Project 2 Part 3

Answer each of the following questions in **BLUE** colored font.

Project:

Include name of the team members.

Ty Painter & Tonnar Castellano

What is the general status of the project? Describe the roadblocks (if you have any) 3-4 sentences.

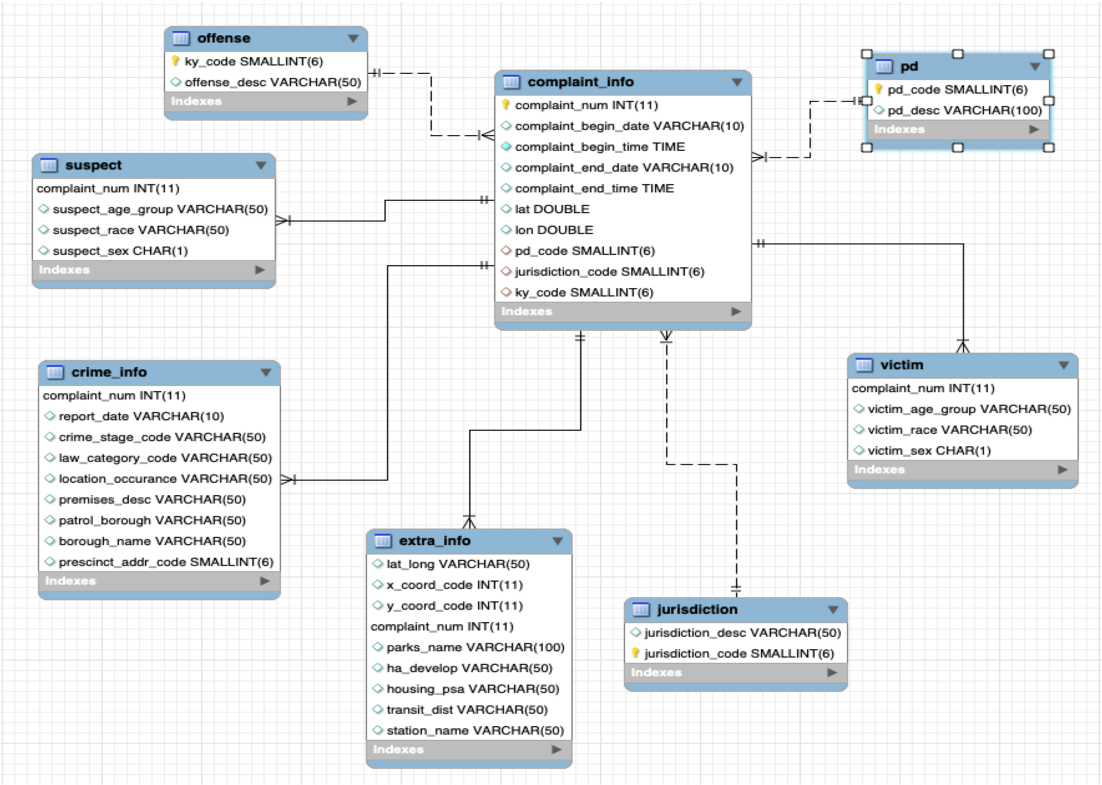
Our project is on track. We encountered some roadblocks handling dirty data within our data set. We decided to create a dirty_data table to remove NULL/duplicate primary key values. We are currently having trouble creating a trigger to accept proper date values.

Database:

By this assignment submission date, you should have your decomposition completed with all the tables ready in database. You should have also implemented many of the advanced features. Clearly label the screenshots.

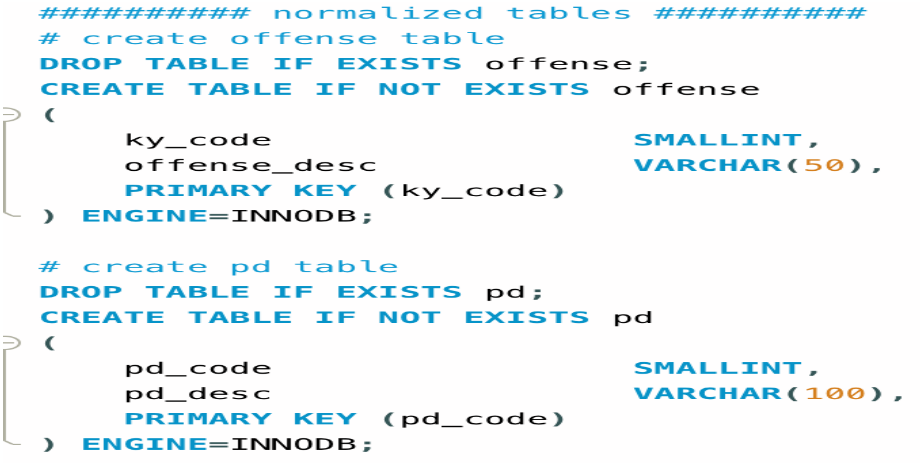
Include the UML screenshot.

UML screenshot

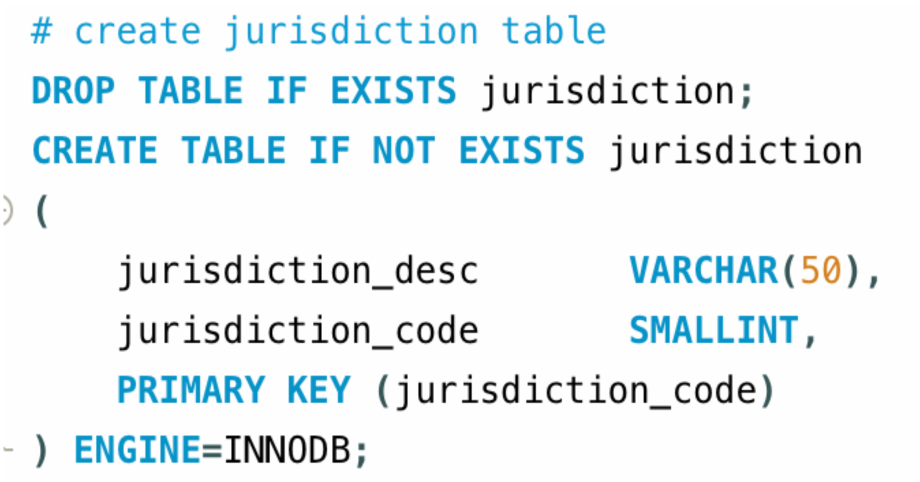


Include the screenshots of decomposed tables.

