Covid Contacts

Team 9

Mark Brogan, Alec Tang

CS 157A Section 80

Project Fundamentals

This project, and its purpose is to utilize and develop a clean, simple web application to be used by the general public and doctors alike, *“Covid Contacts”* brings together users by allowing connection to a doctor in the user-base, giving access to quick diagnostics and over-the-web chatting for tips and tricks to helping us all stay healthy.

The Goal

With the pandemic, in its growing infection rate, Covid Contacts wishes to alleviate public stress and unknowing of hotspots and true, factual data. People across the globe can connect and talk with others about the state of themselves. Users will also be able to search articles and statistics with a location basis to see how their area is doing.

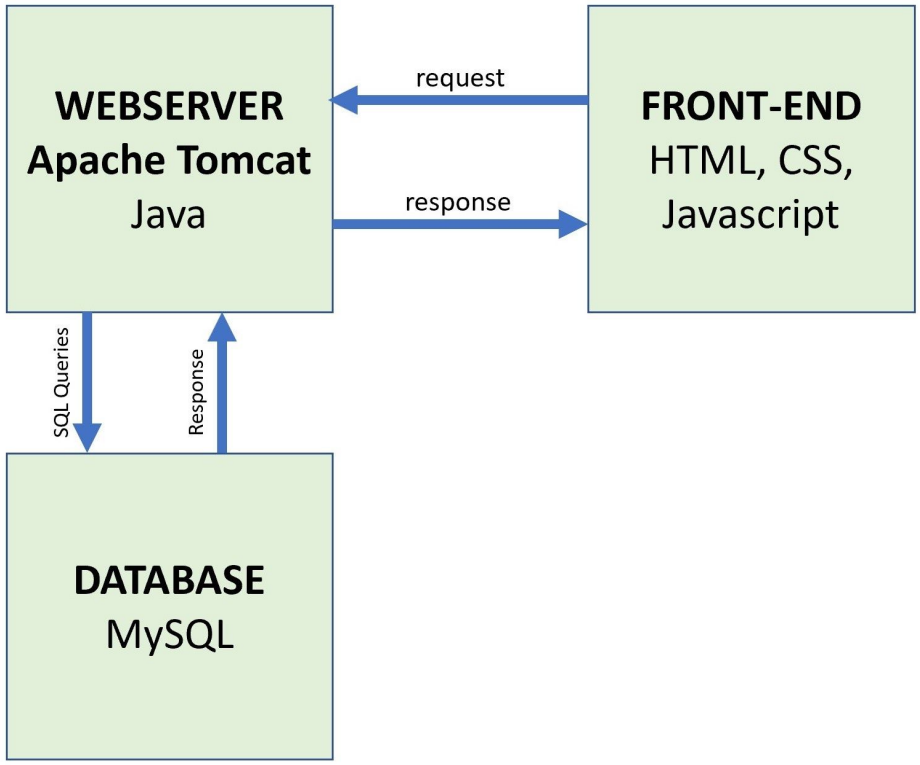
Most social media hubs do not allow for this communication between members and doctors, with access to non-convoluted data that is filled with extraneous information. Articles posted and allowed into the database will be screened for validity and factual data.

Doctors will be able to post helpful links to the ‘Stories’ area for users to reference on best practices on how to stay safe during the foreseeable future.

Join Forces, Team Up Against Covid-19

Covid Contacts, with strict enforcement of factual data, will be the leading information hub for the public to get data at the click of a button. Join up *virtually* against Covid-19 with certified doctors that have your best interests at heart. Fight against the spread with *Covid Contacts.*

Systems and Requirements



Front-End

Users will be able to use Covid Contacts with intuitive HTML and CSS. Our team hand crafted the UI for functionality and useability for all users.

Web Server

Covid Contacts connects the database and user-facing information by Apache Tomcat serverletts. This handles server requests to the database from user and back-end with Java code,

Database

With using MySQL Community Server as our RDBMS to manage all the data being queried and altered.

Functional Requirements

Sign-Up

- The user shall be able to sign up for Covid Contact using email and password credentials.
- The system shall update the database based on the user's information.

Sign-In

- The user shall be able to sign in to Covid-Contact using email and password credentials.
- If the user does not input their credentials correctly, they will be prompted to try signing in again.
- The system shall update the database based on the user's information.

Password Update

- When the user clicks the Password Update link in the user webpage, a panel will appear asking for their email address.
- The user will also have to input their new password and it will be updated in the database.

Doctor-Patient Matching

- Doctors will be assigned to a User upon creation of a new appointment.
- Users can also contact certain doctors to manually match with.
- The system will update the database on the appointment creation.

Making Appointment

- The user shall have the ability to make an appointment with their doctor.
- On the appointment screen, the user will be able to type in a brief description about their symptoms and concerns and choose the available slot on their doctor's schedule for the appointment.

- The system shall update the database based on the user action.

Display Doctor Contact

- The user will be able to view their doctor contact by inputting their credentials.
- The display will show the doctor's name, email address, and personal phone number

Display Appointment

- The user can view their appointments when they input their credentials.
- The display will show the doctor that was chosen for their appointment, the hospital name, symptoms of the user, and date slot that the user desired.

Display Covid Source

- The user shall be able to view news available on the database. The user will be able to search new sources by category. The categories are stories, stats, and safety.
- When the user searches, the user can view the news title, description, and a url.

Add Friend

- The user shall be able to add another user to their friend list. The user will have to input the phone number and email of the friend in order to make the relationship with the friend

Display Friend's List

- The user shall be able to view their friends when they input their credentials.
- Displayed will be their friends name, phone number, and email address.

Add Group Chat

- The user will be able to input a new group chat by inputting their credentials and their choice of group name.
- The user will also become the owner of the group

Add Group Members

- The user will also be able to input new members into a groupchat.

- The user will input a new group member with the group members credentials and the group name that the member will be added to. This will also create a connection between the user and the member being inputted.

Display Group Chat

- The user will be able to view a certain group chat based on the chat's name. The user will input the group chat credentials and then the user will be able to view the members of the group chat.
- The display will show the group name, names of the members, and the members phone number and email address.

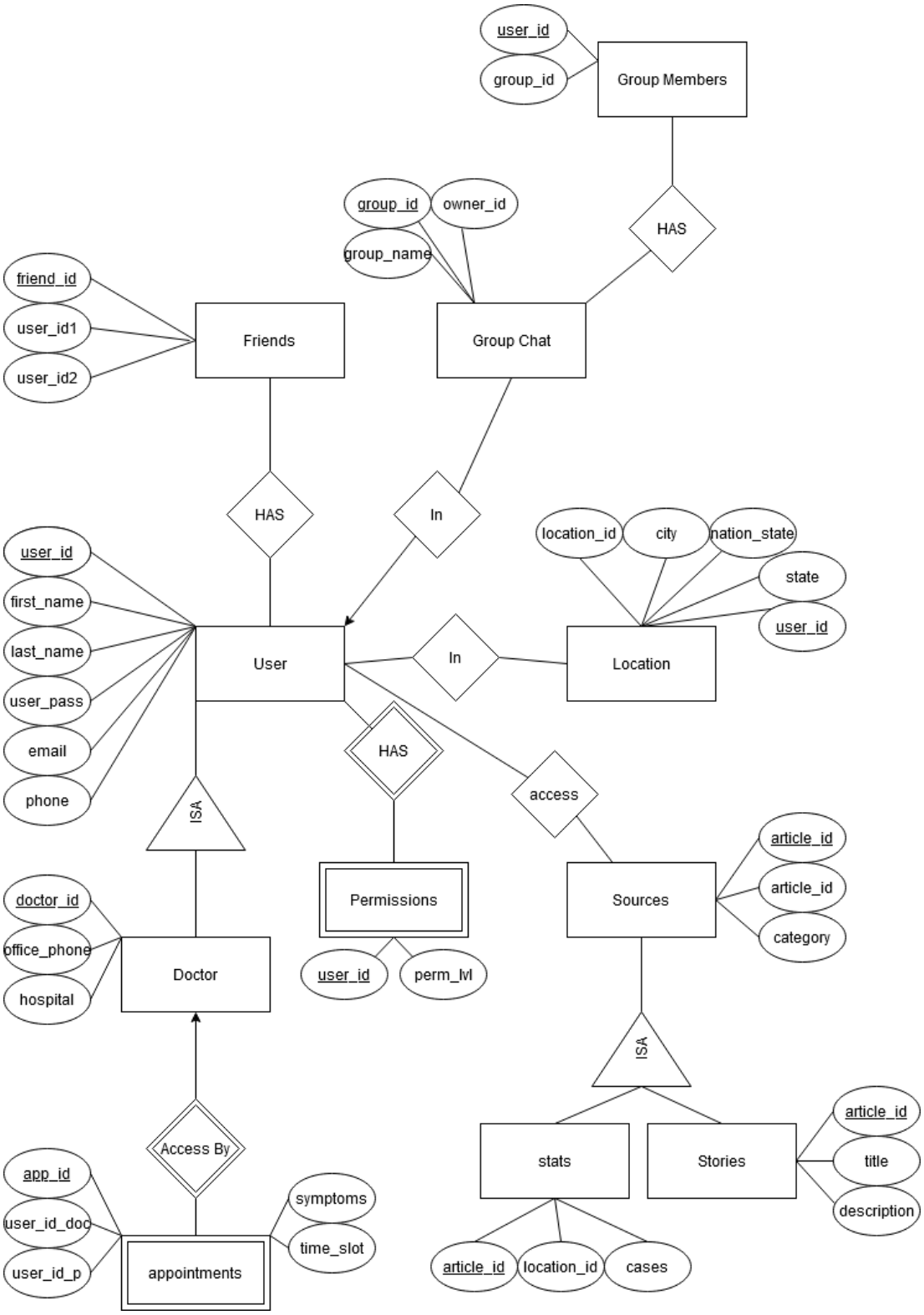
Delete Group Chat

- The user will be able to delete a group chat by inputting the group chat credentials.
- The database will also be updated to not have the group chat anymore.

Functional Process

Covid Contacts will allow users to access Covid information can connect with their peers. Users will sign up with their name and password. Users will also input their contact information(phone number and email) so they will be able to connect with their friends and doctors. Once the user has signed up, each user in each category will have access to the application and its basic functions. Users and consumers will be given limited access to logs and data provided by the statistics they wish to view. The data that can be accessed will be news stories and statistics of Covid. The user can connect with other users as friends. Users will be able to also create group chats to connect with other users. Users will also be able to connect with doctors as well so the user can easily access their doctor if a user has concerns about medical issues and Covid.

E/R Diagram


user_id
group_id
Group Members
HAS
group_id
owner_id
group_name
friend_id
user_id1
user_id2
Friends
Group Chat
HAS
In
location_id
city
nation_state
state
user_id
user_id
first_name
last_name
user_pass
email
phone
User
In
Location
HAS
access
ISA
article_id
article_id
category
Permissions
Sources
doctor_id
office_phone
hospital
Doctor
user_id
perm_lvl
ISA
Access By
app_id
user_id_doc
user_id_p
symptoms
time_slot
appointments
stats
Stories
article_id
title
description
article_id
location_id
cases

Explanation

User

User is an entity that represents the person who will be using Covid Contacts. A user will be created when the person signs up with the attributes; name, password, email, and phone. The user will be given a unique id as well.

Friends

A user will be able to add friends to their contact list and will be stored in a friends entity set. This entity set will store the friend relationship between users. Once a friend is added, the user will see their id, name, phone number, and email.

Doctor

The doctor entity set will contain doctors who can be accessed by the user if they have concerns about their health or Covid. A doctor will have a doctor_id to designate a user as a doctor, office phone number, and hospital. This creates a relationship to a user as a doctor.

Location

The location entity set will have the locations of the users and friends. This set will hold the id, city, and state of the users and friends, so the people using Covid Contacts will know where each other are located.

Group

Group entity set will have the group chats that the user and friends are in. Each group chat entity will have a unique group number, a group name, and the names of the people in the group

UserHasFriends

UserHasFriends is a relationship set that connects a user with the friends entity set. A user will be able to connect with multiple friends.

UserInGroup

UserInGroup is a relationship set that connects the users within the group entity set. There will only be one user within a group entity.

GroupChatHasGroupMembers

GroupChatHasGroupMembers is a relationship set that connects the user and friends within the group entity set. There can be multiple users within a group.

UserInLocation

UserInLoaction is a relationship set that connects users with a specific location. A user will only have one location assigned to them.

AppointmentRecords

AppointmentRecords will keep track of the appointments the user has with their doctor. An appointment will be identified by an appointment number, the user's symptoms, time slot, and date of the appointment.

AppointmentRecordsHasDoctor

AppointmentRecordsHasDoctor will be connected to a doctor to help identify the appointment identification. An appointment record will have one or more doctors.

CovidSources

CovidSources entity set will contain all of the news sources that have covid news stories and statistics. Covid sources will be identified by the news source name and the category it is under.

UserAccessCovidSources

UserAccessCovidSources is the relationship set that connects users with Covid sources so they can share information to their friends. A user will have access to multiple covid sources.

Stats

Stats is a subclass of the covid sources that will contain the cases of covid in each area. Each entity will have the number of cases, city, and state so users will know covid stats in their area.

Stories

Stories is a subclass of covid sources that will contain any news headlines of covid. Users will be able to share this information to their friends. Stories will have a headline name, source, and description of new story.

Permissions

Permissions will link a permissions number to users which will help identify who is a regular user or a doctor user. This table will include the users_id and a permissions number - 1 for regular user and two for a doctor.

Covid Contacts Database Schema

Entity Sets

1. User(user_id, first_name, last_name, user_password, email, phone_number)

2. Friend(friend_id, user_id1, user_id2)
3. Group_Chats(group_id, group_name, owner_id)
4. Group_Members(user_id, group_id)
5. Doctor(doctor_id, office_phone, hospital)
6. Location(location_id, city, nation_state, user_id)
7. Appointments(appt_id, user_id_doc, user_id_p, symptoms, time_slot)
8. Sources(article_id, article_url, category)
9. Stats(article_id, location_id, cases)
10. Stories(article_id, title, description)
11. Permissions(user_id, permissions)

Tables

Users

```
mysql> SELECT * FROM covid_contacts.user;
+---------+------------+-----------+----------------------+---------------------------+------------+
| user_id | first_name | last_name | user_pass            | email                     | phone      |
+---------+------------+-----------+----------------------+---------------------------+------------+
|       1 | Donna      | Rodriguez | DonnaRodriguez1105   | DonnaRodriguez@sjsu.edu   | 4084044711 |
|       2 | Bruce      | Scott     | BruceScott426        | BruceScott@sjsu.edu       | 4084046261 |
|       3 | Willie     | Faires    | WillieFaires2482     | WillieFaires@sjsu.edu     | 4084044180 |
|       4 | David      | Reed      | DavidReed759         | DavidReed@sjsu.edu        | 4084045065 |
|       5 | Miguel     | Kelly     | MiguelKelly2409      | MiguelKelly@sjsu.edu      | 4084049797 |
|       6 | Elizabeth  | Walker    | ElizabethWalker114   | ElizabethWalker@sjsu.edu  | 4084043273 |
|       7 | Cathy      | Hines     | CathyHines1818       | CathyHines@sjsu.edu       | 4084048870 |
|       8 | Leta       | Elam      | LetaElam691          | LetaElam@sjsu.edu         | 4084043774 |
|       9 | Sean       | Miller    | SeanMiller764        | SeanMiller@sjsu.edu       | 4084047970 |
|      10 | Danny      | Segarra   | DannySegarra1144     | DannySegarra@sjsu.edu     | 4084046202 |
|      11 | Paul       | Attridge  | PaulAttridge68       | PaulAttridge@sjsu.edu     | 4084047273 |
|      12 | Damien     | Deaver    | DamienDeaver1545     | DamienDeaver@sjsu.edu     | 4084041250 |
|      13 | Mae        | Mosley    | MaeMosley2478        | MaeMosley@sjsu.edu        | 4084043967 |
|      14 | Rosanna    | Adkins    | RosannaAdkins1097    | RosannaAdkins@sjsu.edu    | 4084048563 |
|      15 | Angelo     | Winstead  | AngeloWinstead2313   | AngeloWinstead@sjsu.edu   | 4084042459 |
|      16 | Ryan       | Jones     | RyanJones2108        | RyanJones@sjsu.edu        | 4084043431 |
|      17 | Patricia   | Young     | PatriciaYoung26      | PatriciaYoung@sjsu.edu    | 4084046243 |
|      18 | Lori       | Humes     | LoriHumes1193        | LoriHumes@sjsu.edu        | 4084047523 |
|      19 | Laurence   | Deardorff | LaurenceDeardorff692 | LaurenceDeardorff@sjsu.edu | 4084045164 |
|      20 | Earl       | Craig     | EarlCraig65          | EarlCraig@sjsu.edu        | 4084047596 |
|      21 | Perry      | Rowan     | PerryRowan2230       | PerryRowan@sjsu.edu       | 4084041852 |
|      22 | Alvin      | Alvarado  | AlvinAlvarado2102    | AlvinAlvarado@sjsu.edu    | 4084045426 |
|      23 | Joshua     | Pew       | JoshuaPew177         | JoshuaPew@sjsu.edu        | 4084045733 |
|      24 | Jennifer   | Kennard   | JenniferKennard2492  | JenniferKennard@sjsu.edu  | 4084041062 |
|      25 | Gerald     | Murphy    | GeraldMurphy1716     | GeraldMurphy@sjsu.edu     | 4084043175 |
```

