Objectives

Develop a contact manager

When a user first accesses the website, they can either login or sign up (register).

Once they are logged in with a valid account, in their list of contacts, they can:

* Create a new contact.
* Read the information of a contact.
* Update the information of a contact.
* Delete a contact.
* Search for contact records.

Requirements

* Remote Server
  + Domain Name: [contactsclassproject.xyz](http://contactsclassproject.xyz/) (Used for testing for now)
* PowerPoint Slides (in Google for now for collaboration purposes)
  + Use Case diagram
  + Gantt chart
  + Entity relationship diagram (ERD)
* Used Postman for API usage.

Remote Server Steps

* Buy a remote server through Digital Ocean.
* Buy a domain name through NameCheap.
* Connect the domain name to the remote server.
  + In Name Cheap, go to advanced DNS.
  + Click on the Add New Record button
  + Select URL Redirect Record from the drop-down menu for Type, put @ (your bare domain e.g., yourdomain.tld) for Host and use http://ipaddress:port for Value (IP address and port should be replaced with actual values):
    - For this assignment, the port is 80.
  + Use masked redirect to display your domain name.
  + Click on Save All Changes once the setup is completed. It may take up to 30 minutes for the new settings to be accepted.

Database Creation Steps

* **Open command prompt and run SSH**
* **ssh root@MyDomainOrIPAddress**
* **Install Putty in order to easily SSH in to your droplet**
* Then enter your password.
  + Please note that everything you essentially need is already installed in the droplet. This includes MySQL Apache, and PHP.
* Connect to MySQL: **mysql -u root -p** (then enter your password)
  + ***Note: There are three levels of password you will need to think about. This is the first, which gives you access to MySQL from the command line.***
* Here are the steps to create the database, tables, and working data.
* 1. Create database
  + create database contactsWebsite;
  + use contactsWebsite;
* 2. Create tables
  + User Table
    - Contains the login information for all users of the site. (More can be added if preferred)
      * Username
      * Password
      * First name
      * Last name

CREATE TABLE `contactsWebsite`.`Users` ( `ID` INT NOT NULL AUTO\_INCREMENT , `FirstName` VARCHAR(50) NOT NULL DEFAULT '' , `LastName` VARCHAR(50) NOT NULL DEFAULT '' , `Username` VARCHAR(50) NOT NULL DEFAULT '' , `Password` VARCHAR(50) NOT NULL DEFAULT '' , PRIMARY KEY (`ID`)) ENGINE = InnoDB;

* + - * CREATE TABLE `contactsWebsite `.`Users`
      * (
      * `ID` INT NOT NULL AUTO\_INCREMENT ,
      * `FirstName` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `LastName` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `Username` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `Password` VARCHAR(50) NOT NULL DEFAULT '' ,
      * PRIMARY KEY (`ID`)
      * ) ENGINE = InnoDB;
  + Contacts Table
    - Contains all contact entries that were created by every user. (Remember this backend, this isn’t a problem)
      * First Name
      * Last Name
      * Email
      * Phone
      * UserID (Id of the user that created the contact)

CREATE TABLE `contactsWebsite`.`Contacts`(`ID` INT NOT NULL AUTO\_INCREMENT , `FirstName` VARCHAR(50) NOT NULL DEFAULT '' , `LastName` VARCHAR(50) NOT NULL DEFAULT '' , `Phone` VARCHAR(50) NOT NULL DEFAULT '' , `Email` VARCHAR(50) NOT NULL DEFAULT '' , `UserID` INT NOT NULL DEFAULT '0' , PRIMARY KEY (`ID`)) ENGINE = InnoDB;

* + - * CREATE TABLE `contactsWebsite `.`Contacts`
      * (
      * `ID` INT NOT NULL AUTO\_INCREMENT ,
      * `FirstName` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `LastName` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `Phone` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `Email` VARCHAR(50) NOT NULL DEFAULT '' ,
      * `UserID` INT NOT NULL DEFAULT '0' ,
      * PRIMARY KEY (`ID`)
      * ) ENGINE = InnoDB;
* 3. Insert Records into the tables
  + USE contactsWebsite;

insert into Users (FirstName,LastName, Username,Password) VALUES ('Rick','Leinecker','RickL','COP4331');

insert into Users (FirstName,LastName, Username,Password) VALUES ('Sam','Hill','SamH','Test');

insert into Users (FirstName,LastName, Username,Password) VALUES ('Rick','Leinecker','RickL','5832a71366768098cceb7095efb774f2');

insert into Users (FirstName,LastName, Username,Password) VALUES ('Sam','Hill','SamH','0cbc6611f5540bd0809a388dc95a615b');

insert into Contacts (FirstName,LastName, Phone, Email, UserID) VALUES ('Emma','Lewis','38024884390489','emma.lewis@gmail.com', 1);

insert into Contacts (FirstName,LastName, Phone, Email, UserID) VALUES ('Emma','Lewis','38024884390489','emma.lewis@gmail.com', 3);

insert into Contacts (FirstName,LastName, Phone, Email, UserID) VALUES ('Lisa','Williams','38024884390489','lisa.williams@gmail.com', 2);

insert into Contacts (FirstName,LastName, Phone, Email, UserID) VALUES ('Lisa','Williams','38024884390489','lisa.williams@gmail.com', 4);

* 4. Create a User
  + Use contactsWebsite;

create user 'TheRandom' identified by 'WeLoveCOP4331';

* + Now we need to grant permissions to the database for that user:

grant all privileges on contactsWebsite.\* to 'TheRandom'@'%';

* 5. Testing

Select \* from Users;

Select \* from Contacts;

* + Also:

Select \* from Contacts where UserID=1;

Select \* from Contacts where UserID=4;

* 6. Demonstration of Working Database:

Login.php

When a user tries to log in, it checks if the username and password given are in the User table.

If not, return an invalid user and prompts them to enter a new username and password.

If it is, changes to dashboard webpage.

Register.php

Where a new user registers.

User must type in a Username, Password, First name, Last name. This information is made into a new record and inserted into the User table.

Dashboard.php

Has a navigation side bar.

* Creation webpage
* Deletion webpage
* Update webpage

Has a search bar (for searching contact names).

* The Contacts table is accessed and the first name, last name, and UserID of each record are used.
* When a name or part of a name is searched, if there is a match.
  + returns the matches below the search bar.
* If not matches.
  + Shows the message “no result.”
* Each matching contact shows the contact’s information, and a button for updating and deletion. (This idea is subject to change)
  + When the update button is pressed:
    - Text area appears
    - User can input a new first name, last name, phone, and email address.
    - When they are done, they must press a button to update.
    - The record for that contact in the Contacts table is changed.
    - Changes are made to the website.
  + When the delete button is pressed:
    - Contact is removed from matching search list.
    - Contact record is removed from the Contact table.

Check each .php after making with postman.

Potential Design for Website (Subject to Change):