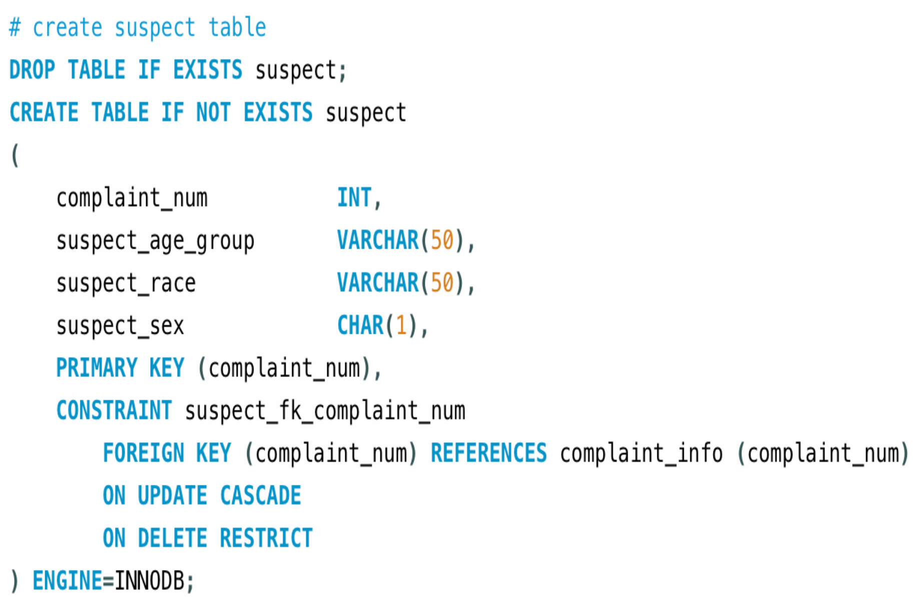
decomposed tables



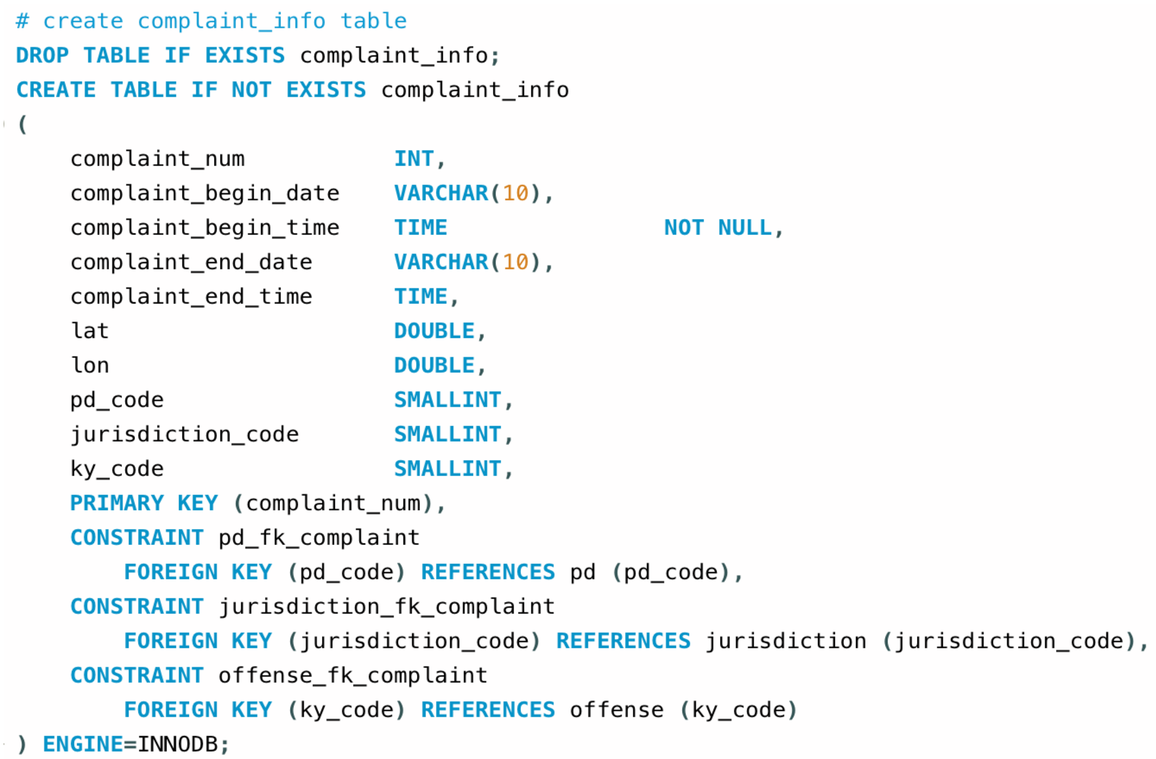
```
########## normalized tables ##########
# create offense table
DROP TABLE IF EXISTS offense;
CREATE TABLE IF NOT EXISTS offense
(
    ky_code                     SMALLINT,
    offense_desc                VARCHAR(50),
    PRIMARY KEY (ky_code)
) ENGINE=INNODB;

# create pd table
DROP TABLE IF EXISTS pd;
CREATE TABLE IF NOT EXISTS pd
(
    pd_code                     SMALLINT,
    pd_desc                     VARCHAR(100),
    PRIMARY KEY (pd_code)
) ENGINE=INNODB;
```