Friends

```
mysql> SELECT * FROM covid_contacts.friends;
+-----------+----------+----------+
| friend_id | user_id1 | user_id2 |
+-----------+----------+----------+
|       100 |        7 |       74 |
|       101 |        8 |       20 |
|       102 |       36 |       49 |
|       103 |       26 |        2 |
|       104 |       48 |        4 |
|       105 |       31 |       40 |
|       106 |       30 |        6 |
|       107 |        3 |       54 |
|       108 |       50 |       70 |
|       109 |       45 |       31 |
|       110 |       24 |       45 |
|       111 |       65 |       57 |
|       112 |       61 |        2 |
|       113 |       74 |       22 |
|       114 |       57 |       46 |
|       115 |       41 |       56 |
|       116 |       74 |       64 |
|       117 |        6 |       18 |
|       118 |       29 |        9 |
|       119 |       61 |        4 |
|       120 |       74 |        6 |
|       121 |       11 |       18 |
|       122 |       18 |       22 |
|       123 |       26 |       18 |
|       124 |        1 |       62 |
+-----------+----------+----------+
25 rows in set (0.00 sec)
```

Group Chat

```
1 •  SELECT * FROM `Covid Contacts`.group_chat;
```

100% 1:1

Result Grid | Filter Rows: Search | Edit: | Export/Import:

group_id	group_name	owner_id
1	Zoom	23
2	Google	18
3	MySQL	31
4	SJSU	33
5	Good Times	4
6	Family Chat	47
7	Group Name #4	6
8	Programmers	34
9	3-tier	33
10	team-9	22
11	minty	14
12	Covid Contacts	90
13	Team13	5
14	Apple	18
15	Car Lovers	9

Result Grid | Form Editor | Field Types | Query Stats

Group Members

```
mysql> SELECT * FROM covid_contacts.group_members;
+---------+----------+
| user_id | group_id |
+---------+----------+
|       1 |        7 |
|       2 |        2 |
|       3 |        6 |
|       4 |        5 |
|       5 |        2 |
|       6 |        5 |
|       7 |       10 |
|       8 |        1 |
|       9 |        7 |
|      10 |        9 |
|      11 |        5 |
|      12 |        9 |
|      13 |        1 |
|      14 |        7 |
|      15 |       10 |
|      16 |        3 |
|      17 |        4 |
|      18 |        1 |
|      19 |        8 |
|      20 |       11 |
|      21 |        5 |
|      22 |        5 |
|      23 |        1 |
|      24 |       11 |
|      25 |        7 |
```

Doctor

```
mysql> SELECT * FROM covid_contacts.doctor;
+-----------+--------------+-----------------+
| doctor_id | office_phone | hospital        |
+-----------+--------------+-----------------+
|         1 |   5104039878 | Pepegas Unitied |
|         2 |   5104039730 | Yellow Blade    |
|         3 |   5104039468 | Kaiser          |
|         4 |   5104033756 | Pepegas Unitied |
|         5 |   5104039586 | Yellow Blade    |
|         6 |   5104031515 | Red Cross       |
|         7 |   5104035658 | Kaiser          |
|         8 |   5104032479 | Red Cross       |
|         9 |   5104032043 | Kaiser          |
|        10 |   5104031828 | Pepegas Unitied |
|        11 |   5104032745 | Kaiser          |
|        12 |   5104033612 | Red Cross       |
|        13 |   5104035059 | Kaiser          |
|        14 |   5104036410 | Pepegas Unitied |
|        15 |   5104033981 | Red Cross       |
|        16 |   5104034849 | Yellow Blade    |
|        17 |   5104039036 | Red Cross       |
|        18 |   5104031065 | Red Cross       |
|        19 |   5104035127 | Pepegas Unitied |
|        20 |   5104038450 | Blue Shield     |
|        21 |   5104038355 | Red Cross       |
|        22 |   5104036124 | Blue Shield     |
|        23 |   5104032595 | Blue Shield     |
|        24 |   5104038257 | Kaiser          |
|        25 |   5104037118 | Kaiser          |
+-----------+--------------+-----------------+
25 rows in set (0.00 sec)
```

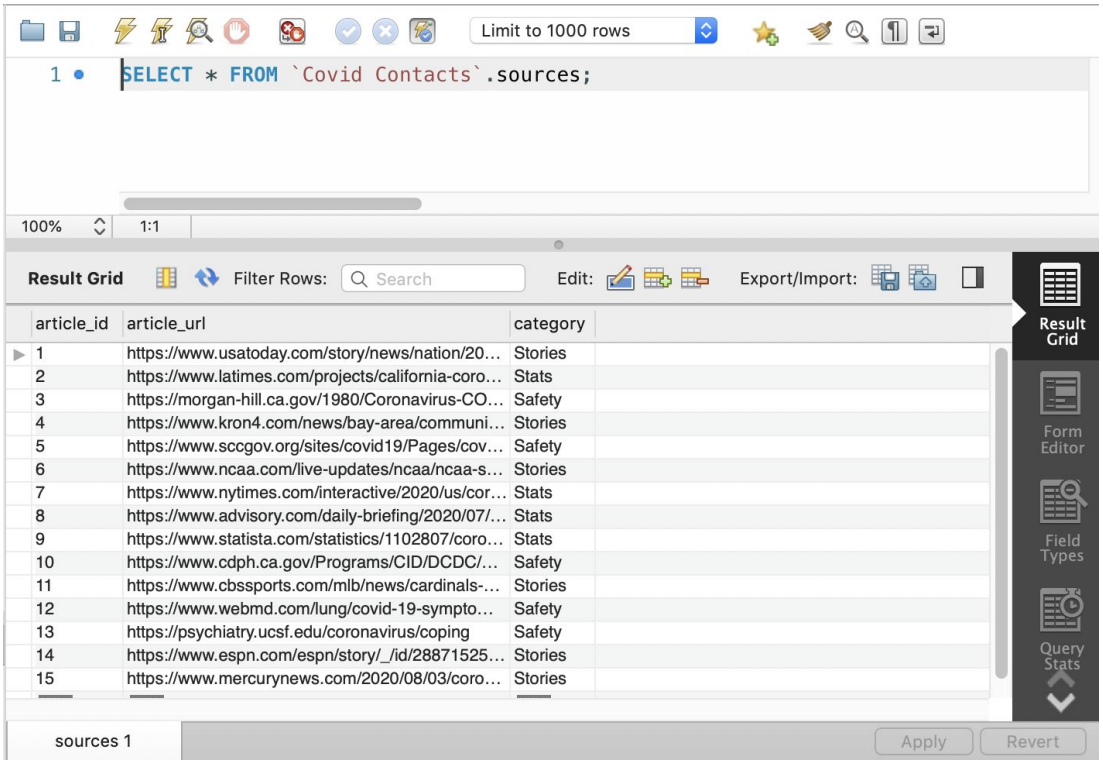
Location

```
mysql> SELECT * FROM covid_contacts.location;
+-------------+---------------+--------------+------------+---------+
| location_id | city          | nation_state | state      | user_id |
+-------------+---------------+--------------+------------+---------+
|           5 | Santa Cruz    | USA          | California |       1 |
|           6 | Tahoe         | USA          | California |       2 |
|           3 | San Francisco | USA          | Califonia  |       3 |
|           2 | Santa Clara   | USA          | California |       4 |
|           2 | Santa Clara   | USA          | California |       5 |
|           4 | Fremont       | USA          | California |       6 |
|           1 | San Jose      | USA          | California |       7 |
|           1 | San Jose      | USA          | California |       8 |
|           6 | Tahoe         | USA          | California |       9 |
|           3 | San Francisco | USA          | Califonia  |      10 |
|           4 | Fremont       | USA          | California |      11 |
|           3 | San Francisco | USA          | Califonia  |      12 |
|           5 | Santa Cruz    | USA          | California |      13 |
|           6 | Tahoe         | USA          | California |      14 |
|           1 | San Jose      | USA          | California |      15 |
|           5 | Santa Cruz    | USA          | California |      16 |
|           3 | San Francisco | USA          | Califonia  |      17 |
|           3 | San Francisco | USA          | Califonia  |      18 |
|           2 | Santa Clara   | USA          | California |      19 |
|           1 | San Jose      | USA          | California |      20 |
|           3 | San Francisco | USA          | Califonia  |      21 |
|           2 | Santa Clara   | USA          | California |      22 |
|           1 | San Jose      | USA          | California |      23 |
|           6 | Tahoe         | USA          | California |      24 |
|           4 | Fremont       | USA          | California |      25 |
```

Appointments

```
mysql> SELECT * FROM covid_contacts.appointments;
+----------------+-------------+-----------+----------------------+------------+
| appointment_id | user_id_doc | user_id_p | symptoms             | time_slot  |
+----------------+-------------+-----------+----------------------+------------+
|              1 |          17 |        48 | sinus infection      | 9/1/2020   |
|              2 |           4 |        73 | cough                | 12/19/2020 |
|              3 |           7 |        49 | yellow fever         | 12/17/2020 |
|              4 |          15 |        71 | unknown              | 11/18/2020 |
|              5 |          14 |        66 | depression           | 10/28/2020 |
|              6 |          13 |        53 | broken bone          | 10/5/2020  |
|              7 |          14 |        32 | Common Cold          | 10/30/2020 |
|              8 |           2 |        75 | sinus infection      | 9/5/2020   |
|              9 |          20 |        70 | erectile dysfunction | 10/20/2020 |
|             10 |           4 |        28 | depression           | 8/17/2020  |
|             11 |           7 |        50 | cough                | 10/17/2020 |
|             12 |          18 |        50 | hypoglycimea         | 11/5/2020  |
|             13 |          10 |        66 | Common Cold          | 8/17/2020  |
|             14 |          19 |        56 | Fever                | 11/16/2020 |
|             15 |          14 |        62 | heart ache           | 9/11/2020  |
+----------------+-------------+-----------+----------------------+------------+
15 rows in set (0.00 sec)
```

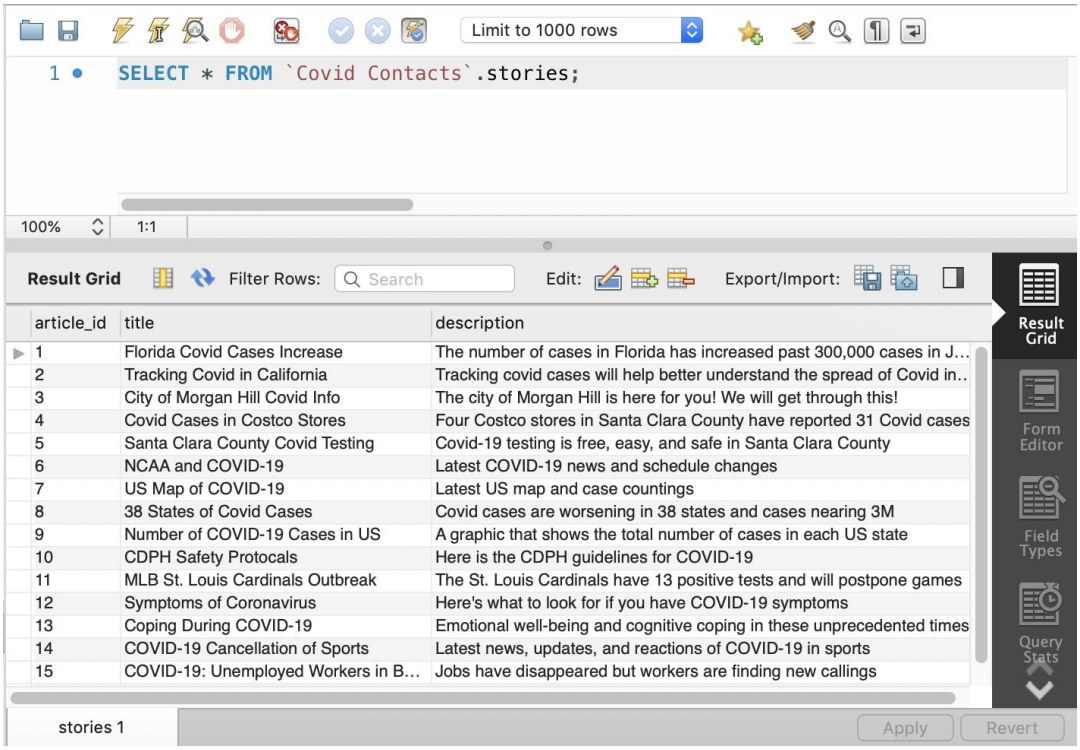
Sources



Stats

```
mysql> SELECT * FROM covid_contacts.stats;
+------------+-------------+-------+
| article_id | location_id | cases |
+------------+-------------+-------+
|          1 |           2 |  1748 |
|          2 |           6 |  6347 |
|          3 |           2 |  1748 |
|          4 |           3 |  4723 |
|          5 |           3 |  4723 |
|          6 |           3 |  4723 |
|          7 |           2 |  1748 |
|          8 |           1 | 11546 |
|          9 |           4 | 12530 |
|         10 |           4 | 12530 |
|         11 |           4 | 12530 |
|         12 |           2 |  1748 |
|         13 |           4 | 12530 |
|         14 |           4 | 12530 |
|         15 |           2 |  1748 |
+------------+-------------+-------+
15 rows in set (0.00 sec)
```

Stories



article_id	title	description
1	Florida Covid Cases Increase	The number of cases in Florida has increased past 300,000 cases in J…
2	Tracking Covid in California	Tracking covid cases will help better understand the spread of Covid in..
3	City of Morgan Hill Covid Info	The city of Morgan Hill is here for you! We will get through this!
4	Covid Cases in Costco Stores	Four Costco stores in Santa Clara County have reported 31 Covid cases
5	Santa Clara County Covid Testing	Covid-19 testing is free, easy, and safe in Santa Clara County
6	NCAA and COVID-19	Latest COVID-19 news and schedule changes
7	US Map of COVID-19	Latest US map and case countings
8	38 States of Covid Cases	Covid cases are worsening in 38 states and cases nearing 3M
9	Number of COVID-19 Cases in US	A graphic that shows the total number of cases in each US state
10	CDPH Safety Protocals	Here is the CDPH guidelines for COVID-19
11	MLB St. Louis Cardinals Outbreak	The St. Louis Cardinals have 13 positive tests and will postpone games
12	Symptoms of Coronavirus	Here's what to look for if you have COVID-19 symptoms
13	Coping During COVID-19	Emotional well-being and cognitive coping in these unprecedented times
14	COVID-19 Cancellation of Sports	Latest news, updates, and reactions of COVID-19 in sports
15	COVID-19: Unemployed Workers in B…	Jobs have disappeared but workers are finding new callings

