

## CSE530 Database Management Project --- St. Louis Housing

Project by: Tianpei Li, Yanqing Sun, Wei Wang, Kun Zhang

### 1 Readme – direction on how to run the project.

- a Front End: url, any login and what user can do

url for this housing website:

<http://52.33.71.235/~CSE530/cse530/>

Please click or copy and paste the link to log into this website.

The frontend of this website will provide several layers for searching and results listing: in the search layer: client side user can search houses by zipcode, address; In the results layer: matched results will be listed based on the queries from search layer, then user can choose a desired result and enter into third layer; Third layer: provides detailed house information and support distance queries for more information.

The direction on how to query data with this website:

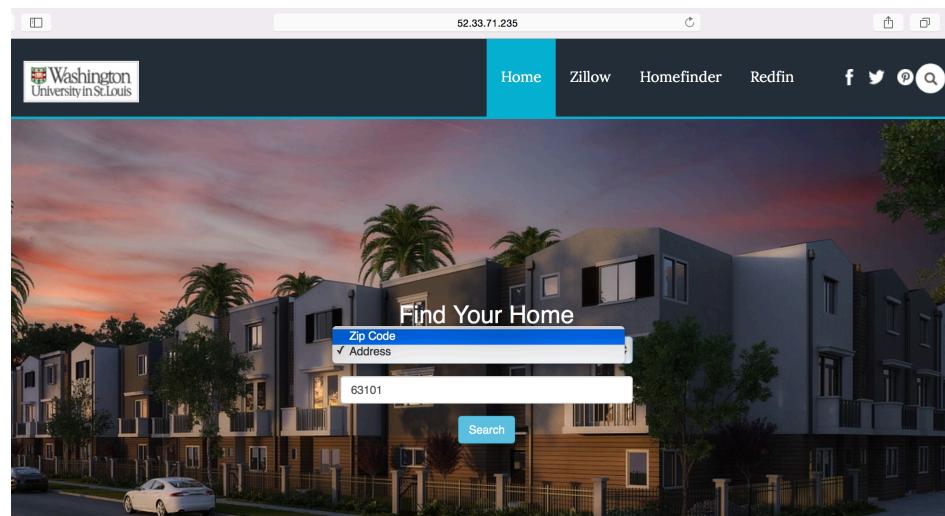
When you log into the homepage, you will find that the housing website supports house searching based on zip code or address as you will. By default, you can find houses based on address. Just input address you interested and press search button, it will list all the houses, which match the address you inputted. Or you can select 'zipcode' option, which supports your searching based on zip code. Just input 5-digital zip code number, and press search. It will list all the houses located in this zip code area.

Once you find the house based on the query. Click on the house, it will show you all the detailed information about the house. More importantly, you can also find other valuable information related to this house, which includes the demography data in the area (such as: ratio of Female to male, economic information, Age distribution, Race distribution, etc), how secure of this location (such as: how many offenders live nearby), how convenient of this location (such as: schools and restaurants nearby).

Listed are some shortnap about the website:

1: Search layer, search houses by zip code or address:

**zip code range:** For this project, data are only included the St. Louis City area with the listed zip codes: 63147, 63120, 63115, 63112, 63113, 63107, 63108, 63106, 63110, 63103, 63101, 63102, 63104, 63139, 63118, 63109, 63116, 63111



2. display results based on the query:

52.33.71.235

St.Louis Housing Back to Home

## Available Houses

Address	Bedrooms	Bathrooms	Value	Size	Zest_value	Zipcode	City	State
55 Maryland Plz	3	3	\$214900	1616	\$364780	63108	Saint Louis	MO
28 Carriage Lan	5	4	\$485000	2995	\$523576	63108	Saint Louis	MO
30 32 N Boyle A	4	4	\$339900	2500	\$225232	63108	St Louis	MO
4716 Westminste	7	8	\$1750	15500	\$1416	63108	Saint Louis	MO
4938 Pershing P	2	3	\$289500	1964	\$270293	63108	Saint Louis	MO
4563 Laclede Av	2	1	\$165000	960	\$151594	63108	Saint Louis	MO
4542 Maryland A	5	4	\$639000	3272	\$480768	63108	Saint Louis	MO
4480 Maryland A	2	2	\$169000	1196	\$182039	63108	Saint Louis	MO
5123 Cates Ave	2	3	\$599000	1687	\$564066	63108	Saint Louis	MO
5173 Kensington	7	4	\$795000	5766	\$640421	63108	Saint Louis	MO
4141 Enright Av	4	3	\$285000	2450	\$357477	63108	Saint Louis	MO
40 Portland Pl	2	2	\$209900	1286	\$258671	63108	Saint Louis	MO
37 Portland Pl	2	2	\$245000	1197	\$226717	63108	Saint Louis	MO
5083 Washington	2	2	\$209900	1689	\$282420	63108	Saint Louis	MO

3. After you click on a specific house, the third layer page will display the specific house information, as well as all the valuable information related to this house with fancy pictures. Those information includes: Offenders, Schools and Restaurants nearby the house, and some useful demography data in figures.