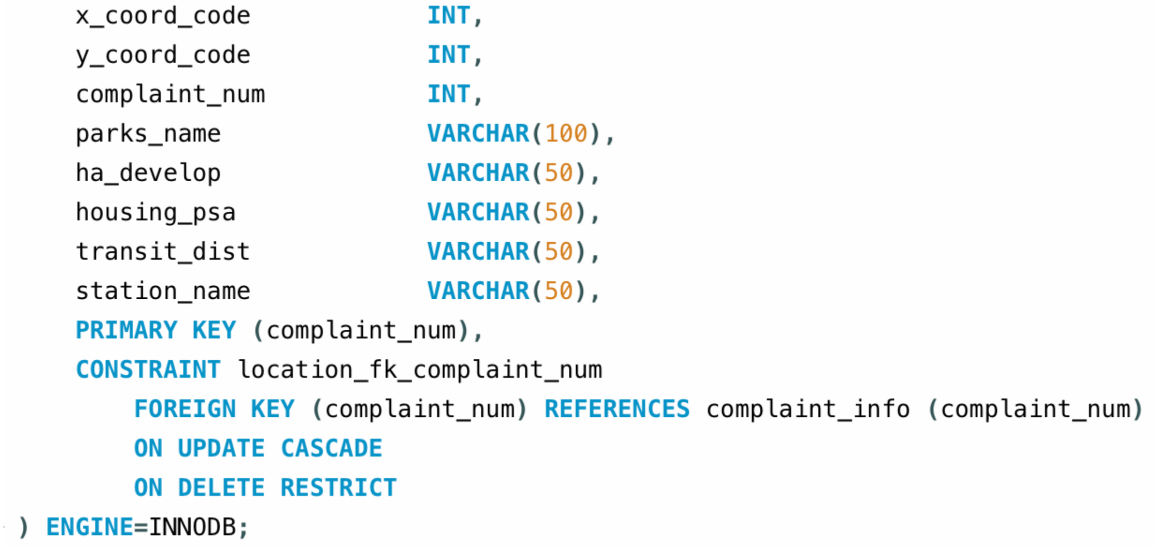
```
# create jurisdiction table
DROP TABLE IF EXISTS jurisdiction;
CREATE TABLE IF NOT EXISTS jurisdiction
(
    jurisdiction_desc       VARCHAR(50),
    jurisdiction_code       SMALLINT,
    PRIMARY KEY (jurisdiction_code)
) ENGINE=INNODB;
```



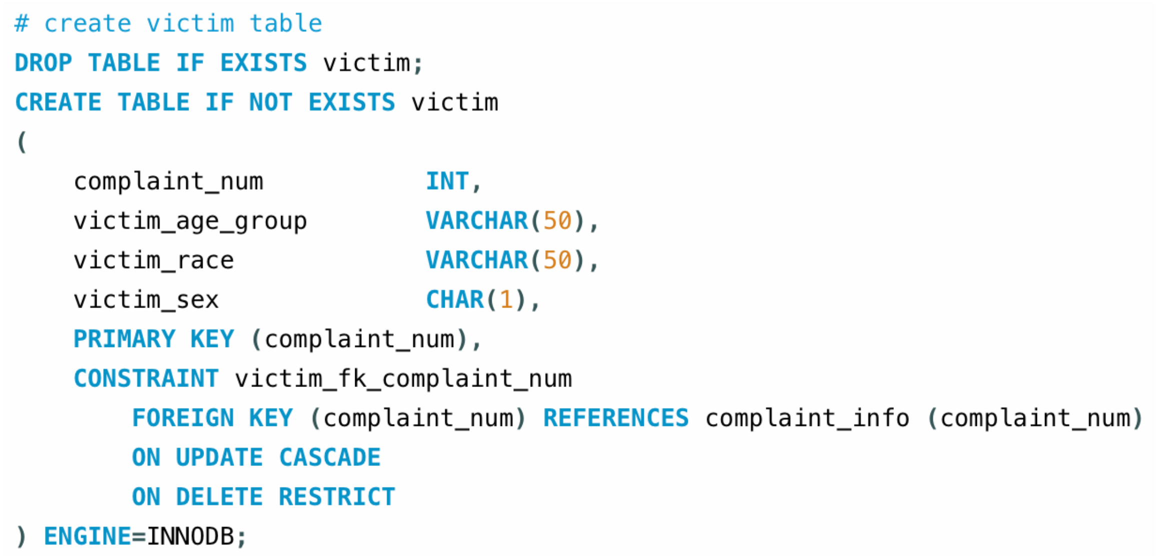
```
# create suspect table
DROP TABLE IF EXISTS suspect;
CREATE TABLE IF NOT EXISTS suspect
(
    complaint_num           INT,
    suspect_age_group       VARCHAR(50),
    suspect_race            VARCHAR(50),
    suspect_sex             CHAR(1),
    PRIMARY KEY (complaint_num),
    CONSTRAINT suspect_fk_complaint_num
        FOREIGN KEY (complaint_num) REFERENCES complaint_info (complaint_num)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
) ENGINE=INNODB;
```



```
# create complaint_info table
DROP TABLE IF EXISTS complaint_info;
CREATE TABLE IF NOT EXISTS complaint_info
(
    complaint_num           INT,
    complaint_begin_date    VARCHAR(10),
    complaint_begin_time    TIME                NOT NULL,
    complaint_end_date      VARCHAR(10),
    complaint_end_time      TIME,
    lat                     DOUBLE,
    lon                     DOUBLE,
    pd_code                 SMALLINT,
    jurisdiction_code       SMALLINT,
    ky_code                 SMALLINT,
    PRIMARY KEY (complaint_num),
    CONSTRAINT pd_fk_complaint
        FOREIGN KEY (pd_code) REFERENCES pd (pd_code),
    CONSTRAINT jurisdiction_fk_complaint
        FOREIGN KEY (jurisdiction_code) REFERENCES jurisdiction (jurisdiction_code),
    CONSTRAINT offense_fk_complaint
        FOREIGN KEY (ky_code) REFERENCES offense (ky_code)
) ENGINE=INNODB;
```