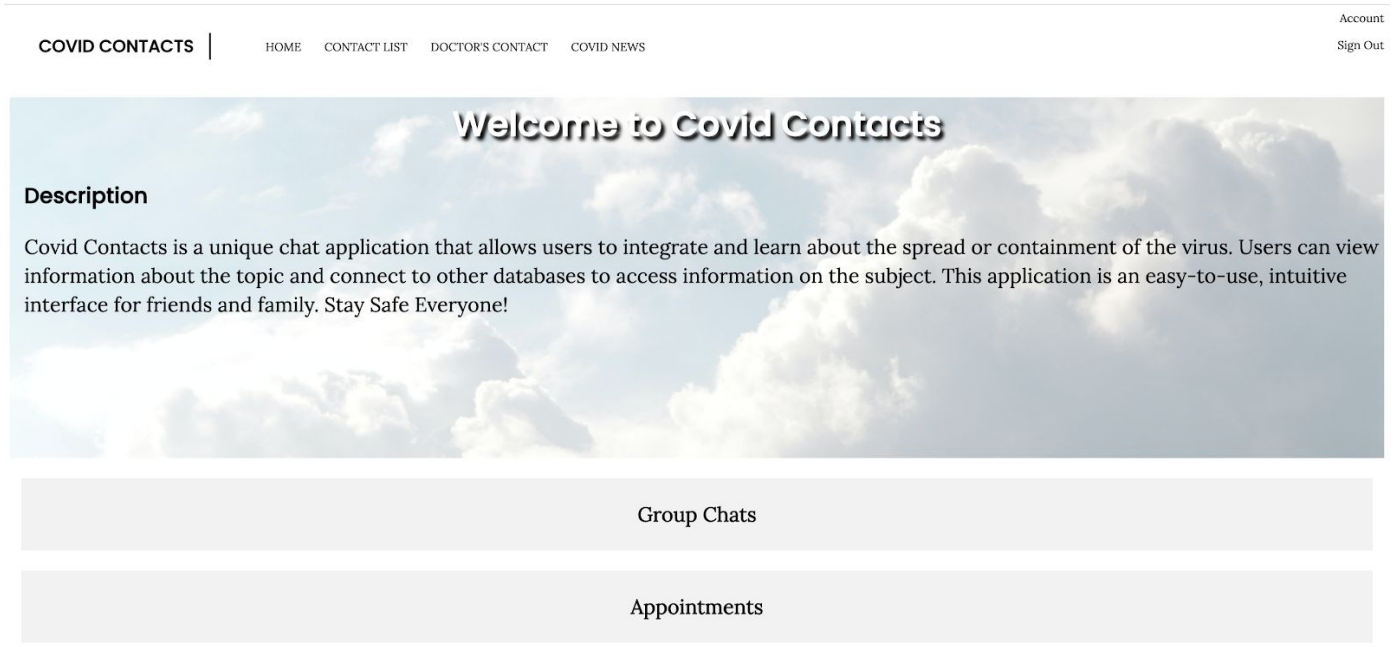
Permissions

```
mysql> SELECT * FROM covid_contacts.permissions;
+---------+------------------+
| user_id | permission_level |
+---------+------------------+
|       1 |                2 |
|       2 |                2 |
|       3 |                2 |
|       4 |                2 |
|       5 |                2 |
|       6 |                2 |
|       7 |                2 |
|       8 |                2 |
|       9 |                2 |
|      10 |                2 |
|      11 |                2 |
|      12 |                2 |
|      13 |                2 |
|      14 |                2 |
|      15 |                2 |
|      16 |                2 |
|      17 |                2 |
|      18 |                2 |
|      19 |                2 |
|      20 |                2 |
|      21 |                2 |
|      22 |                2 |
|      23 |                2 |
|      24 |                2 |
|      25 |                2 |
```

Functional Code and GUI

User Base

Home Page



Landing page upon connecting to the webapp or on successful sign-in or sign-up. On the home page the user will be able to go to any page of our application.

Home Page Front End Code

This is the html code for how the home page was set up. Consists of having the home page have access to all of the applications. Also a quick description of Covid Contacts application

insertGroupName.jsp | redirect.jsp | success.jsp | updatePassword.jsp | index.html

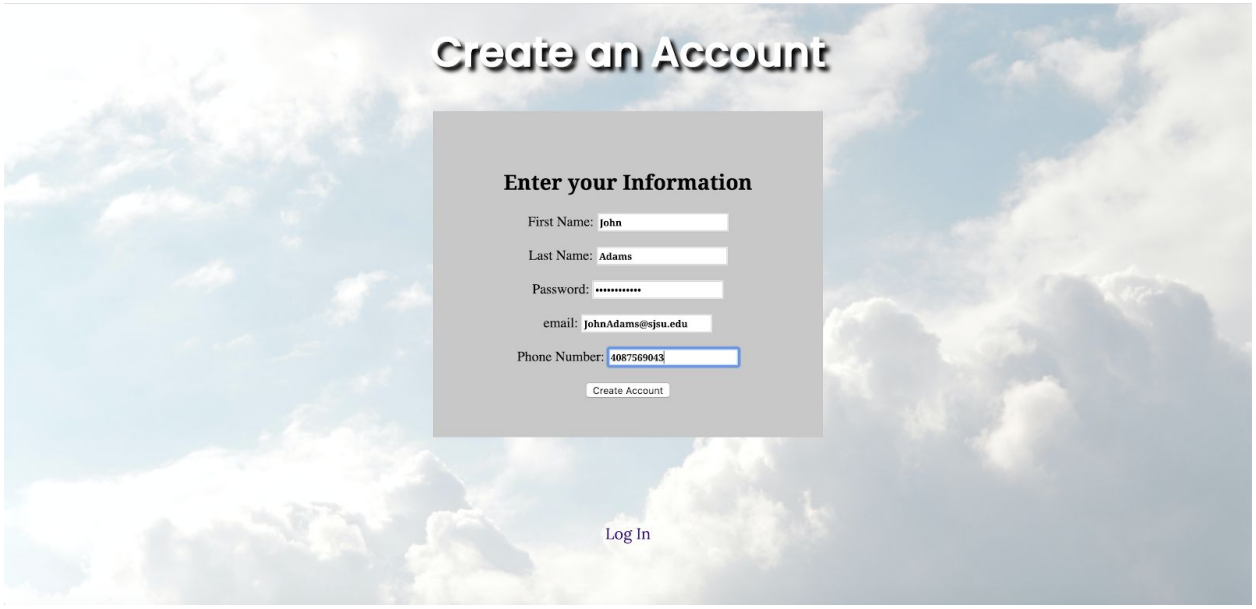
```
<head>
  <title>Covid Contacts</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://fonts.googleapis.com/
  css2?family=Lora&family=Noto+Serif:wght@700&family=Open+Sans+Condensed:wght@300;700&family=Poppins:wght@500&display=swap" rel="stylesheet">
  <link rel = "stylesheet" href = "../css/style.css">
</head>

<body class = "frontpage">
  <header>
    <p class="logo">COVID CONTACTS</p>

    <nav>
      <ul class = "indexlist">
        <li><a href="index.html">Home</a></li>
        <li><a href="friends.html">Contact List</a></li>
        <li><a href="doctor.html">Doctor's Contact</a></li>
        <li><a href="covidnews.html">Covid News</a></li>
      </ul>
      <a href = "user.html" class="user-links">Account</a><br /><br />
      <a href="signInSignUp.html" class="user-links">Sign Out</a>
    </nav>
  </header>
  <main>
    <section class="index-banner">
      <h2 class = "title">Welcome to Covid Contacts</h2>
      <div class = "descr">
      <h3>Description</h3>
      <p>
        Covid Contacts is a unique chat application that allows users to integrate and learn about the spread or containment of the virus. Users can view
        information about the topic and connect to other databases to access information on the subject. This application is an easy-to-use, intuitive interface
        for friends and family. Stay Safe Everyone!
      </p>
      </div>
```

/Library/Tomcat/webapps/test/html/index.html 1:1 LF UTF-8 HTML GitHub Git (0)

```
        for friends and family. Stay Safe Everyone!
      </p>
      </div>
    </section>

    <section class="index-links">
      <a href="groupChat.html"><div class="index-boxlink-1">
        <h3>Group Chats</h3>
      </div>
    </a>
      <a href="appointment.html"><div class="index-boxlink-1">
        <h3>Appointments</h3>
      </div>
    </a>
    </section>
  </main>


</body>

</html>
```

/Library/Tomcat/webapps/test/html/index.html 1:1 LF UTF-8 HTML GitHub Git (0)

Sign-Up



Page for new users to sign-up and be added into the user-base. The user will input their desired credentials in these inputs.

Sign-Up Front End Code

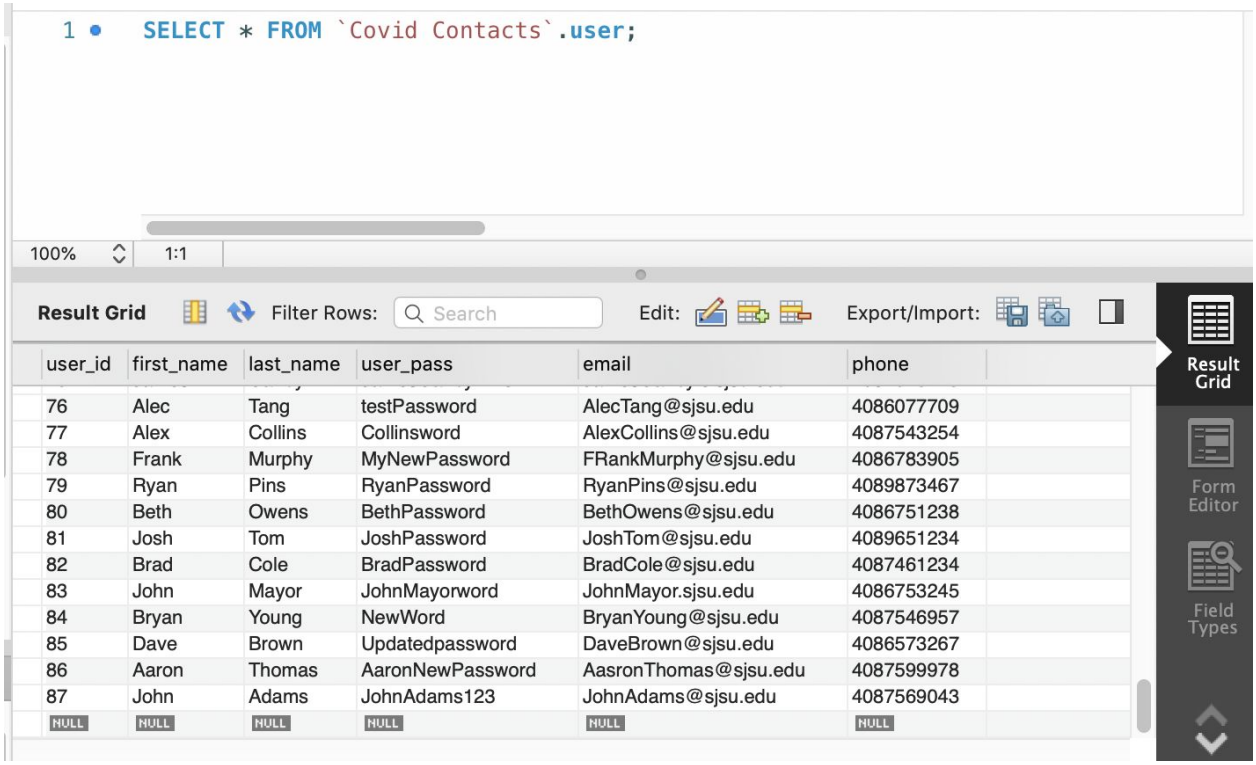
This is the front end code of the set up of the signup page. There are inputs for the new user that will be used to create a new user and a button to finish creating the user.

index.html | signUp.html

```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link href="https://fonts.googleapis.com/
    css2?family=Lora&family=Noto+Serif:wght@700&family=Open+Sans+Condensed:wght@300;700&family=Poppins:wght@500&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="../css/style.css">
    <title>Covid Contacts - SignUp</title>
  </head>
  <body class = "signUpBody">

    <h1 class = "Create-account">Create an Account</h1>
    <div class="form">
      <form class="register" action="../jsp/connect.jsp" method="post">

        <h2>Enter your Information</h2>
        First Name: <input type="text" name="fname" id="fname" /><br/><br/>
        Last Name: <input type="text" name="lname" id="lname" /><br/><br/>
        Password: <input type="password" name="password" id="password" /><br/><br/>
        email: <input type="text" name="email" id="email" /><br/><br/>
        Phone Number: <input type="text" name="phone" id="phone" /><br/><br/>
        <button>Create Account</button>
      </form>

    </div>

    <div class="check">
      <a href="signInSignUp.html">Log In</a>
    </div>

  </body>
</html>

```

/Library/Tomcat/webapps/test/html/signUp.html 1:1 LF UTF-8 HTML GitHub

Table



user_id	first_name	last_name	user_pass	email	phone
76	Alec	Tang	testPassword	AlecTang@sjsu.edu	4086077709
77	Alex	Collins	Collinsword	AlexCollins@sjsu.edu	4087543254
78	Frank	Murphy	MyNewPassword	FRankMurphy@sjsu.edu	4086783905
79	Ryan	Pins	RyanPassword	RyanPins@sjsu.edu	4089873467
80	Beth	Owens	BethPassword	BethOwens@sjsu.edu	4086751238
81	Josh	Tom	JoshPassword	JoshTom@sjsu.edu	4089651234
82	Brad	Cole	BradPassword	BradCole@sjsu.edu	4087461234
83	John	Mayor	JohnMayorword	JohnMayor.sjsu.edu	4086753245
84	Bryan	Young	NewWord	BryanYoung@sjsu.edu	4087546957
85	Dave	Brown	Updatedpassword	DaveBrown@sjsu.edu	4086573267
86	Aaron	Thomas	AaronNewPassword	AasronThomas@sjsu.edu	4087599978
87	John	Adams	JohnAdams123	JohnAdams@sjsu.edu	4087569043
NULL	NULL	NULL	NULL	NULL	NULL

From the SIgn up page, the database was able to retrieve the user signing up and John Adams was inserted into the database.