52.33.71.235

St.Louis Housing Back to Home

**55 Maryland Plz**

- Value: \$214900
- Size: 1616 sqft
- Bedrooms: 3
- Bathrooms: 3

Danger! 34 Recorded Offenders within 1 Mile of the house!

Warning! 814 Recorded Offenders within 5 Miles of the house!

Info! 1262 Recorded Offenders within 10 Miles of the house!

Schools within 1 mile click to show them

Restaurants within 2 miles click to show them

**Female/Male**

Male/Female

Total Population: 1568

FusionCharts XT Trial

**Economic Information**

Average House Value: \$289.5K

Income per Household: \$35.64K

In USD

FusionCharts XT Trial

**Age Distribution**

Age Distribution

FusionCharts XT Trial

**Race**

Race Distribution

FusionCharts XT Trial

Legend for Age Distribution:

- Under 10
- 10 to 20
- 20 to 40
- 40 to 60
- Over 85

Legend for Race:

- White
- Black
- Hispanic
- Asian
- Indian
- Hawaiian

b. Back End: access to database, username and password.

For this project, we used AWS EC2 as developing platform, Apache + PHP as our server backend to handle requests from client side and retrieve data from database.

Access to database: <http://52.33.71.235/phpmyadmin/>  
Username: root

## 2 Project Description

### a Project Goal

The existing house searching websites, such as Zillow.com, scout.com etc, only provide customers the house information. Those websites may not provide their customers an objective evaluation system, which really helps them to find a perfect accommodation. In our project, we will build a better house searching website, which helps people find the satisfied house as their expected. With this website, customers can easily find all valuable information about the house, as well a full view of the particular district where the house locates, with which wise decisions could be made.

So, the goal of this project is to build a housing website – called **St. Louis Housing**, which can provide multiple ways for customers to find their desired house. In this website, we will provide information include: houses on market, offenders nearby, demography data associate with the house, most of the facilities such as schools or famous restaurants nearby the house. With this website, people can easily find their desired house based on house information, or the other information they interested.

- b **Data** – What data do you have? Where did you download the data? What did you have to do to scrape, and clean the data? Size of data – number of attributes and number of observation

For this project, data are only included the St. Louis City area with the listed zip codes: 63147, 63120, 63115, 63112, 63113, 63107, 63108, 63106, 63110, 63103, 63101, 63102, 63104, 63139, 63118, 63109, 63116, 63111.

This CSE530 database includes 8 tables (house, district, offender, offense\_type, race, district\_has\_race, schools, and restaurant) and two 2 views, with total 2518 records of real data.

The listed table includes the information as: where did we download the data for each table? How do we get and clean the data? Numbers of attributes and observations for each table.