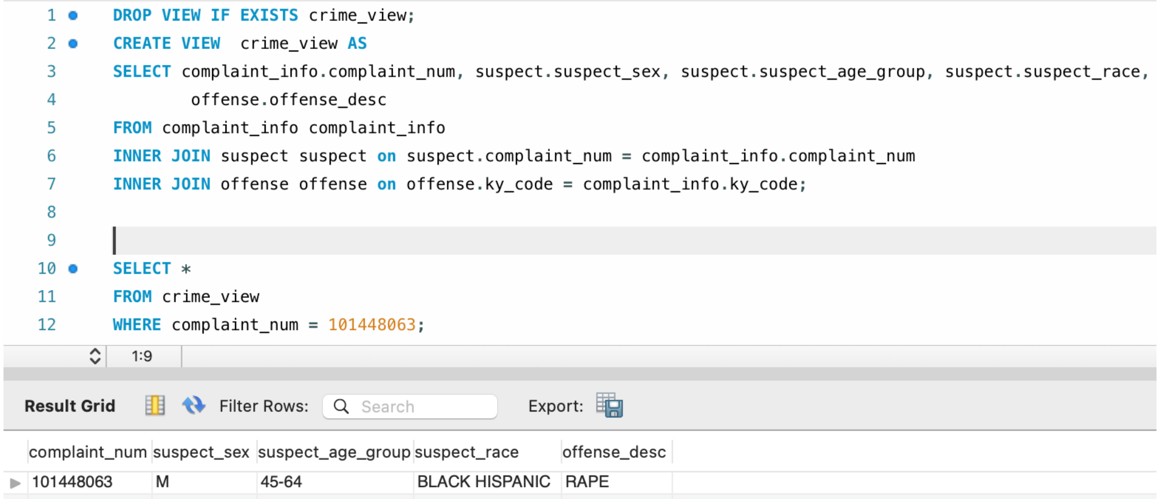
```
    x_coord_code            INT,
    y_coord_code            INT,
    complaint_num           INT,
    parks_name              VARCHAR(100),
    ha_develop              VARCHAR(50),
    housing_psa             VARCHAR(50),
    transit_dist            VARCHAR(50),
    station_name            VARCHAR(50),
    PRIMARY KEY (complaint_num),
    CONSTRAINT location_fk_complaint_num
        FOREIGN KEY (complaint_num) REFERENCES complaint_info (complaint_num)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
) ENGINE=INNODB;
```



```
# create victim table
DROP TABLE IF EXISTS victim;
CREATE TABLE IF NOT EXISTS victim
(
    complaint_num           INT,
    victim_age_group        VARCHAR(50),
    victim_race             VARCHAR(50),
    victim_sex              CHAR(1),
    PRIMARY KEY (complaint_num),
    CONSTRAINT victim_fk_complaint_num
        FOREIGN KEY (complaint_num) REFERENCES complaint_info (complaint_num)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
) ENGINE=INNODB;
```

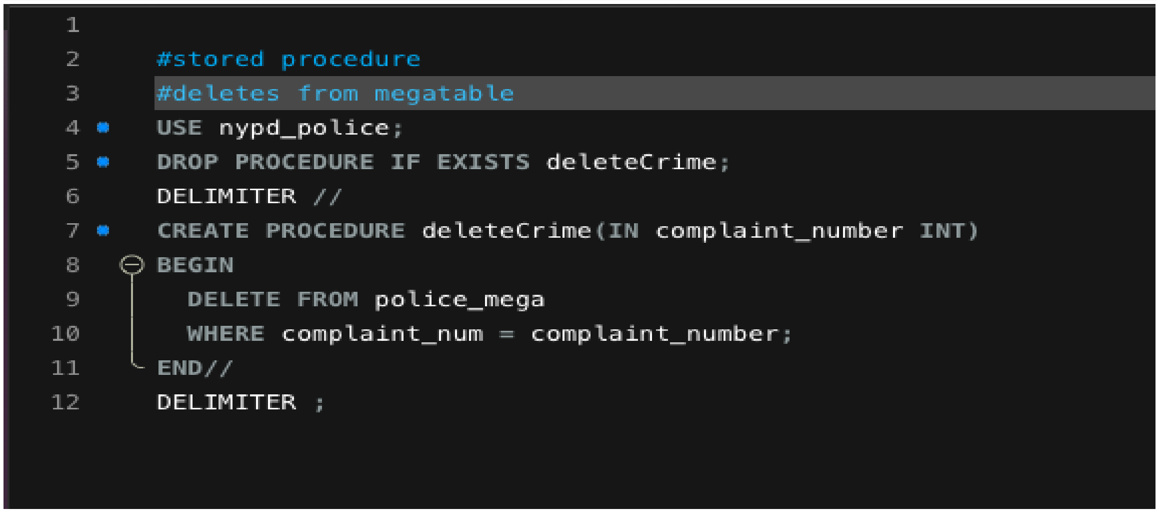
Include screenshots of your implementation of some advanced features (Stored procedures, views, triggers etc.) In terms of percentage how much you think you have completed on database side of the project? Describe the completed work in 2-3 sentences. Describe roadblocks, if any.