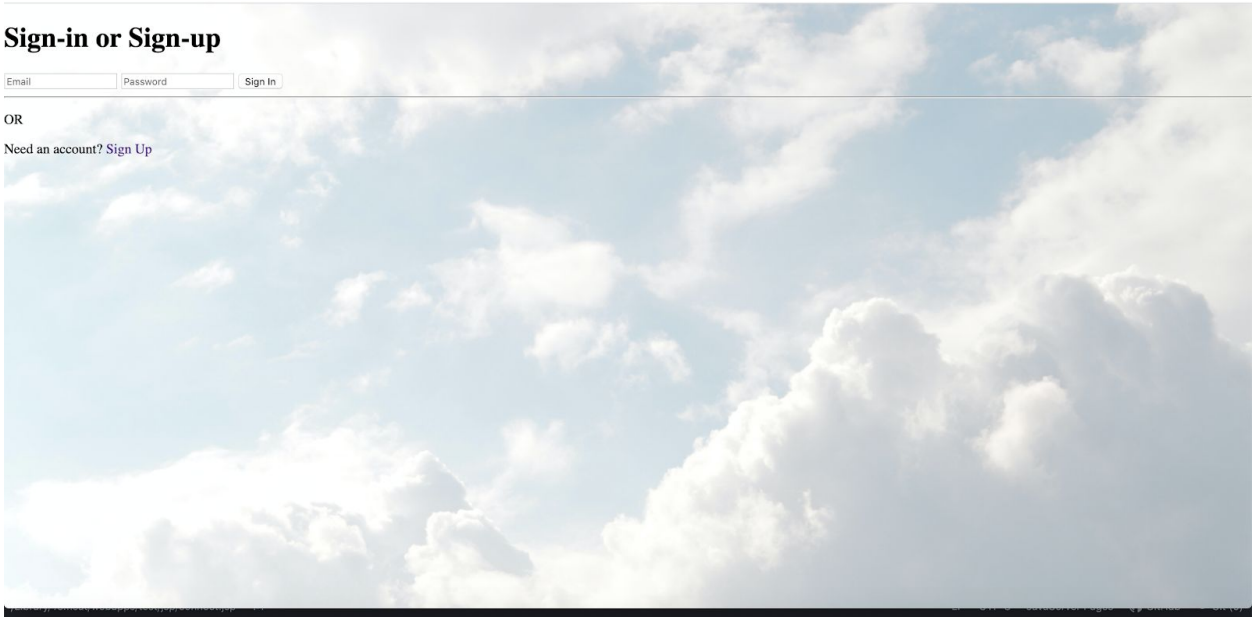
Sign-Up Back End Code

index.html | signUp.html | connect.jsp

```jsp
<%@ page import = "java.sql.*"%>

  <%
    String firstName = request.getParameter("fname");
    String lastName = request.getParameter("lname");
    String password = request.getParameter("password");
    String email = request.getParameter("email");
    String phone = request.getParameter("phone");

    try{
      Class.forName("com.mysql.jdbc.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
      PreparedStatement ps = conn.prepareStatement("insert into user(first_name, last_name, user_pass, email, phone) values(?,?,?,?,?);");
      ps.setString(1,firstName);
      ps.setString(2,lastName);
      ps.setString(3,password);
      ps.setString(4,email);
      ps.setString(5,phone);

      int x = ps.executeUpdate();
      if(x>0)
      {
        out.println("Success");
      }
      else
      {
        out.println("Fail");
      }
    }
    catch(Exception e){
      out.println(e);
    }
  %>

```

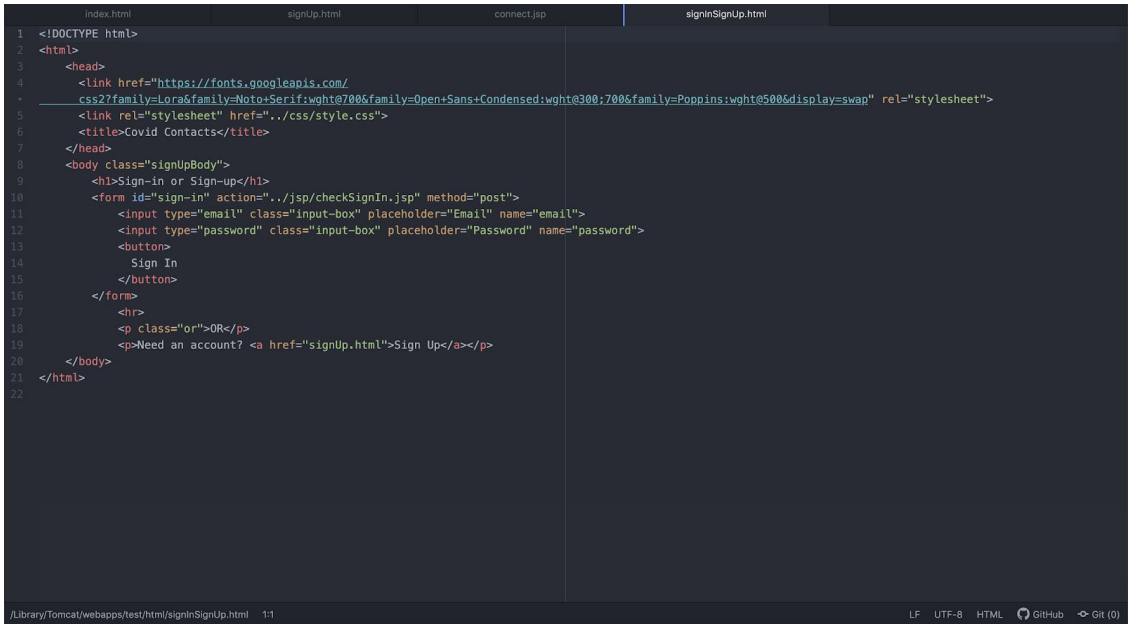
Connect.jsp is linked with the sign up page and, once the user creates a new user, those inputs are able to be taken and entered into the Insert query. If the credentials are correctly inputted, “success” will be displayed to show the query has worked.

Sign-In



Page for existing users to sign-in and connect to Covid Contacts. The user will input their email and password to log in and get to the home page.

Sign-in Front End Code



This is the html for signing in. The user can sign in or click the sign up link to be a user.

Sign-in Back End Code

signInSignUp.html checkSignIn.jsp

```jsp
<%@ page import = "java.sql.*"%>

  <!DOCTYPE html>
  <html lang="en" dir="ltr">
    <head>
      <meta charset="utf-8">
      <title></title>
    </head>
    <body>
      <%! String email, password; %>
      <%
        email = request.getParameter("email");
        password = request.getParameter("password");
        try
        {
          Class.forName("com.mysql.jdbc.Driver");
          Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
          PreparedStatement ps = conn.prepareStatement("SELECT * FROM user WHERE email = ? AND user_pass = ?");

          ps.setString(1,email);
          ps.setString(2,password);

          ResultSet rs = ps.executeQuery();

          boolean check = false;
          while(rs.next())
          {
            out.println("ID: " + rs.getInt(1) + "<br/><br/>");
            out.println("FName: " + rs.getString(2) + "<br/><br/>");
            out.println("\t\t"+ "LName: " +rs.getString(3) + "<br/><br/>");
            out.println("\t\t"+ "Password: " + rs.getString(4) + "<br/><br/>");
            out.println("\t\t"+ "email: " +rs.getString(5) + "<br/><br/>");
            out.println("\t\t"+ "phone: " +rs.getString(6) + "<br/><br/>");
            check = true;
```

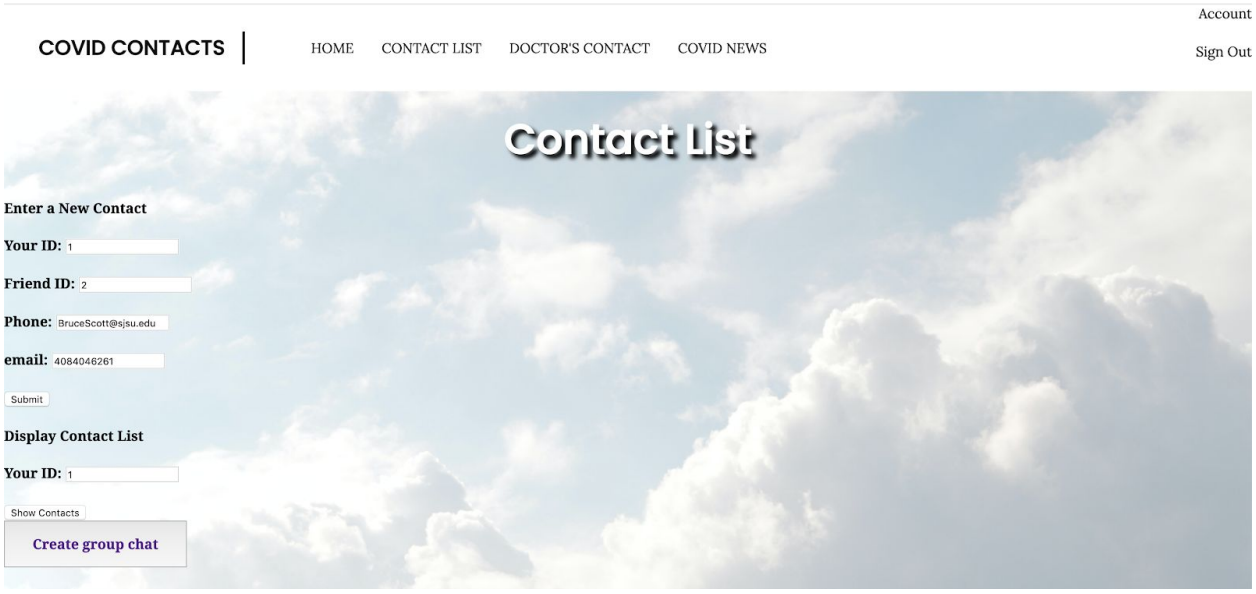
/Library/Tomcat/webapps/test/jsp/checkSignIn.jsp 2:1 LF UTF-8 JavaServer Pages GitHub Git (0)

```jsp
            out.println("\t\t"+ "email: " +rs.getString(5) + "<br/><br/>");
            out.println("\t\t"+ "phone: " +rs.getString(6) + "<br/><br/>");
            check = true;

          }


        if(check == true)
        {
          out.println("Successful login");
          %> <jsp:include page="success.jsp"/> <%
        }
        else
        {
          out.println("Try again");
          %> <jsp:include page="redirect.jsp"/> <%
        }
      }
        catch(Exception e)
        {
          out.println("SQLException caught: " + e.getMessage());
        }
        %>
    </body>
  </html>
```

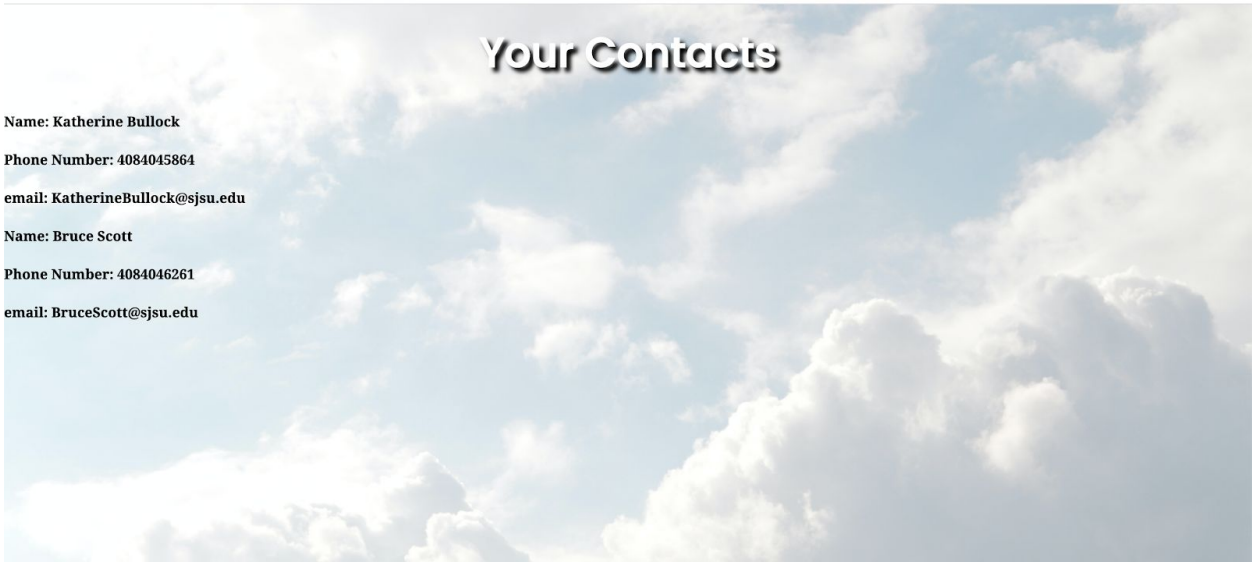
/Library/Tomcat/webapps/test/jsp/checkSignIn.jsp 2:1 LF UTF-8 JavaServer Pages GitHub Git (0)

Once the user has successfully input their credentials, they will be taken to a successful login page so they can also go to the home page. If not then, they will receive a “try again” message because of invalid credentials.

Friends



Area where current users can add/view friends that are also in the user-base. This screen shot also shows user ID 1 connecting with user ID 2 to create a friend relationship. Also user ID 1 also can view friends.



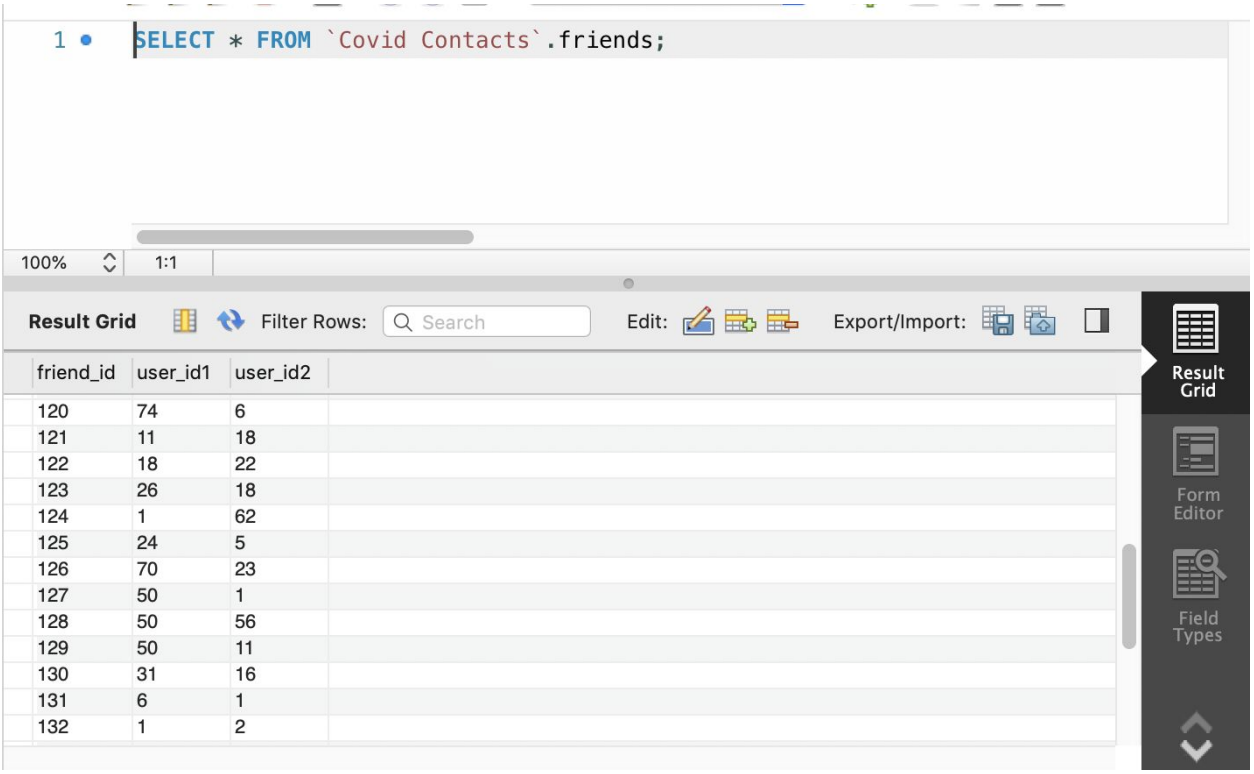
This is the page the user will be taken to once the user clicks show contacts. These are the users that are friends of the user ID 1.

Friends Front End Code

```
        </ul>
        <a href = "user.html" class="user-links">Account</a><br /><br />
        <a href="signInSignUp.html" class="user-links">Sign Out</a>
      </nav>
    </header>

  <h1 class="flist">Contact List</h1>
<div class="form-friends">
    <form class="insertFriend" action="../jsp/insertFriend.jsp" method="post">
      <h4>Enter a New Contact</h4>
      Your ID: <input type="number" name="userID" id="userID" /><br /><br />
      Friend ID: <input type="number" name="friendID" id="friendID" /><br /><br />

      Phone: <input type="text" name="phone" id="phone" /><br /><br />
      email: <input type="text" name="email" id="email" /><br /><br />
      <button>Submit</button>
    </form>


  <form class="display-friends" action="../jsp/displayFriends.jsp" method="post">
    <h4>Display Contact List</h4>
    Your ID: <input type="number" name="userDisplayID" id="userDisplayID" /><br /><br />
    <button>Show Contacts</button>
  </form>



</div>
  <button type="button1" class="groupChat-button">
    <a href="groupChat.html">Create group chat</a>

</body>

</html>

```

This html consists of inputs where the user can input their credentials and friends credentials to add new friend or to display their contact list. Also there is a groupchat button to go to group chat page.

Table



This is the table to store the connections between users as friends. As you can see once the user added a friend, a friend connection was made in the database to connect the user. This connection can be used to help display the user's contact list.