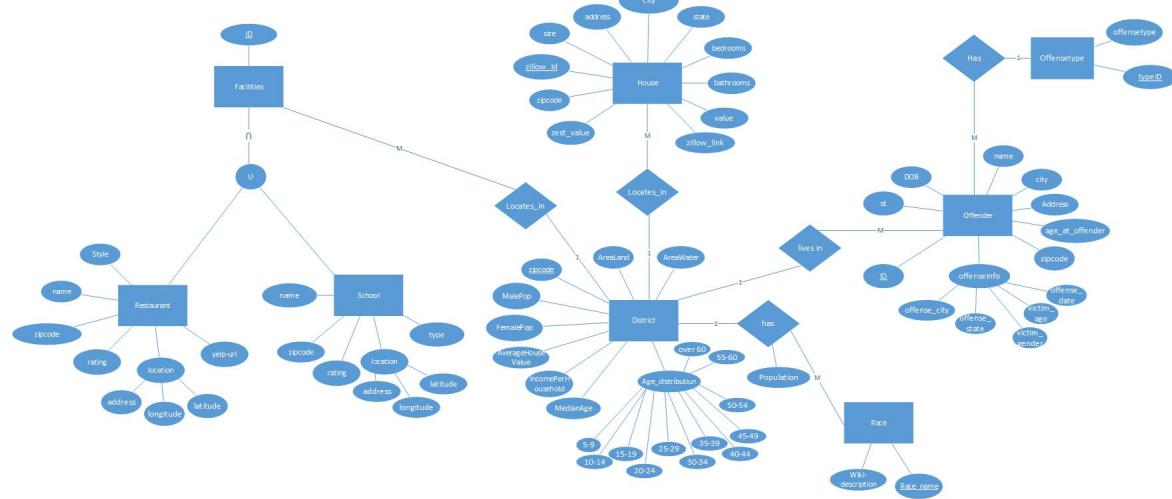
Schem a Name	Data Source	How do we get and trim the data	Number of attributes	Data detail and description
House	Zillow.com	Python scripts to send request and parse html responses in order to extract data.  Assign -1 to unknown value,bedrooms,bathrooms.	13 attributes, including (address, city, state, value, zillow_link, bathrooms, bedrooms, zestvalue, etc), with total 456 records.	We assign zillow_id as the primary key of House. <a href="http://52.33.71.235/~CSE530/data/house.txt">http://52.33.71.235/~CSE530/data/house.txt</a>
Schools	greatschools.org	PHP scrapy framework to scrabble data.	8 attributes, including (id, name, address, zipcode, rating, type, latitude, longitude), with total 372 records.	Providing schools' information for different area; <a href="http://52.33.71.235/~CSE530/data/school.txt">http://52.33.71.235/~CSE530/data/school.txt</a>
Offender	<a href="http://www.msdp.dps.missouri.gov">www.msdp.dps.missouri.gov</a>	Downloaded the excel file from the website. Added the Longitude and altitude values for each address.	16 attributes, including (id, name, address, zipcode, offense_type, etc), with total 1264 records.	<a href="http://52.33.71.235/~CSE530/data/offender.txt">http://52.33.71.235/~CSE530/data/offender.txt</a>
Offense_type	This table is normalized from 'Offender' table based on the offense type.		2 attributes (typeID, offense_type), with total 198 records.	<a href="http://52.33.71.235/~CSE530/data/offense_type.txt">http://52.33.71.235/~CSE530/data/offense_type.txt</a>
district	Zip-codes.com	API	32 attributes including (zipcode, MalePop, ect), with total 18 records.	<a href="http://52.33.71.235/~CSE530/data/district.txt">http://52.33.71.235/~CSE530/data/district.txt</a>

districtHasRace	Zip-codes.com	API	3 attributes, including (raceName, population and zipcode), with total 108 records;	<a href="http://52.33.71.235/~CSE530/data/district_has_race.txt">http://52.33.71.235/~CSE530/data/district_has_race.txt</a>
restaurant	yelp.com	API	9 attributes, including (name, yelpid, ect), with total 96 records.	<a href="http://52.33.71.235/~CSE530/data/restaurant.txt">http://52.33.71.235/~CSE530/data/restaurant.txt</a>
race	A small table with two attributes, raceName and wikilink linked to the description of the race. With total 6 records.			<a href="http://52.33.71.235/~CSE530/data/race.txt">http://52.33.71.235/~CSE530/data/race.txt</a>

### 3 Data Modeling

#### a ER Diagram:

See the attached files (ER\_Diagram.vsv or ER\_Diagram.pdf) for more details. Listed is the figure of this ER diagram:



#### b Normalized Table:

We normalized the offender schema into two tables: offender and offendertype based on the offense type.

## 4 Programming

### a System description

Database System: MySQL (RDMS)  
System Server: Ubuntu 14.04 LTS  
Server Side Language: LAMP (Linux, Apache, MySQL, PHP)  
Frontend (HTML5, Bootstrap, JavaScript)

Our project is deployed in LAMP architecture. PHP was used to retrieve data from our database and handle post/get request. Also, Ajax was implemented to handle some special cases in our data.

In front end, JQuery was implemented to insert or hide/show some HTML elements. And Bootstrap 3 was used to make our Web app more concise and beautiful.

Cool JavaScript library: JavaScript library – FunisionCharts –was used to create chart and map for the third layer web pages.