We are through most of the advanced features. We would like to add a trigger and and another stored procedure. We are 80% done.



```
DROP VIEW IF EXISTS crime_view;
CREATE VIEW  crime_view AS
SELECT complaint_info.complaint_num, suspect.suspect_sex, suspect.suspect_age_group, suspect.suspect_race,
        offense.offense_desc
FROM complaint_info complaint_info
INNER JOIN suspect suspect on suspect.complaint_num = complaint_info.complaint_num
INNER JOIN offense offense on offense.ky_code = complaint_info.ky_code;


SELECT *
FROM crime_view
WHERE complaint_num = 101448063;
```

complaint_num	suspect_sex	suspect_age_group	suspect_race	offense_desc
101448063	M	45-64	BLACK HISPANIC	RAPE



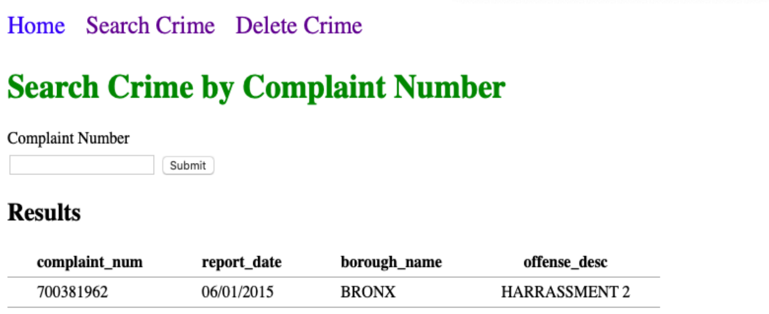
```

#stored procedure
#deletes from megatable
USE nypd_police;
DROP PROCEDURE IF EXISTS deleteCrime;
DELIMITER //
CREATE PROCEDURE deleteCrime(IN complaint_number INT)
BEGIN
  DELETE FROM police_mega
  WHERE complaint_num = complaint_number;
END//
DELIMITER ;
```

Front end:

By this assignment submission date, you should have a working interface showing data from the database in live connection. You should also have multiple UI pages. Clearly label the screenshots.

Attach screenshots of the browser showing different UI pages. (At least 2 excluding home page)

UI Page 1



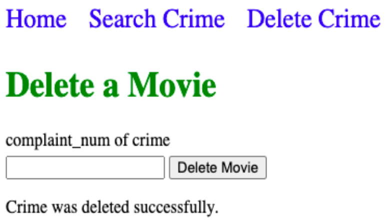
Search Crime by Complaint Number

Complaint Number

Results

complaint_num	report_date	borough_name	offense_desc
700381962	06/01/2015	BRONX	HARRASSMENT 2

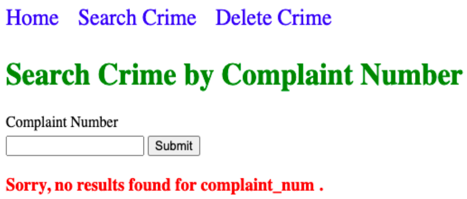
UI Page 2



Delete a Movie

complaint_num of crime

Crime was deleted successfully.