Friends Back End Code

signInSignUp.html | friends.html | displayFriends.jsp

```jsp
<%@ page import = "java.sql.*"%>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
      <link href="https://fonts.googleapis.com/
      css2?family=Lora&family=Noto+Serif:wght@700&family=Open+Sans+Condensed:wght@300;700&family=Poppins:wght@500&display=swap" rel="stylesheet">
      <link rel="stylesheet" href="../css/friendstyle.css">
    <title>Covid Contacts Friends</title>
  </head>

    <body class="groupChatLayout">
        <h1 class=flist>Your Contacts</h1>

    <%
      String userID = request.getParameter("userDisplayID");
      int user = Integer.parseInt(userID);
      try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
        PreparedStatement ps = conn.prepareStatement("SELECT first_name, last_name, phone, email FROM user, friends WHERE user.user_id = friends.user_id2 AND
        friends.user_id1 = ?;");
        ps.setInt(1, user);
        ResultSet rs = ps.executeQuery();
        while(rs.next())
        {
          out.println("Name: " + rs.getString(1) + " " + rs.getString(2) + "<br/><br/>");
          out.println("Phone Number: " + rs.getString(3) + "<br/><br/>");
          out.println("email: " + rs.getString(4) + "<br/><br/>");
        }
      }
      catch(Exception e){
        out.println(e);
      }
    %>
```

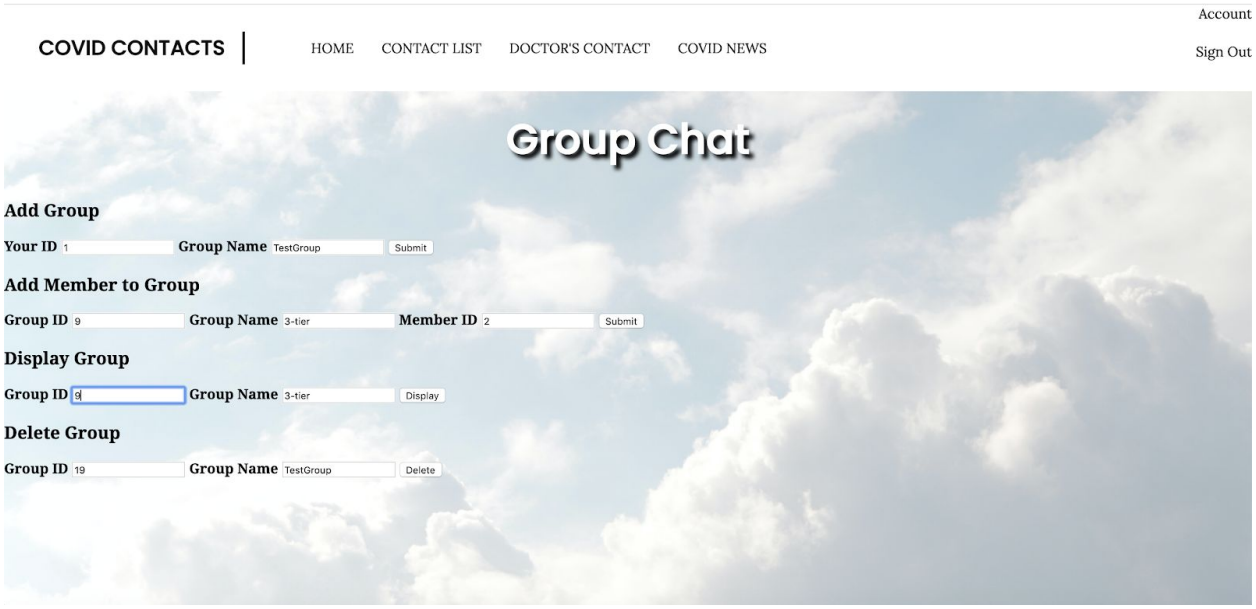
This is the jsp code for displaying the friends of a user. It displays the name, phone number, and email address of their friends.

signInSignUp.html | friends.html | insertFriend.jsp

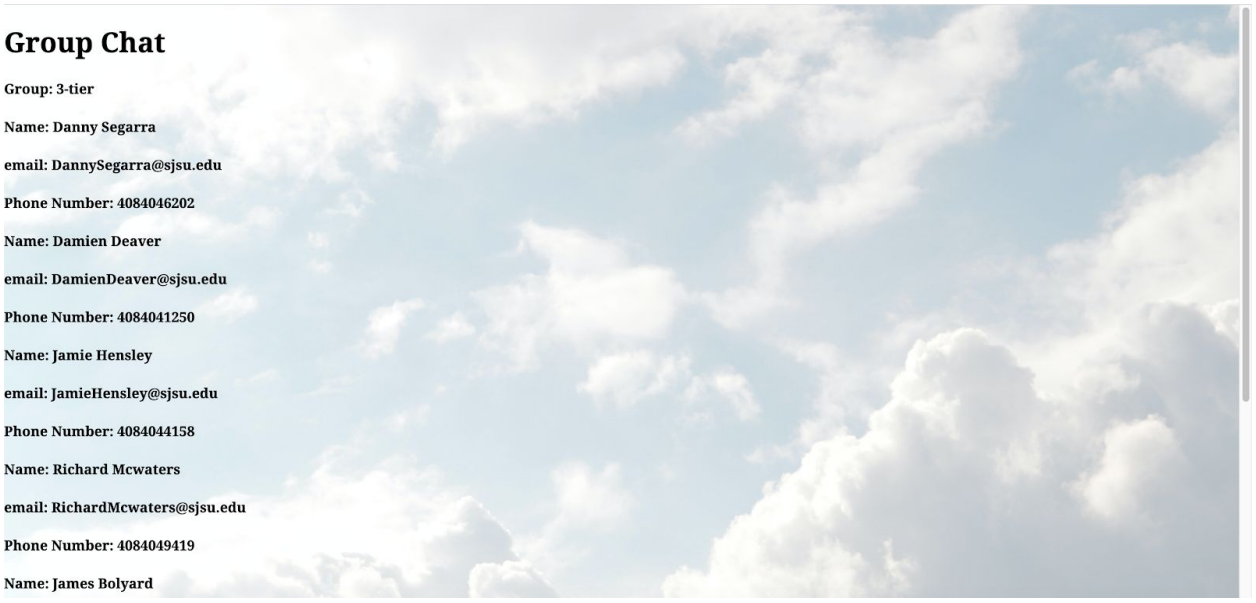
```jsp
  <%
    String userID = request.getParameter("userID");
    String friendID = request.getParameter("friendID");

    int user = Integer.parseInt(userID);
    int friend = Integer.parseInt(friendID);

    try{
      Class.forName("com.mysql.jdbc.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
      PreparedStatement ps = conn.prepareStatement("INSERT INTO friends (user_id1, user_id2)" +
                "VALUES (?,?)");


      ps.setInt(1, user);
      ps.setInt(2, friend);


      int x = ps.executeUpdate();
      if(x>0)
      {
        out.println("Success");
        %> <jsp:include page="success.jsp"/> <%
      }
      else
      {
        out.println("Fail");
      }
    }
    catch(Exception e){
      out.println(e);
    }
  %>
```

This is the code for creating a friend. This inserts the user and friend's user id into the table to create a new relationship between users The SQL statement creates a new relationship between users.

Groups



Page where users can add/view/delete groups with other users in the user-base. This will not affect friends or users.



This is the page the user will be taken to once they click on display group. For example, these are all of the users in the 3-tier chat

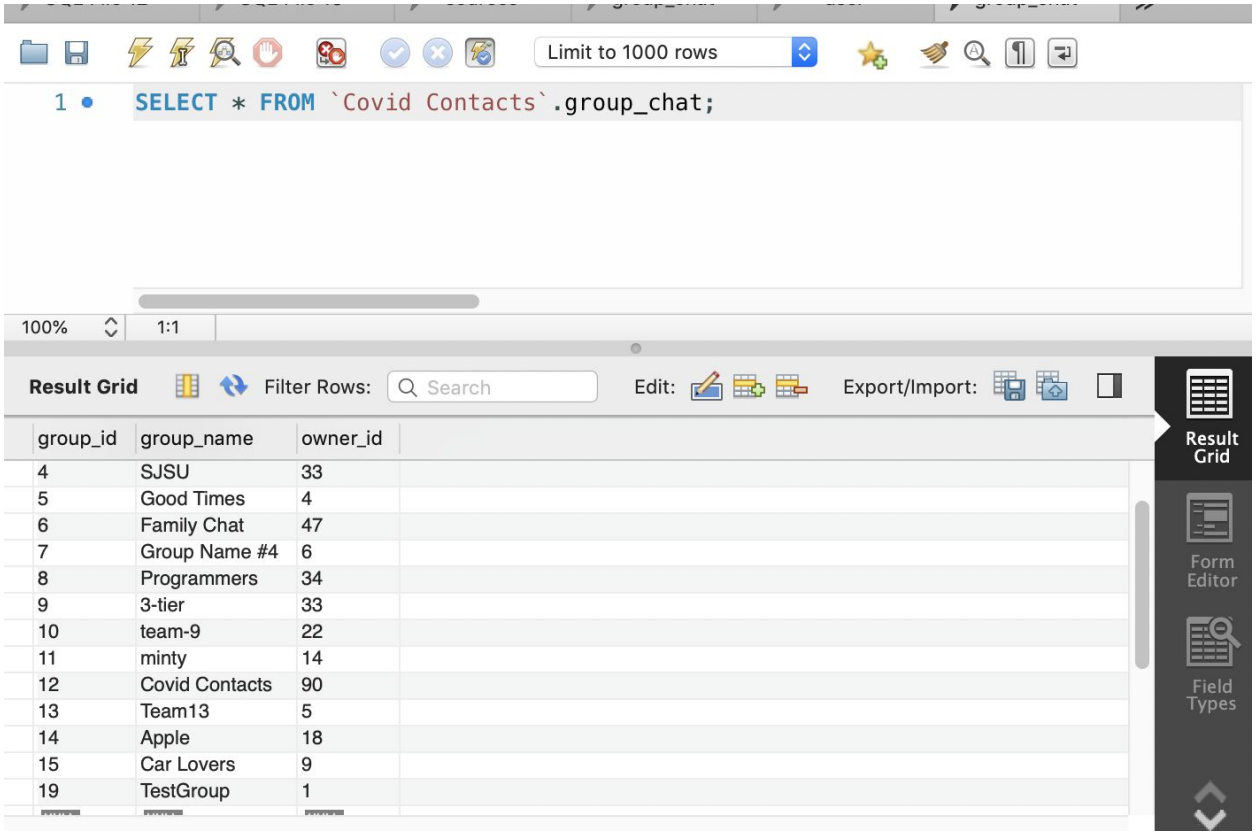
Groups Front End Code

```
    </header>
  <h1 class="groupChatHeaders">Group Chat</h1>


    <form class="insertGroupName" action="../jsp/insertGroupName.jsp" method="post">
      <h2 class="groupTextHeaders">Add Group</h2>

        <label for="ownerID">Your ID<label>
        <input id="ownerIDTextBox" type="text" name="ownerID">

        <label for="newGroup">Group Name<label>
        <input id="newGroupTextBox" type="text" name="newGroup">

        <button id="addGroupButton">Submit</button>
    </form>


    <form class="insertGroupMembers" action="../jsp/insertGroupMembers.jsp" method="post">
      <h2 class="groupTextHeaders">Add Member to Group</h2>

        <label for="groupID">Group ID<label>
        <input id="groupIDTextBox" type="text" name="groupID">

        <label for="groupName">Group Name<label>
        <input id="groupNameTextBox" type="text" name="groupName">

        <label for="newMemberID">Member ID<label>
        <input id="newMemberIDTextBox" type="text" name="newMemberID">



         <button>Submit</button>
    </form>
```

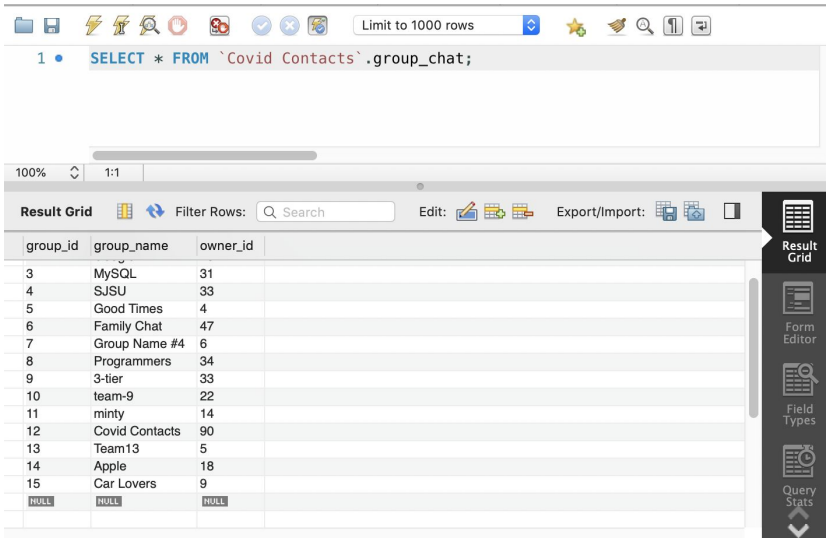
```
        <input id="groupNameTextBox" type="text" name="groupName">

        <label for="newMemberID">Member ID<label>
        <input id="newMemberIDTextBox" type="text" name="newMemberID">



         <button>Submit</button>
    </form>

    <form class="displayGroup" action="../jsp/displayGroup.jsp" method="post">
      <h2 class="groupTextHeaders">Display Group</h2>

        <label for="groupID">Group ID<label>
        <input id="groupIDTextBox" type="text" name="displayGroupID">

        <label for="groupName">Group Name<label>
        <input id="groupNameTextBox" type="text" name="groupName">


         <button>Display</button>
    </form>

    <form class="deleteGroupName" action="../jsp/deleteGroupName.jsp" method="post">
      <h2 class="groupTextHeaders">Delete Group</h2>
      <label for="groupID">Group ID<label>
      <input id="groupIDTextBox" type="text" name="deleteGroupID">
      <label for="groupID">Group Name<label>
      <input id="groupIDTextBox" type="text" name="deleteGroupName">
      <button>Delete</button>

    </form>
```

This is the html for group chat. The user will see options to add a group, add group members to a group, view a group chat, or delete a group.

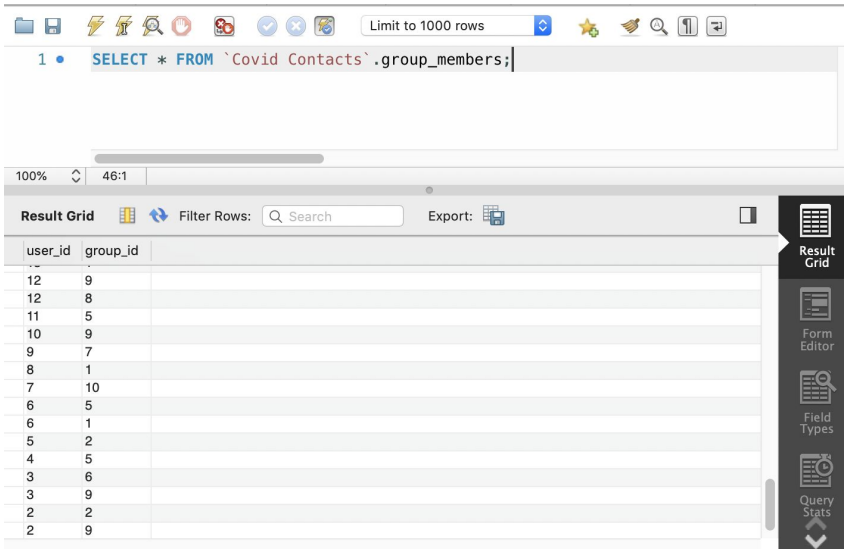
Tables



This is the group_chat table that stores the group chat names and also the owner of the group chat. As you can see once the user inputted a new group name it got inputted into the database.



Also, you can see that, once the user inputted the TestGroup to be deleted, the group name was deleted from the database.



This is group_members table which stores relationship between a user and a group. As you can see, the user added a member to the group and it inserted a new relationship.

