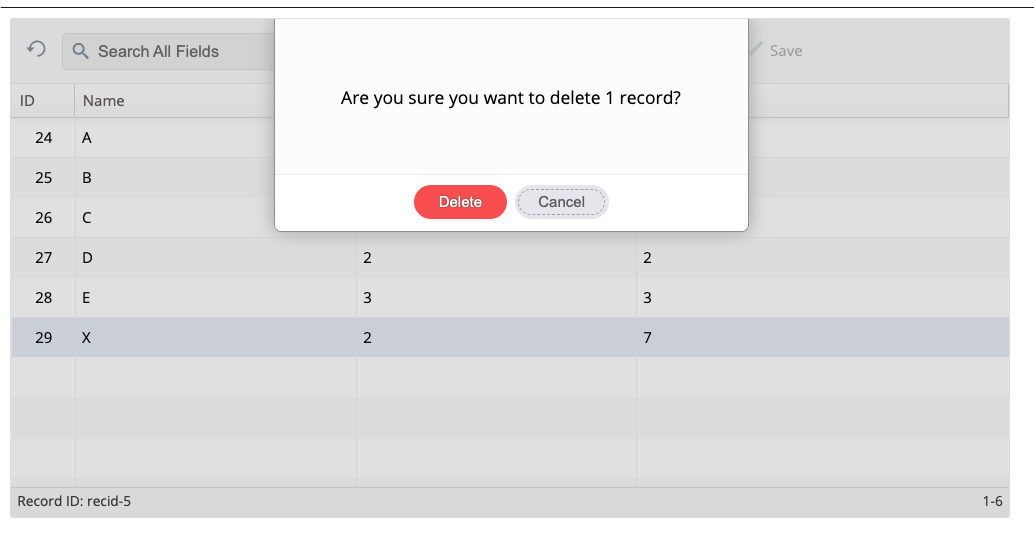


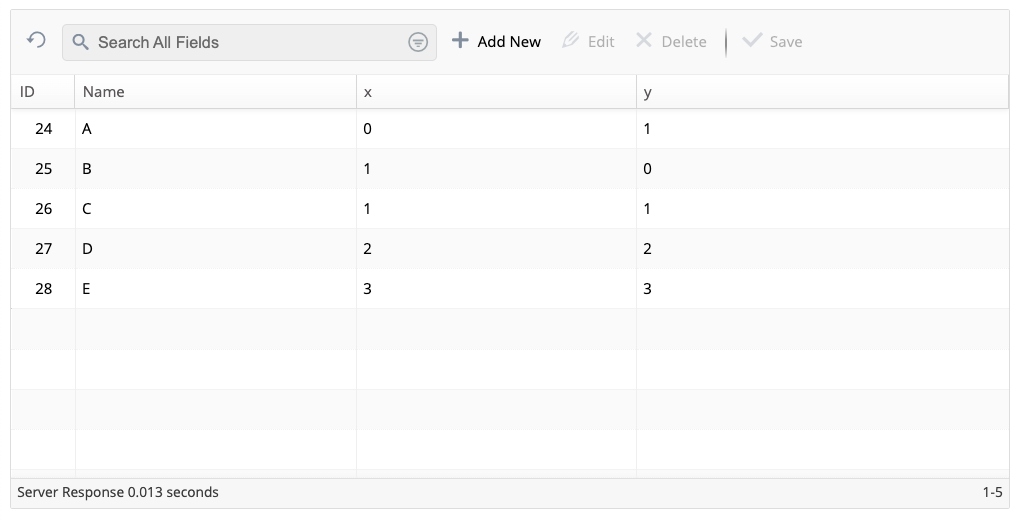












# Trial Task

Your task is to create a simple web application to display a set of points, and allow adding, editing and deleting points. You may use any language and framework you like. You should make the completed application available in a gitlab or github repository, and be prepared to discuss it during the interview.

psql postgres

create role visionlink login password 'secretpwd';

create database visionlink with owner 'visionlink';

grant all privileges on database visionlink to visionlink;

alter user visionlink with superuser;

visionlink=# alter user visionlink with password 'trialtask';

The application should assume that a PostgreSQL database already contains the table of points, with the following schema:

CREATE TABLE point (

id SERIAL PRIMARY KEY,

name text NOT NULL,

x integer NOT NULL,

y integer NOT NULL

);

insert into point (name, x, y) values ('A', 0, 1);

insert into point (name, x, y) values ('B', 1, 0);

insert into point (name, x, y) values ('C', 1, 1);

insert into point (name, x, y) values ('D', 2, 2);

insert into point (name, x, y) values ('E', 3, 3);

Each point represents a named location on a plane, with integer coordinates. The application will have a configuration file of some sort where the connection string to the database is found.

**The home page of the application will display the existing points in a table, sorted by name.** Below is an example display:

|  |  |  |
| --- | --- | --- |
| **Name** | **X** | **Y** |
| A | 0 | 1 |
| B | 1 | 0 |
| C | 1 | 1 |
| D | 2 | 2 |
| E | 3 | 3 |

Feel free to style the table in a way that you consider aesthetically pleasing. Clicking any of the rows will bring up a page for editing or deleting that point. There will also be a way to add a new point, indicated by a button marked “New” or “+”, or something similar.

The edit page will allow modification of the Name, X and Y values, like so:

|  |  |
| --- | --- |
| **Name** | D |
| **X** | 2 |
| **Y** | 2 |

Again, feel free to style as you consider appropriate.

Below the editing table, if there is at least one other point, there will be a list of the nearest point(s) and a list of the farthest point(s), like so:

Nearest points at distance 1.4:

|  |  |  |
| --- | --- | --- |
| **Name** | **X** | **Y** |
| C | 1 | 1 |
| E | 3 | 3 |

Farthest points at distance 2.2:

|  |  |  |
| --- | --- | --- |
| **Name** | **X** | **Y** |
| A | 0 | 1 |
| B | 1 | 0 |

The distance will be shown rounded to the nearest 0.1. If there is a unique nearest point, the heading will say “Nearest point at distance …”, and only that point will appear. Likewise, if there is a unique farthest point, the heading will say “Farthest point at distance …”.

As the user edits the X and Y values for the point being edited, the display of the nearest point(s) and farthest point(s) will update in real time. However, the point is not actually updated in the database until the user clicks the Save button.

The Save button will be disabled until some change has been made, and the data is valid (Name is nonblank and not matching the name of any other point, and X and Y are decimal integer values). Clicking Save will then update the point and return to the home page.

There will also be a Reset button that restores the current values for the point. This button will also be disabled until some change has been made. The newly entered data need not be valid.

There will also be a Delete button that deletes the point and returns to the home page.

There will also be a link that simply returns to the home page, discarding any changes.

The page to add a new point will be similar to the edit page, except that the Name, X and Y fields will start out blank, and there will be no Delete button. The Save button will create the new point with the data entered.