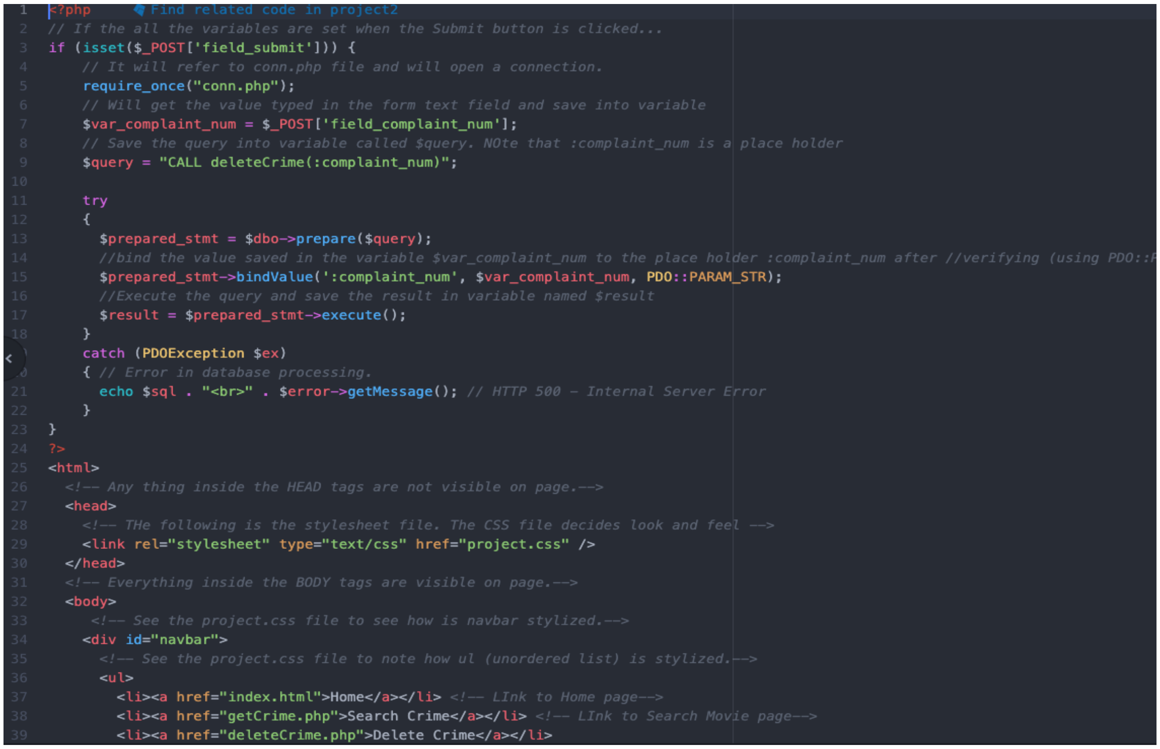
Search Crime by Complaint Number

Complaint Number

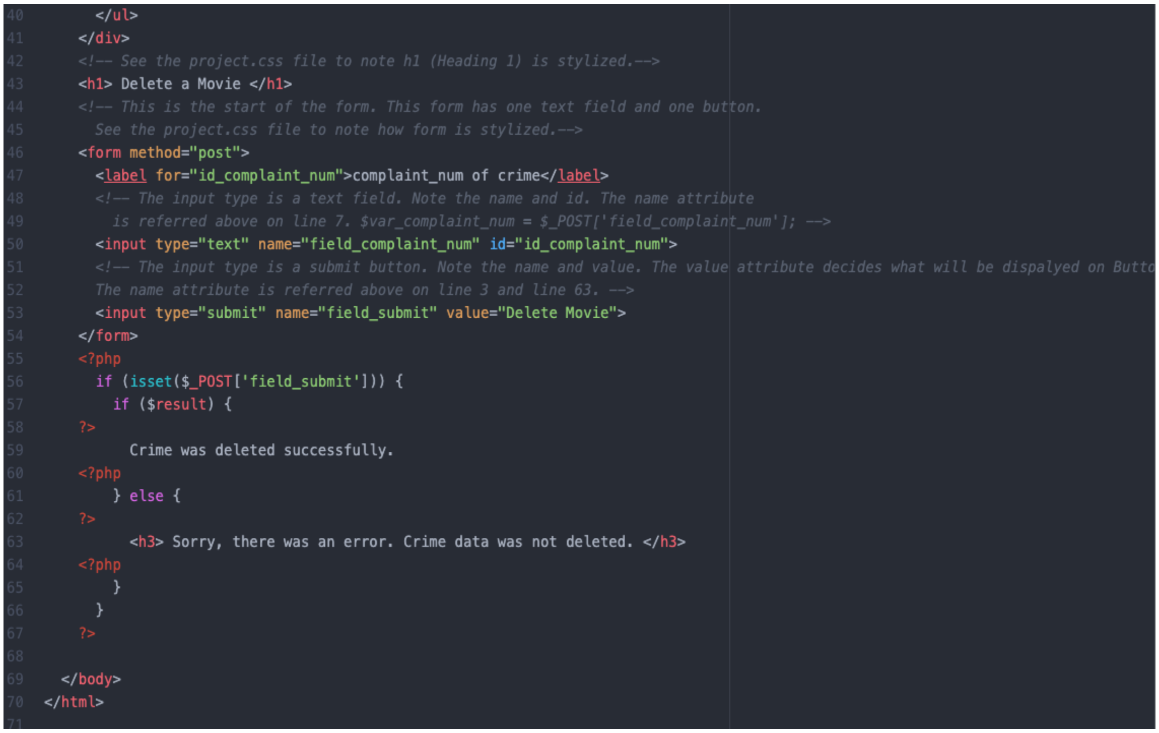
Sorry, no results found for complaint_num .

Attach screenshots of relevant part of your front-end code showing calls to the database. This can be embedded SQL statements or calls to procedures etc.

Call to delete a crime

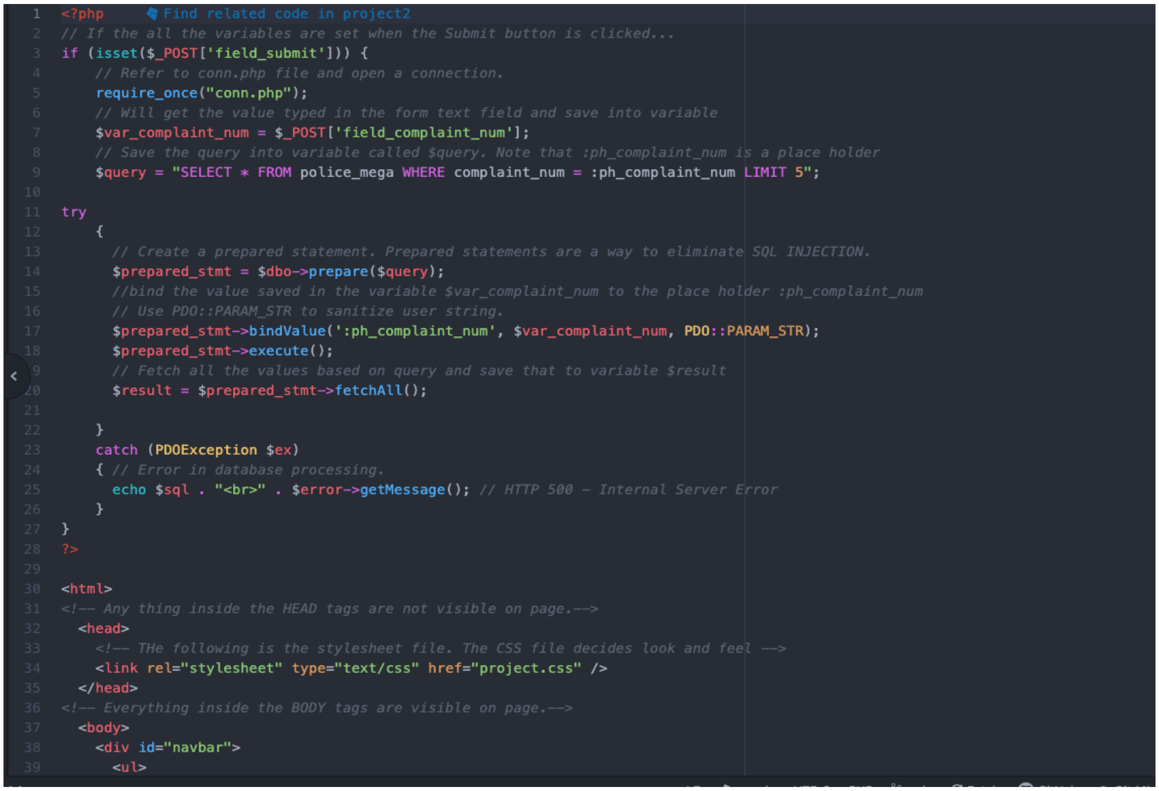


```
1  <?php      Find related code in project2
2  // If the all the variables are set when the Submit button is clicked...
3  if (isset($_POST['field_submit'])) {
4      // It will refer to conn.php file and will open a connection.
5      require_once("conn.php");
6      // Will get the value typed in the form text field and save into variable
7      $var_complaint_num = $_POST['field_complaint_num'];
8      // Save the query into variable called $query. NOte that :complaint_num is a place holder
9      $query = "CALL deleteCrime(:complaint_num)";
10
11     try
12     {
13       $prepared_stmt = $dbo->prepare($query);
14       //bind the value saved in the variable $var_complaint_num to the place holder :complaint_num after //verifying (using PDO::P
15       $prepared_stmt->bindValue(':complaint_num', $var_complaint_num, PDO::PARAM_STR);
16       //Execute the query and save the result in variable named $result
17       $result = $prepared_stmt->execute();
18     }
19     catch (PDOException $ex)
20     { // Error in database processing.
21       echo $sql . "<br>" . $error->getMessage(); // HTTP 500 - Internal Server Error
22     }
23 }
24 ?>
25 <html>
26   <!-- Any thing inside the HEAD tags are not visible on page.-->
27   <head>
28     <!-- THe following is the stylesheet file. The CSS file decides look and feel -->
29     <link rel="stylesheet" type="text/css" href="project.css" />
30   </head>
31   <!-- Everything inside the BODY tags are visible on page.-->
32   <body>
33      <!-- See the project.css file to see how is navbar stylized.-->
34     <div id="navbar">
35       <!-- See the project.css file to note how ul (unordered list) is stylized.-->
36       <ul>
37         <li><a href="index.html">Home</a></li> <!-- LInk to Home page-->
38         <li><a href="getCrime.php">Search Crime</a></li> <!-- LInk to Search Movie page-->
39         <li><a href="deleteCrime.php">Delete Crime</a></li>
```