Groups Back End Code

```
<%
  String groupID = request.getParameter("groupID");
  int group = Integer.parseInt(groupID);
  String memberID = request.getParameter("newMemberID");
  int member = Integer.parseInt(memberID);



  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("INSERT INTO group_members VALUES(?,?);");
    ps.setInt(1, member);
    ps.setInt(2, group);

    int x = ps.executeUpdate();
    if(x>0)
    {
      %> <jsp:include page="success.jsp"/> <%
      out.println("Success");
    }
    else
    {
      out.println("Fail");
    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code to retrieve the inputs from the browser to insert into the group_members table. It adds a new relationship between a user and a group name.

```jsp
<%
  String userID = request.getParameter("ownerID");
  int user = Integer.parseInt(userID);
  String groupName = request.getParameter("newGroup");

  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("INSERT INTO group_chat (group_name, owner_id)" + "VALUES (?, ?);");
    ps.setString(1,groupName);
    ps.setInt(2,user);

    int x = ps.executeUpdate();
    if(x>0)
    {
      %> <jsp:include page="success.jsp"/> <%
      out.println("Success");
    }
    else
    {
      out.println("Fail");
    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code to retrieve the groupName and the user to insert a new group name and owner into the group_chat table.

signInSignUp.html | friends.html | groupChat.html | displayGroup.jsp | insertGroupName.jsp | insertGroupMembers.jsp

```jsp
  <body class="doctorpage">
    <h1 class="flist">Group Chat</h1>

<%
  String groupID = request.getParameter("displayGroupID");
  int group = Integer.parseInt(groupID);


try{
  Class.forName("com.mysql.jdbc.Driver");
  Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
  PreparedStatement ps = conn.prepareStatement("SELECT group_chat.group_name, user.first_name, user.last_name, user.email, user.phone FROM user, group_members,
  group_chat WHERE group_chat.group_id = group_members.group_id AND group_members.user_id = user.user_id AND group_chat.group_id = ?;");
  ps.setInt(1, group);

  ResultSet rs = ps.executeQuery();

  boolean display = true;

  while(rs.next())
  {
    if(display == true)
    {
      out.println("Group: " + rs.getString(1) + "<br/><br/>");
      display = false;
    }
    out.println("Name: " + rs.getString(2) + " " + rs.getString(3) + "<br/><br/>");
    out.println("email: " +rs.getString(4) + "<br/><br/>");
    out.println("Phone Number: " +rs.getString(5) + "<br/><br/>");

  }
}
catch(Exception e){
  out.println(e);
}
```

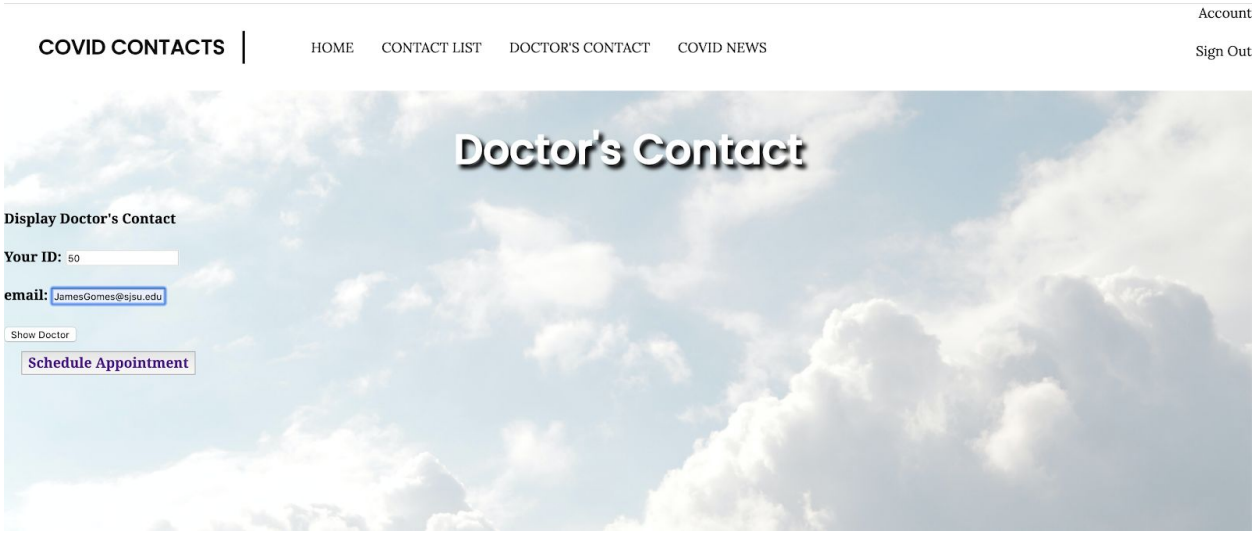
This is the code to retrieve the group that the user wants to view and display the users in the group chat back to the user. This displays the group name once, the group members' names, emails, and phone numbers.

signInSignUp.html | friends.html | groupChat.html | deleteGroupName.jsp | displayGroup.jsp

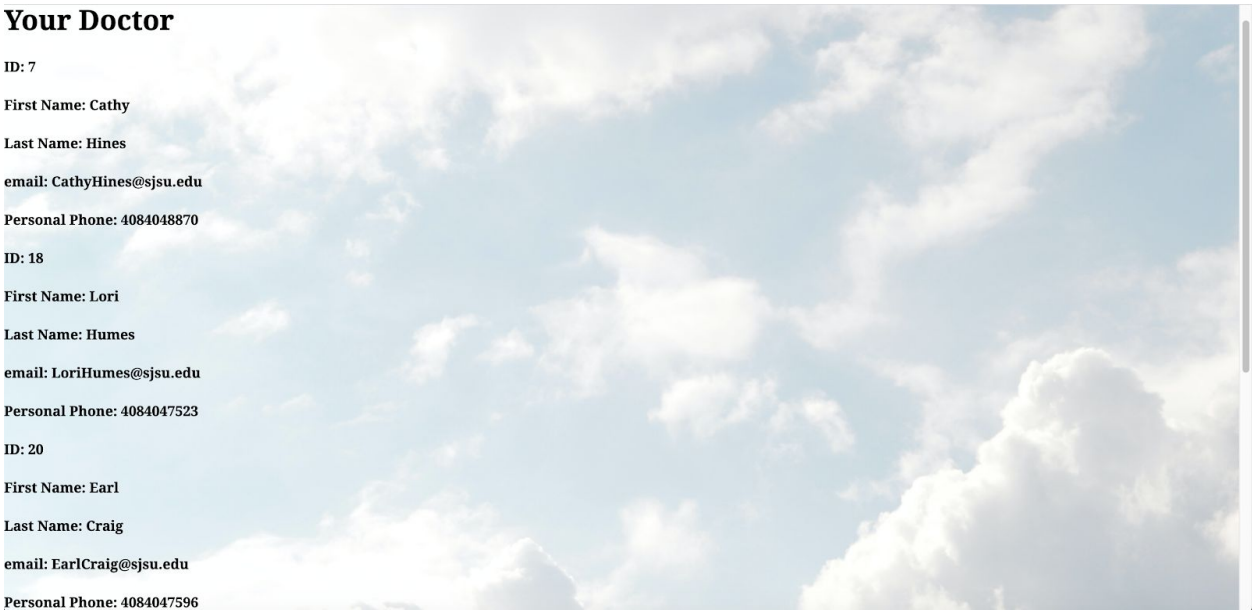
```jsp
<%@ page import = "java.sql.*"%>

<%
  String deleteGroupID = request.getParameter("deleteGroupID");
  int groupID = Integer.parseInt(deleteGroupID);
  String groupName = request.getParameter("deleteGroupName");

  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("DELETE FROM group_chat WHERE group_name = ? AND group_id = ?;");
    ps.setString(1,groupName);
    ps.setInt(2,groupID);

    int x = ps.executeUpdate();
    if(x>0)
    {
      out.println("Success");
    }
    else
    {
      out.println("Fail");
    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code for deleting a group chat. This retrieves the user's inputs of the group id and the group name and the query deletes the entity in the database.

Doctor



Base page where users can view their doctor's information. The user will input their credentials to view their doctor contact. The user can also navigate to the appointment page with the schedule appointment button.



Once the user's credentials are inputted, the user will be able to view their doctor's contacts. The user can see their name, email, and phone number.

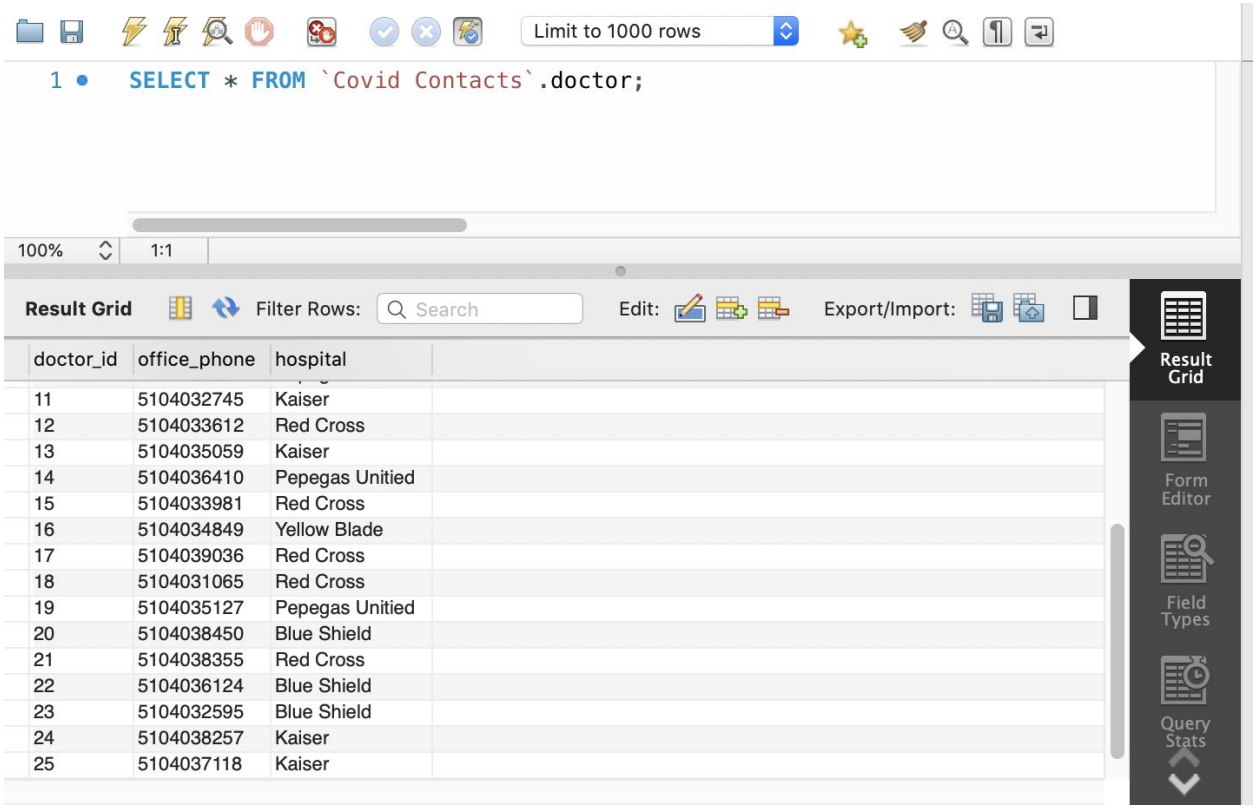
Doctor Front End Code

signInSignUp.html | friends.html | doctor.html

```
</head>
<header>
    <p class="logo">COVID CONTACTS</p>

    <nav>
      <ul class = "indexlist">
        <li><a href="index.html">Home</a></li>
        <li><a href="friends.html">Contact List</a></li>
        <li><a href="doctor.html">Doctor's Contact</a></li>
        <li><a href="covidnews.html">Covid News</a></li>
      </ul>
      <a href = "user.html" class="user-links">Account</a><br /><br />
      <a href="signInSignUp.html" class="user-links">Sign Out</a>
    </nav>
  </header>
  <body class = "doctorpage">
    <h1 class = "doctor">Doctor's Contact</h1>



    <div class="form-doctor">
      <form class="display-doctor" action="../jsp/displayDoctor.jsp" method="post">
        <h4>Display Doctor's Contact</h4>
        Your ID: <input type="number" name="userDisplayID" id="userDisplayID" /><br /><br />
        email: <input type="text" name="useremail" id="useremail" /><br /><br />
        <button>Show Doctor</button>
      </form>
    </div>

    <button type="button2" class="schedule-button">
      <a href="appointment.html">Schedule Appointment</a>
    </button>
  </body>
</html>
```

This is the html for the user to input their credentials into the browser for the database to receive. There is also a schedule appointment button to navigate to the appointments page.

Doctor Table



doctor_id	office_phone	hospital
11	5104032745	Kaiser
12	5104033612	Red Cross
13	5104035059	Kaiser
14	5104036410	Pepegas Unitied
15	5104033981	Red Cross
16	5104034849	Yellow Blade
17	5104039036	Red Cross
18	5104031065	Red Cross
19	5104035127	Pepegas Unitied
20	5104038450	Blue Shield
21	5104038355	Red Cross
22	5104036124	Blue Shield
23	5104032595	Blue Shield
24	5104038257	Kaiser
25	5104037118	Kaiser

This stores the relationship between users to determine who is a doctor. This table and relationship can also be used for displaying appointments and doctor for the user.

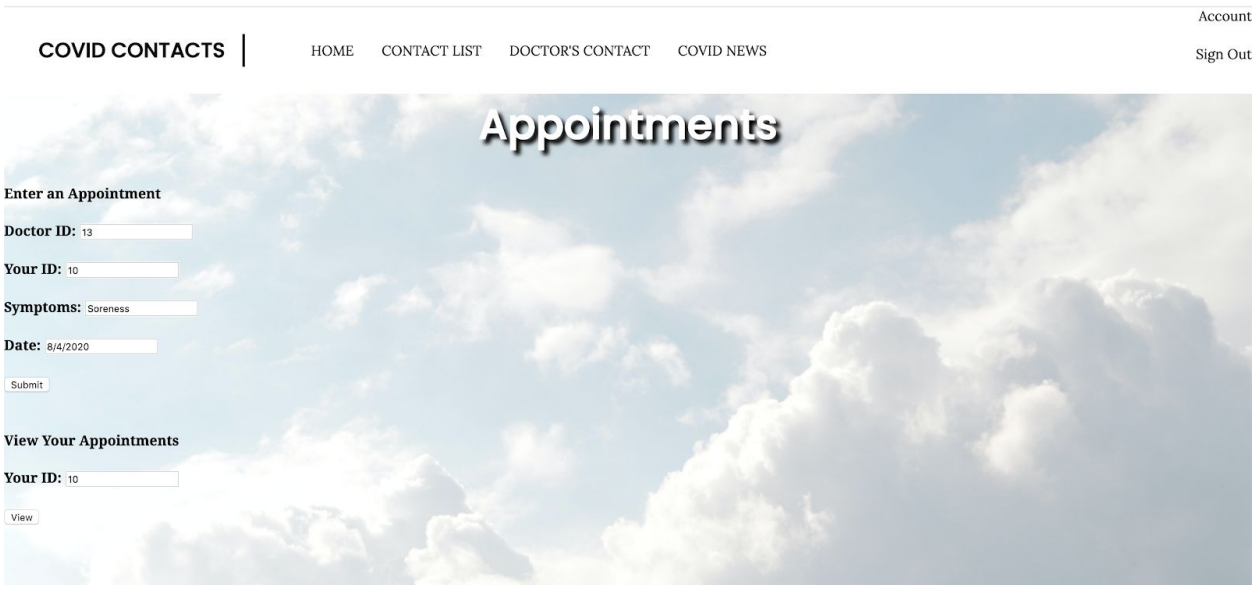
Doctor Back End Code

signInSignUp.html | friends.html | doctor.html | displayDoctor.jsp

```
    <h1 class=flist>Your Doctor</h1>

<%
  String userID = request.getParameter("userDisplayID");

  int user = Integer.parseInt(userID);

  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("SELECT appointments.user_id_doc, user.first_name, user.last_name, user.email, user.phone FROM user,
    appointments, doctor WHERE appointments.user_id_doc = doctor.doctor_id AND doctor.doctor_id  = user.user_id AND appointments.user_id_p = ?;");


    ps.setInt(1, user);