### b Sample SQL statements and scripts

- 1) To normalize the offender type, we created a table - offendertype, and then based on the offense type from offender table to normalize into two tables. The SQL statements are listed below:

- a). create a offense\_type table and input data based on the type from offender table:

```
CREATE TABLE offense_type
(typeID int, offendertype varchar(100),
PRIMARY KEY (typeID));
```

- b). update data in offender table, replace the types with their related type ID:

```
UPDATE offender d
JOIN offense_type s
ON s.offendertype = d.offense
SET d.offendertype = s.typeID
```

- 2) In table district, since there are two many attributes with the age distribution. To use fewer attributes to include all the age ranges, we created a view – district\_A. The MySQL query is listed below:

```
CREATE VIEW district_A
as SELECT zipcode, AverageHouseValue, IncomePerHouseHold, AverageFamilySize,
AreaLand, AreaWater,MalePop as Malepopulation, FemalePop as FemalePopulation,
under_5+5_9 as AgeUnder10YearsOld, 10_14+15_19 as AgeFrom10to20,
20_24+25_29+30_34+35_39 as AgeFrom20to40, 40_44+45_49+50_54+55_59+60_64 as
AgeFrom40to65, 65_69+70_74+75_79+80_84 as AgeFrom65to85, 85_over+0 as 85YearsOld
FROM district;
```

- 3) Created a view – districttrace, which includes the percent attribute based on the related population.

```
CREATE VIEW districttrace AS
SELECT zipcode, raceName, population,
population/
(SELECT SUM(population)
FROM districtHasRace d
WHERE d.zipcode = r.zipcode)*100 AS percent
FROM districtHasRace r;
```

```

if($search_option=='zipcode'){
    $stmt=$mysqli->prepare('select *from house where zipcode=?');
    //error handler
    if(!$stmt){
        printf("Query Prep Failed: %s\n", $mysqli->error);
        exit;
    }
    $stmt->bind_param('s',$search_value);
    $stmt->execute();
    $query_results=$stmt->get_result();
    $results=array();
    $stmt->close();
    //pass the result to results array
    $index=0;
    while($row=$query_results->fetch_assoc()){
        $results[$index]=array();
        $results[$index]['zipcode']=$row['zipcode'];
        $results[$index]['zillow_id']=$row['zillow_id'];
        $results[$index]['address']=$row['address'];
        $results[$index]['size']=$row['size'];
        $results[$index]['bedrooms']=$row['bedrooms'];
        $results[$index]['bathrooms']=$row['bathrooms'];
        $results[$index]['city']=$row['city'];
        $results[$index]['state']=$row['state'];
        $results[$index]['value']=$row['value'];
        $results[$index]['zest_value']=$row['zest_value'];
        $index++;
    }
    $_SESSION['results']=$results;
}

```

- 5) Sample program code on house searching based on address:

```

else{
    $stmt=$mysqli->prepare('select *from house');
    //error handler
    if(!$stmt){
        printf("Query Prep Failed: %s\n", $mysqli->error);
        exit;
    }
    $results=array();
    $stmt->execute();
    $query_results=$stmt->get_result();
    $stmt->close();
    $index=0;
    while($row=$query_results->fetch_assoc()){
        $target=$row['address'];
        //find the match address
        if(strpos($target,$search_value)!==false || strpos($search_value,$target)!==false){
            $results[$index]=array();
            $results[$index]['zipcode']=$row['zipcode'];
            $results[$index]['zillow_id']=$row['zillow_id'];
            $results[$index]['address']=$row['address'];
            $results[$index]['size']=$row['size'];
            $results[$index]['bedrooms']=$row['bedrooms'];
            $results[$index]['bathrooms']=$row['bathrooms'];
            $results[$index]['city']=$row['city'];
            $results[$index]['state']=$row['state'];
            $results[$index]['value']=$row['value'];
            $results[$index]['zest_value']=$row['zest_value'];
            $index++;
        }
    }
    $_SESSION['results']=$results;
}

```

- 6) Sample scripts on fetch offenders information nearby a house:

```

116 //offender
117 $stmt=$mysqli->prepare("select lat,lng from offender");
118 $stmt->execute();
119 $offender_array=array();
120 $index=0;
121 $query_results=$stmt->get_result();
122 while($row=$query_results->fetch_assoc()){
123     $offender_array[$index++]=$row;
124 }
125 $stmt->close();

```

## 5 Conclusion

- a Did the data support your project goal? Why and why not

The data do support our project goal very well. This project supports regular queries based on house address or zip code, also supplying plenty of data about the security environment, demography, the distance to the desired school or other facilities related to the house.

- b In addition to supporting the project goal, are there any additional queries your database able to support?

In addition, this database also supports house search function based on house value, and the distance to offenders, schools or restaurants.

## 6 Appendix

### All scripts:

All scripts related to this project were packed into cse530.zip file. The cse530.zip includes the listed files:

- a) ER\_Diagram folder (ER diagram about this project in PDF and .vsdx formats);
- b) DataScrape folder (all the script files for scraping data for this project);
- c) QueryForCreateTables.pdf (all the queries for table creation and data population);
- d) SourceCode folder ( all the source codes for this website).