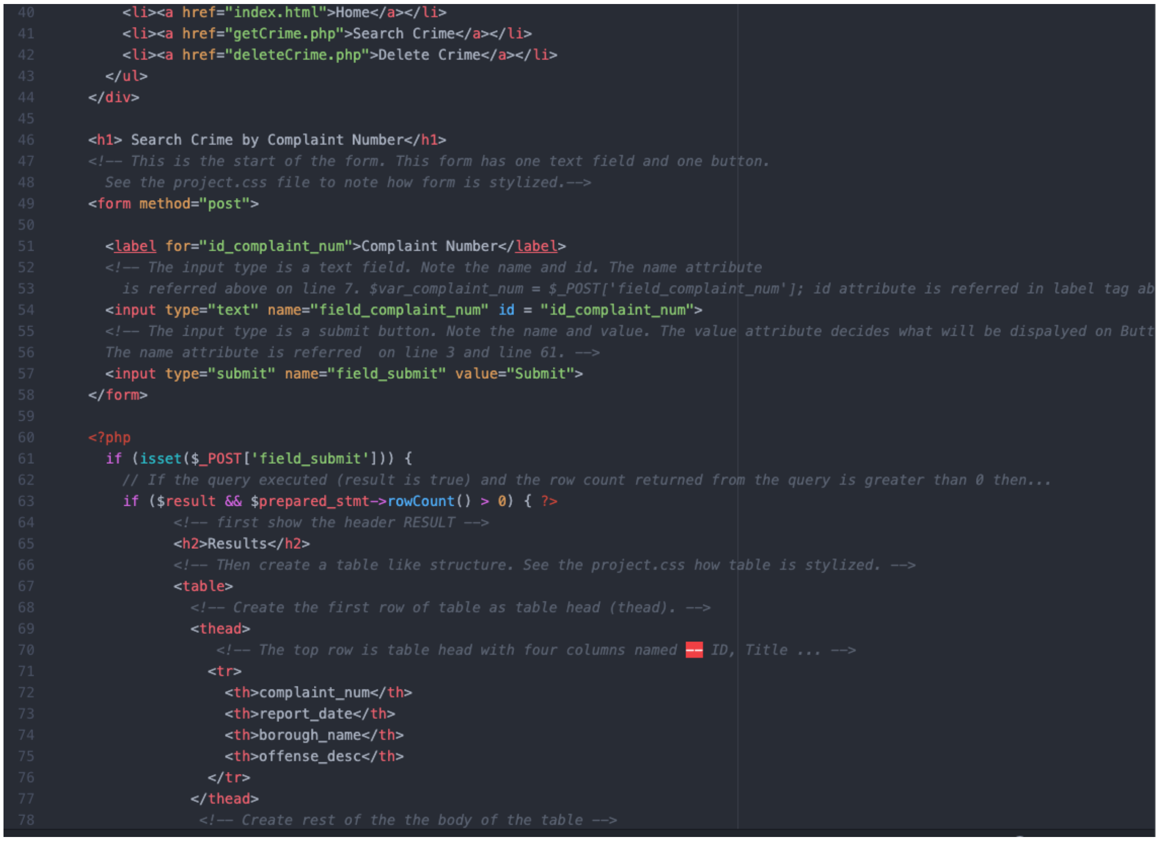


```
40       </ul>
41     </div>
42     <!-- See the project.css file to note h1 (Heading 1) is stylized.-->
43     <h1> Delete a Movie </h1>
44     <!-- This is the start of the form. This form has one text field and one button.
45       See the project.css file to note how form is stylized.-->
46     <form method="post">
47       <label for="id_complaint_num">complaint_num of crime</label>
48       <!-- The input type is a text field. Note the name and id. The name attribute
49         is referred above on line 7. $var_complaint_num = $_POST['field_complaint_num']; -->
50       <input type="text" name="field_complaint_num" id="id_complaint_num">
51       <!-- The input type is a submit button. Note the name and value. The value attribute decides what will be dispalyed on Butto
52       The name attribute is referred above on line 3 and line 63. -->
53       <input type="submit" name="field_submit" value="Delete Movie">
54     </form>
55     <?php
56       if (isset($_POST['field_submit'])) {
57         if ($result) {
58     ?>
59           Crime was deleted successfully.
60     <?php
61         } else {
62     ?>
63           <h3> Sorry, there was an error. Crime data was not deleted. </h3>
64     <?php
65         }
66       }
67     ?>
68
69   </body>
70 </html>
71
```