    ResultSet rs = ps.executeQuery();

    while(rs.next())
    {
      out.println("ID: " + rs.getInt(1) + "<br/><br/>");
      out.println("First Name: " + rs.getString(2) + "<br/><br/>");
      out.println("Last Name: " + rs.getString(3) + "<br/><br/>");
      out.println("email: " + rs.getString(4) + "<br/><br/>");
      out.println("Personal Phone: " + rs.getString(5) + "<br/><br/>");


    }


  }
  catch(Exception e){
    out.println(e);
```

This is the code that receives the input credentials of the user to insert into the SQL statement to display their doctor contact. The doctor ID, name, email, and phone number is displayed.

Appointments



Continuation of Doctor, where users can create or view appointments based on their needs. The user can add appointment based on symptoms and date.

The user then can view the appointment that he or she just inputted and any past appointments made.

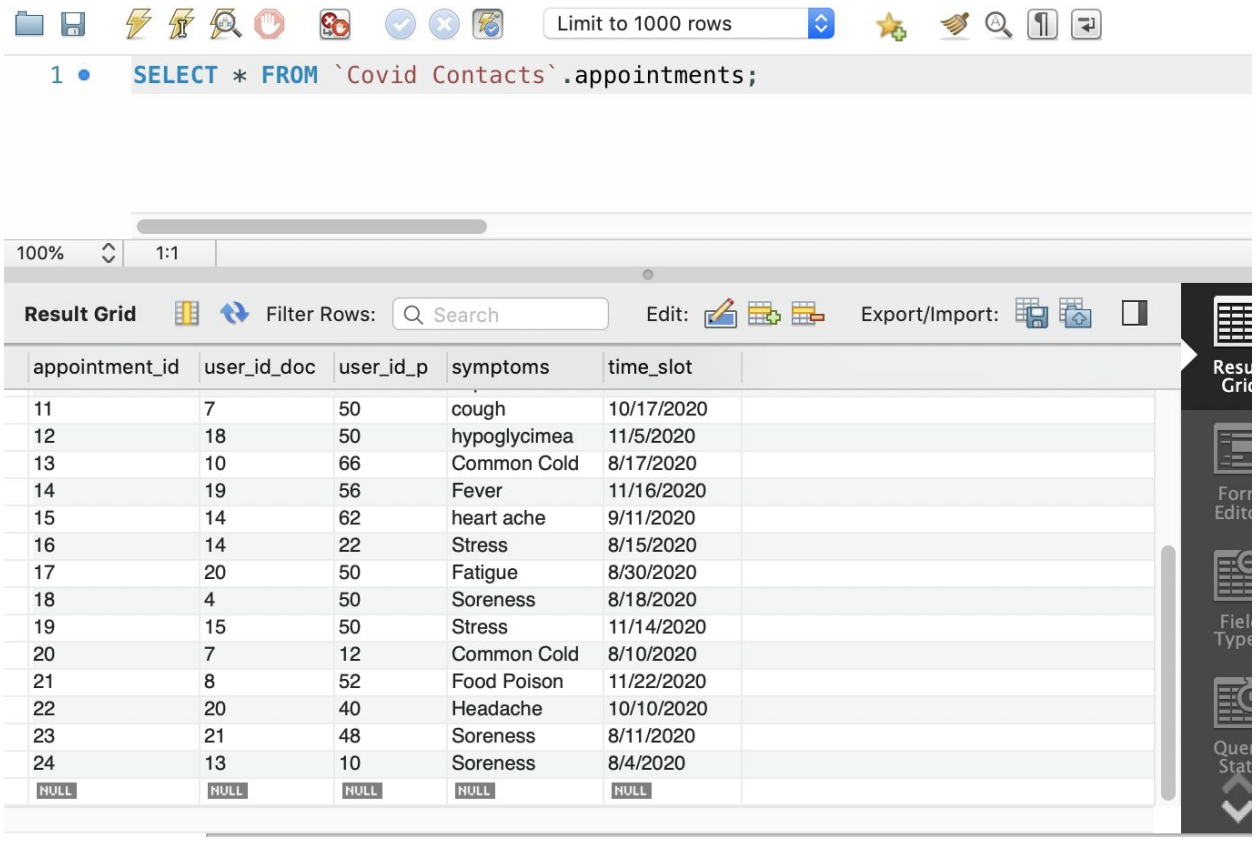
Appointment Front End Code

signInSignUp.html | friends.html | appointment.html

```
12
13  <body class="appointmentpage">
14    <header>
15        <p class="logo">COVID CONTACTS</p>
16
17        <nav>
18          <ul class = "indexlist">
19            <li><a href="index.html">Home</a></li>
20            <li><a href="friends.html">Contact List</a></li>
21            <li><a href="doctor.html">Doctor's Contact</a></li>
22            <li><a href="covidnews.html">Covid News</a></li>
23          </ul>
24          <a href = "user.html" class="user-links">Account</a><br /><br />
25          <a href="signInSignUp.html" class="user-links">Sign Out</a>
26        </nav>
27      </header>
28    <h1 class="appointment">Appointments</h1>
29  <form class="insertAppointment" action="../jsp/insertAppointment.jsp" method="post">
30    <h4>Enter an Appointment</h4>
31      Doctor ID: <input type="number" name="doctorID" id="doctor" /><br/><br/>
32      Your ID: <input type="number" name="userID" id="userID" /><br/><br/>
33            Symptoms: <input type="text" name="symptoms" id="symptoms" /><br/><br/>
34            Date: <input type="text" name="time_slot" id="date" /><br/><br/>
35            <button>Submit</button><br/><br/>
36
37  </form>
38
39  <form class="displayAppointment" action="../jsp/displayAppointment.jsp" method="post">
40    <h4>View Your Appointments</h4>
41    Your ID: <input type="number" name="userID" id="userID" /><br/><br/>
42    <button>View</button>
43  </form>
44
45
46  </body>
```

This is the html for the display of appointment page on browser. There are inputs for user's credentials to add or view their appointments.

Table



appointment_id	user_id_doc	user_id_p	symptoms	time_slot
11	7	50	cough	10/17/2020
12	18	50	hypoglycimea	11/5/2020
13	10	66	Common Cold	8/17/2020
14	19	56	Fever	11/16/2020
15	14	62	heart ache	9/11/2020
16	14	22	Stress	8/15/2020
17	20	50	Fatigue	8/30/2020
18	4	50	Soreness	8/18/2020
19	15	50	Stress	11/14/2020
20	7	12	Common Cold	8/10/2020
21	8	52	Food Poison	11/22/2020
22	20	40	Headache	10/10/2020
23	21	48	Soreness	8/11/2020
24	13	10	Soreness	8/4/2020
NULL	NULL	NULL	NULL	NULL

This is the appointments table to store the appointments of the user. The appointments table will also keep relationships between users and doctors to display doctor and appointment information. As you can see, a new appointment was inserted into the database because of the user's input.

Appointment Back End Code

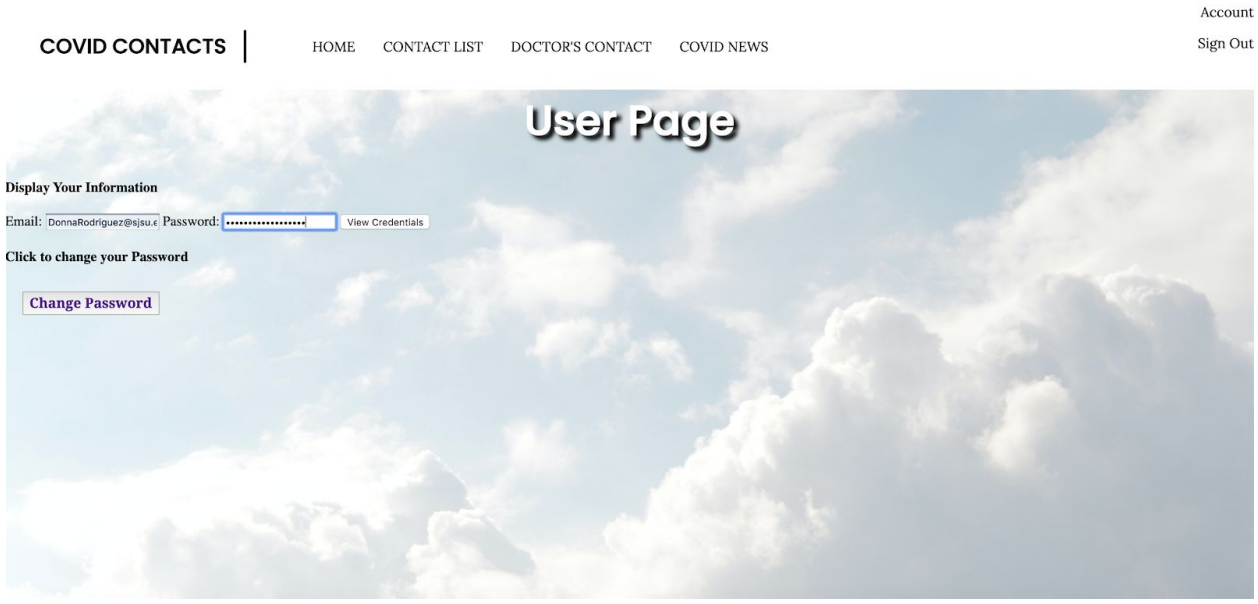
```jsp
<%
  String doctorID = request.getParameter("doctorID");
  int doctor = Integer.parseInt(doctorID);
  String userID = request.getParameter("userID");
  int user = Integer.parseInt(userID);
  String symptoms = request.getParameter("symptoms");
  String date = request.getParameter("time_slot");

  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("INSERT INTO appointments (user_id_doc, user_id_p, symptoms, time_slot) VALUES (?, ?, ?, ?);");

    ps.setInt(1, doctor);
    ps.setInt(2, user);
    ps.setString(3, symptoms);
    ps.setString(4, date);

    int x = ps.executeUpdate();
    if(x>0)
    {
      %> <jsp:include page="success.jsp"/> <%
      out.println("Success");
    }
    else
    {
      out.println("Fail");
    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code to retrieve the user's symptoms and date to be inserted into the database.

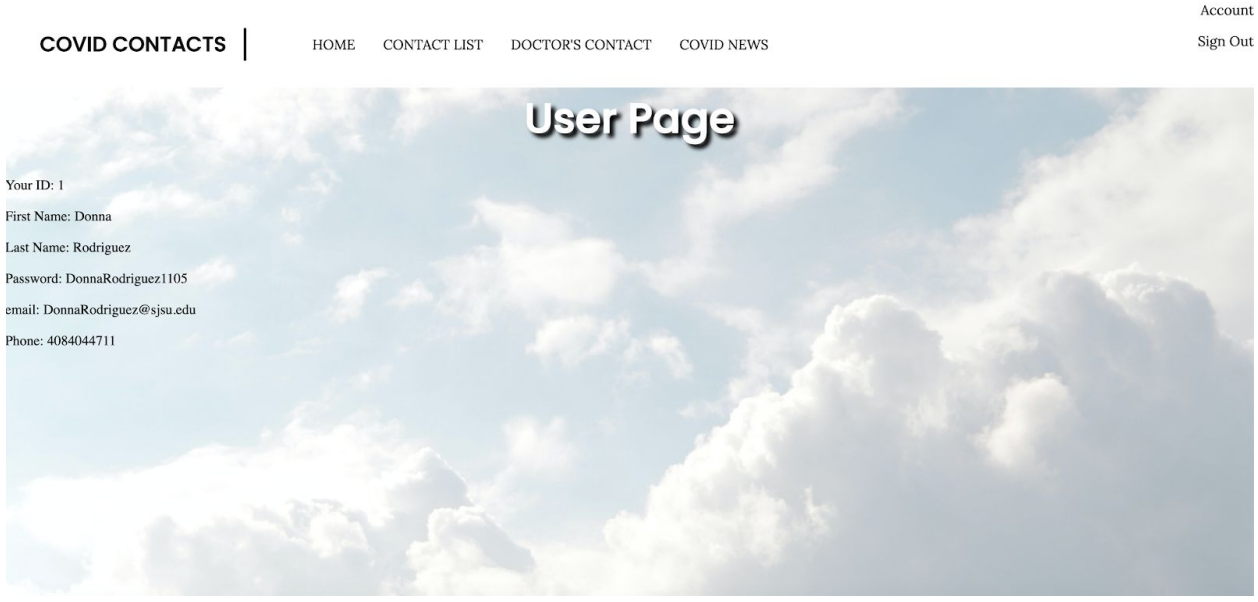
```jsp
      <h1 class="flist">Your Appointments</h1>

<%

  String userID = request.getParameter("userID");
  int user = Integer.parseInt(userID);


  try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("SELECT user.first_name, user.last_name, hospital, symptoms, time_slot FROM appointments, user, doctor WHERE
    appointments.user_id_doc = doctor.doctor_id AND doctor.doctor_id  = user.user_id AND appointments.user_id_p = ?;");

    ps.setInt(1, user);


    ResultSet rs = ps.executeQuery();

    while(rs.next())
    {
      out.println("Doctor Name: " + rs.getString(1) + " " + rs.getString(2) + "<br/><br/>");
      out.println("Hospital: " +rs.getString(3) + "<br/><br/>");
      out.println("Symptoms: " + rs.getString(4) + "<br/><br/>");
      out.println("Date_Slot: " +rs.getString(5) + "<br/><br/>");

    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code for displaying the appointments of the user once the user inputs their credentials.This displays the Doctor's name, hospital, symptoms, and date of the appointment.

UserPage



This is the user's account page to view their credentials once they input their email and password. There is also a change password button to change their password.



Once credentials are inputted, the user can view their full credentials from the database.

UserPage Front End Code

signInSignUp.html friends.html user.html

```
    <link href="https://fonts.googleapis.com/css2?family=Lora&family=Noto+Serif:wght@700&family=Open+Sans+Condensed:wght@300;700&family=Poppins:wght@500&display=swa
    <link rel = "stylesheet" href = "../css/style.css">
  </head>

  <body class = "signUpBody">
    <header>
      <p class="logo">COVID CONTACTS</p>

      <nav>
        <ul class = "indexlist">
          <li><a href="index.html">Home</a></li>
          <li><a href="friends.html">Contact List</a></li>
          <li><a href="doctor.html">Doctor's Contact</a></li>
          <li><a href="covidnews.html">Covid News</a></li>
        </ul>
        <a href = "user.html" class="user-links">Account</a><br /><br />
        <a href="signInSignUp.html" class="user-links">Sign Out</a>
      </nav>
    </header>
    <h1 class = "user">User Page</h1>

    <form class="user-display" action="../jsp/displayUser.jsp" method="post">
      <h4>Display Your Information</h4>
      Email: <input type="email" class="input-box" placeholder="Email" name="email">
      Password: <input type="password" class="input-box" placeholder="Password" name="password">
      <button>View Credentials</button>

    </form>
    <h4>Click to change your Password</h4>
      <button class="changePass-button">
        <a href="passwordRecovery.html">Change Password</a>
      </button>
```

This is the html to format the user page. There are inputs for the user email and password to view their credentials which will be taken into the database.

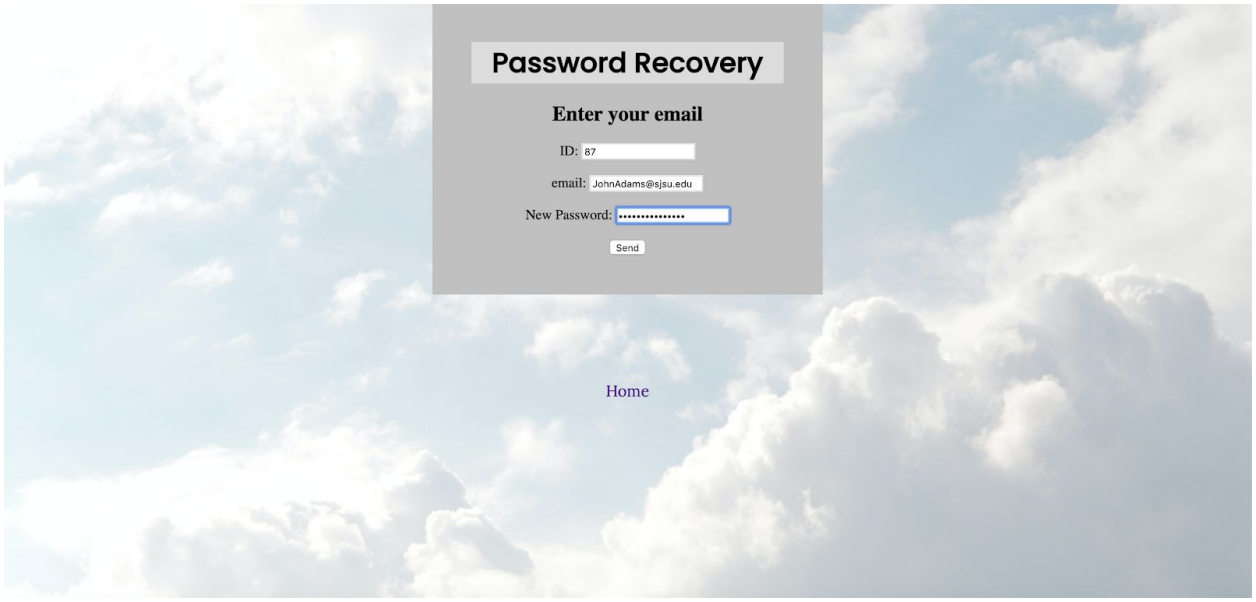
UserPage Back End Code

signInSignUp.html friends.html displayUser.jsp

```jsp
<h1 class = "user">User Page</h1>

<%

  String password = request.getParameter("password");
  String email = request.getParameter("email");


  try
  {
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
    PreparedStatement ps = conn.prepareStatement("SELECT user.user_id, user.first_name, user.last_name, user.user_pass, user.email, user.phone FROM user WHERE
    user.email = ? AND user.user_pass = ?;");
    ps.setString(1,email);
    ps.setString(2,password);


    ResultSet rs = ps.executeQuery();
    while(rs.next())
    {
      out.println("Your ID: " + rs.getInt(1) + "<br/><br/>");
      out.println("First Name: " + rs.getString(2) + "<br/><br/>");
      out.println("Last Name: " + rs.getString(3) + "<br/><br/>");
      out.println("Password: " + rs.getString(4) + "<br/><br/>");
      out.println("email: " + rs.getString(5) + "<br/><br/>");
      out.println("Phone: " + rs.getString(6) + "<br/><br/>");
    }
  }
  catch(Exception e){
    out.println(e);
  }
%>
```

This is the code that retrieves the user's email and password to be put into the SQL statement. Then, the user's full credentials will be displayed based on the user's inputs.

Recover Password



Password recovery page where users can change passwords. The user inputs a new password which will be updated in the database. This makes use of an update query.

Recover Password Front End Code

signInSignUp.html | friends.html | passwordRecovery.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">

<head>
  <meta charset="utf-8">
  <title>Covid Contacts-passwordRecovery</title>
  <link href="https://fonts.googleapis.com/css2?family=Lora&family=Noto+Serif:wght@700&family=Open+Sans+Condensed:wght@300;700&family=Poppins:wght@500&display=swap"
  <link rel="stylesheet" href="../css/style.css">
</head>

<body class="signUpBody">
  <div class="form-recovery">
    <form class="password-recovery" action="../jsp/updatePassword.jsp" method="post">
      <h1 class="recovery-page">Password Recovery</h1>
      <h2 class="enter">Enter your email</h2>
      ID: <input type="number" name="userID" id="userID" /><br /><br />
      email: <input type="text" name="email" id="email" /><br /><br />
      New Password: <input type="password" name="new-password" id="new-password" /><br /><br />
      <button>Send</button>
    </form>


  </div>

  <div class="check">
    <a href="index.html">Home</a>
  </div>
</body>
```

This is the html for the format of the recoverpage. There are inputs for the user's credentials and then an input for their new password to be updated in the database.

Table

```
1 •  SELECT * FROM covid_contacts.user;
```

100% 1:1

Result Grid | Filter Rows: Search | Edit: | Export/Import:

user_id	first_name	last_name	user_pass	email	phone
74	Alice	Clark	AliceClark209	AliceClark@sjsu.edu	4084043973
75	James	Canby	JamesCanby1113	JamesCanby@sjsu.edu	4084046725
76	Alec	Tang	testPassword	AlecTang@sjsu.edu	4086077709
77	Alex	Collins	Collinsword	AlexCollins@sjsu.edu	4087543254
78	Frank	Murphy	MyNewPassword	FRankMurphy@sjsu.edu	4086783905
79	Ryan	Pins	RyanPassword	RyanPins@sjsu.edu	4089873467
80	Beth	Owens	BethPassword	BethOwens@sjsu.edu	4086751238
81	Josh	Tom	JoshPassword	JoshTom@sjsu.edu	4089651234
82	Brad	Cole	BradPassword	BradCole@sjsu.edu	4087461234
83	John	Mayor	JohnMayorword	JohnMayor.sjsu.edu	4086753245
84	Bryan	Young	NewWord	BryanYoung@sjsu.edu	4087546957
85	Dave	Brown	Updatedpassword	DaveBrown@sjsu.edu	4086573267
86	Aaron	Thomas	AaronNewPassword	AasronThomas@sjsu.edu	4087599978
87	John	Adams	NewJohnPassword	JohnAdams@sjsu.edu	4087569043
NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

user 1 | Apply | Revert

Action Output

As you can see the password was updated in the database once the user inputted a new password.

Recover Password Back End Code

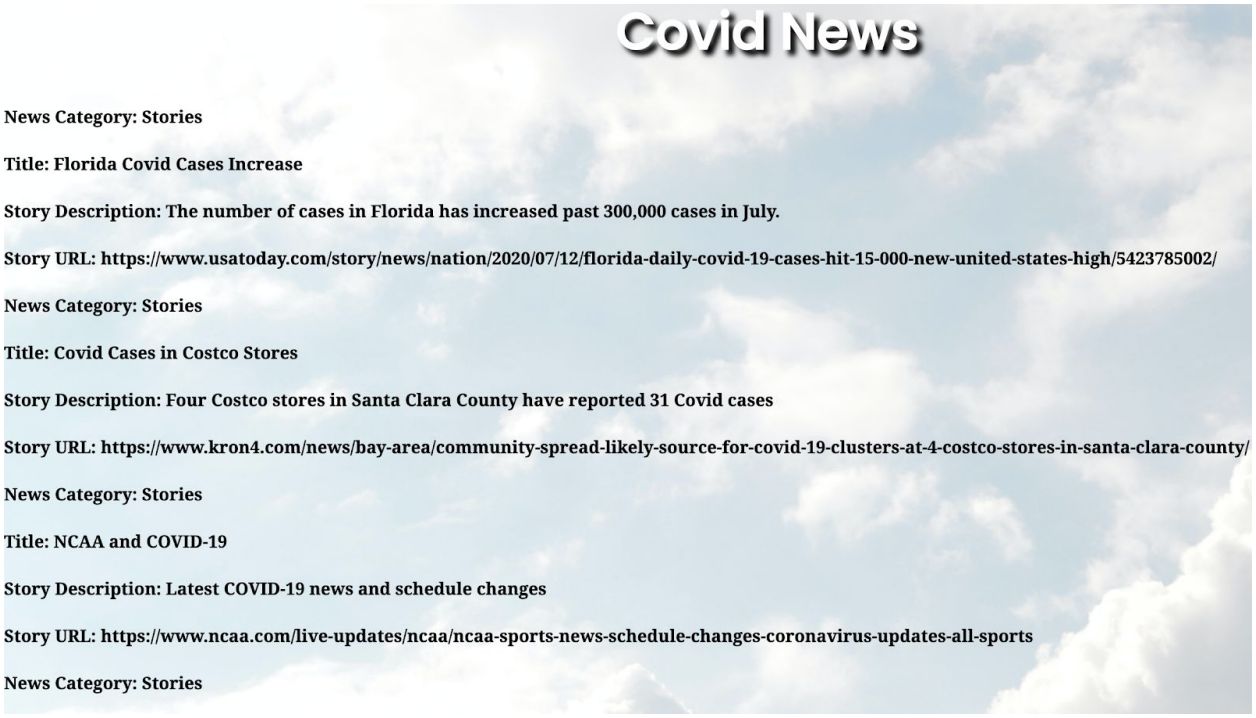
signInSignUp.html | friends.html | updatePassword.jsp

```
<%@ page import = "java.sql.*"%>

  <%
    String userID = request.getParameter("userID");
    int id = Integer.parseInt(userID);
    String email = request.getParameter("email");
    String newPassword = request.getParameter("new-password");
    try{
      Class.forName("com.mysql.jdbc.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
      PreparedStatement ps = conn.prepareStatement("UPDATE user SET user_pass = ? WHERE user_id = ? AND email = ?;");
      ps.setString(1, newPassword);
      ps.setInt(2,id);
      ps.setString(3,email);

      int x = ps.executeUpdate();;

      if(x>0)
      {
        out.println("Success");
      }
      else
      {
        out.println("Fail");
      }
    }
    catch(Exception e){
      out.println(e);
```

This is the code to retrieve the user's credentials and new password to be used for the query. The credentials are used for the database to determine which user entity needs to be changed and the new password is used to set the new password in the database.

CovidNews

This is the page for the user to search covid new sources. The user can search based off categories of Stories, Stats, and Safety.



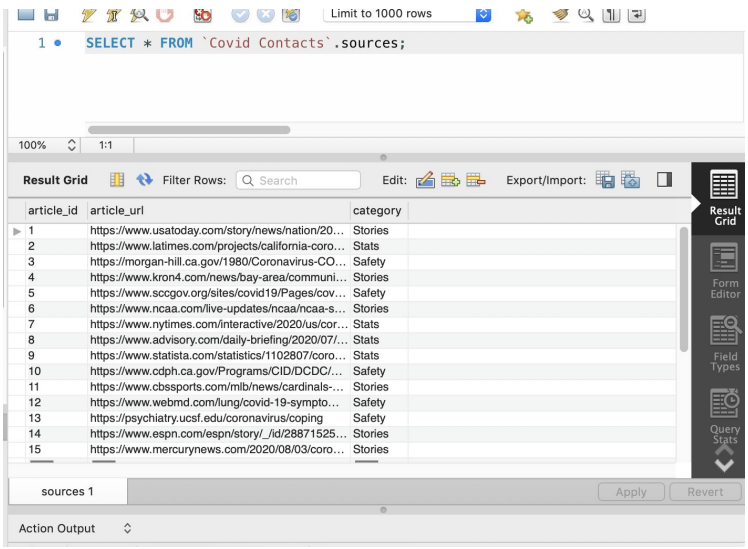
When the user inputs their desired category, a list of news articles will pop up for the user to view.

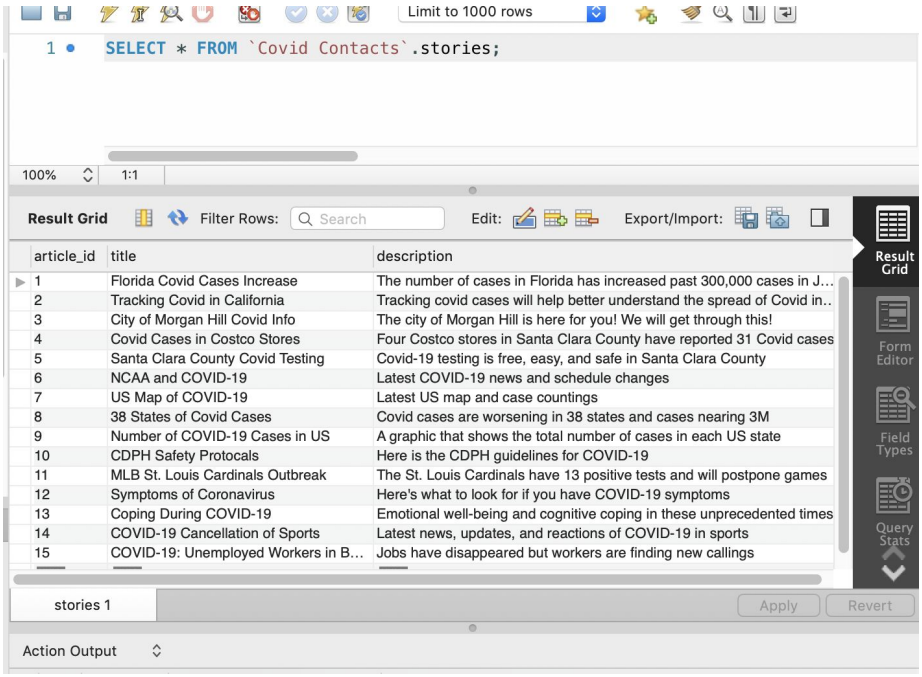
CovidNews Front End Code

```
    <header>
      <p class="logo">COVID CONTACTS</p>

      <nav>
        <ul class = "indexlist">
          <li><a href="index.html">Home</a></li>
          <li><a href="friends.html">Contact List</a></li>
          <li><a href="doctor.html">Doctor's Contact</a></li>
          <li><a href="covidnews.html">Covid News</a></li>
        </ul>
        <a href = "user.html" class="user-links">Account</a><br /><br />
        <a href="signInSignUp.html" class="user-links">Sign Out</a>
      </nav>
    </header>
    <h1 class = "news">Covid News</h1>


        <form class="search-news" action="../jsp/displayNews.jsp" method="post">
          <h4>Search for Covid stories by category</h4>
          <input type="text" name = "category" placeholder="Stories, Stats, Safety">
          <button id="headlines-button" name="headlines-button">Search</button>
        </form>


</body>
```

This is the html to format the CovidNews Page. There are inputs for the category of sources the user wants to view.

Table





These are the tables for the sources and news of Covid. The two tables are connected and can be used to display information of the new article and the source and the category.

CovidNews Back End Code

```
    <body class="newspage">
      <h1 class = "news">Covid News</h1>

  <%
    String category = request.getParameter("category");
    try{
      Class.forName("com.mysql.jdbc.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Covid Contacts","root","B0ldin81");
      PreparedStatement ps = conn.prepareStatement("SELECT sources.category, stories.title, stories.description, sources.article_url FROM sources, stories WHERE
      sources.article_id = stories.article_id AND sources.category = ?;");

      ps.setString(1,category);

      ResultSet rs = ps.executeQuery();

      while(rs.next())
      {
        out.println("News Category: " + rs.getString(1) + "<br/><br/>");
        out.println("Title: " + rs.getString(2) + "<br/><br/>");
        out.println("Story Description: " + rs.getString(3) + "<br/><br/>");
        out.println("Story URL: " +rs.getString(4) + "<br/><br/>");


      }
```

This is the code that will retrieve the category the user wants and will be used for the SQL statement to display the full information of the covid headlines based on the category.

Lesson Learned

Mark Brogan

Throughout the project, we've learned new ways to adapt to a fast pace workflow and use several resources to aid our success. This project proved tough, with losing a member.. Alec and I experienced first hand on shifting workforce and available persons. Unfortunately, this project was about half the time compared to a normal semester. Hands-on experience was relatively short and felt ultimately rushed. For such a large scale project, if I were to do it again with a group, I would definitely have a lot more time to do a project of this scale and have a more structured planning of functionalities and workflow.

Alec Tang

Our creation of the application Covid Contacts was definitely the most challenging experience I have faced in college. With the semester being so short, class being online, learning new languages, and losing a team member, it was very stressful putting together the project with Mark. This was also my first time ever creating a functioning web application, and it put me to the test of learning to build a website through HTML and CSS. Before this class, I mainly only knew Java but this project allowed me to experience other languages. A lot of research of HTML, CSS, SQL, and JSP was dedicated to this project and has made me a more

well-rounded software engineer. Learning new languages and collaborating as a team has definitely opened my eyes to a whole other side of software engineering and has piqued my interests in new languages. Overall, this project helped me to persevere through adversity and I have learned a lot about teamwork, web design, and database management. It was such a gratifying feeling to complete our project that was able to function properly. I have learned a lot about collaborating and brainstorming as a team, which made me motivated to complete our project. From my experience of our project, I can definitely use these skills for my future as a software engineer and as a person.