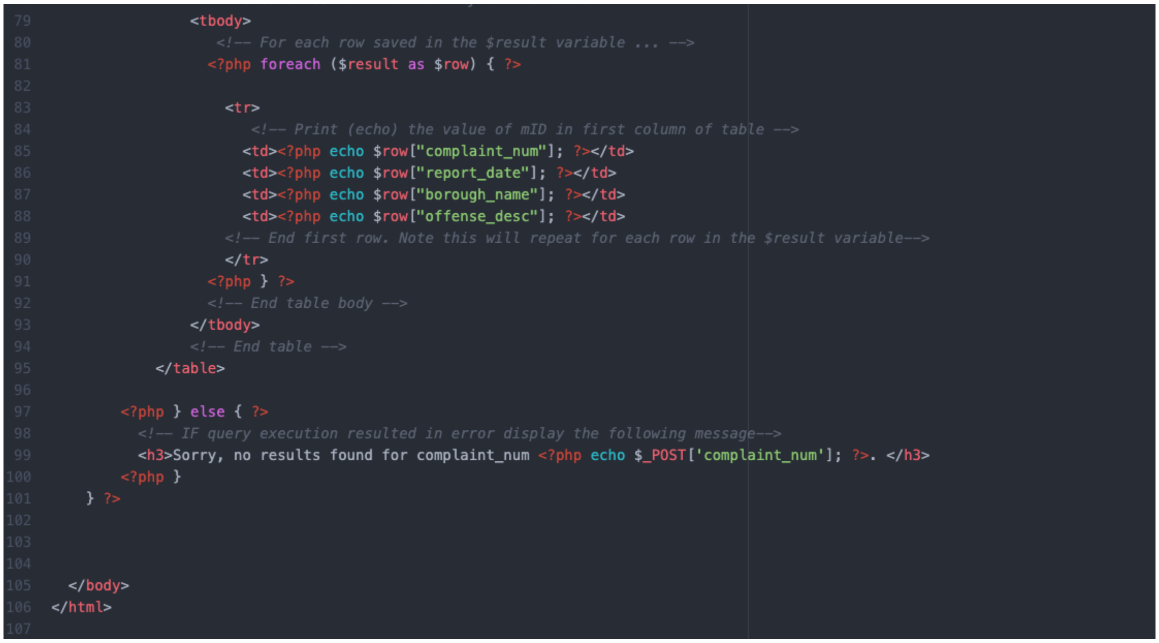
Call to search a crime



```
1  <?php      Find related code in project2
2  // If the all the variables are set when the Submit button is clicked...
3  if (isset($_POST['field_submit'])) {
4      // Refer to conn.php file and open a connection.
5      require_once("conn.php");
6      // Will get the value typed in the form text field and save into variable
7      $var_complaint_num = $_POST['field_complaint_num'];
8      // Save the query into variable called $query. Note that :ph_complaint_num is a place holder
9      $query = "SELECT * FROM police_mega WHERE complaint_num = :ph_complaint_num LIMIT 5";
10
11 try
12     {
13       // Create a prepared statement. Prepared statements are a way to eliminate SQL INJECTION.
14       $prepared_stmt = $dbo->prepare($query);
15       //bind the value saved in the variable $var_complaint_num to the place holder :ph_complaint_num
16       // Use PDO::PARAM_STR to sanitize user string.
17       $prepared_stmt->bindValue(':ph_complaint_num', $var_complaint_num, PDO::PARAM_STR);
18       $prepared_stmt->execute();
19       // Fetch all the values based on query and save that to variable $result
20       $result = $prepared_stmt->fetchAll();
21
22     }
23     catch (PDOException $ex)
24     { // Error in database processing.
25       echo $sql . "<br>" . $error->getMessage(); // HTTP 500 - Internal Server Error
26     }
27 }
28 ?>
29
30 <html>
31 <!-- Any thing inside the HEAD tags are not visible on page.-->
32   <head>
33     <!-- THe following is the stylesheet file. The CSS file decides look and feel -->
34     <link rel="stylesheet" type="text/css" href="project.css" />
35   </head>
36 <!-- Everything inside the BODY tags are visible on page.-->
37   <body>
38     <div id="navbar">
39       <ul>
```



```
40         <li><a href="index.html">Home</a></li>
41         <li><a href="getCrime.php">Search Crime</a></li>
42         <li><a href="deleteCrime.php">Delete Crime</a></li>
43       </ul>
44     </div>
45
46     <h1> Search Crime by Complaint Number</h1>
47     <!-- This is the start of the form. This form has one text field and one button.
48       See the project.css file to note how form is stylized.-->
49     <form method="post">
50
51       <label for="id_complaint_num">Complaint Number</label>
52       <!-- The input type is a text field. Note the name and id. The name attribute
53         is referred above on line 7. $var_complaint_num = $_POST['field_complaint_num']; id attribute is referred in label tag ab
54       <input type="text" name="field_complaint_num" id = "id_complaint_num">
55       <!-- The input type is a submit button. Note the name and value. The value attribute decides what will be dispalyed on Butt
56       The name attribute is referred  on line 3 and line 61. -->
57       <input type="submit" name="field_submit" value="Submit">
58     </form>
59
60     <?php
61       if (isset($_POST['field_submit'])) {
62         // If the query executed (result is true) and the row count returned from the query is greater than 0 then...
63         if ($result && $prepared_stmt->rowCount() > 0) { ?>
64               <!-- first show the header RESULT -->
65               <h2>Results</h2>
66               <!-- THen create a table like structure. See the project.css how table is stylized. -->
67               <table>
68                 <!-- Create the first row of table as table head (thead). -->
69                 <thead>
70                    <!-- The top row is table head with four columns named — ID, Title ... -->
71                   <tr>
72                     <th>complaint_num</th>
73                     <th>report_date</th>
74                     <th>borough_name</th>
75                     <th>offense_desc</th>
76                   </tr>
77                 </thead>
78                  <!-- Create rest of the the body of the table -->
```



```
                <tbody>
                   <!-- For each row saved in the $result variable ... -->
                  <?php foreach ($result as $row) { ?>

                    <tr>
                       <!-- Print (echo) the value of mID in first column of table -->
                      <td><?php echo $row["complaint_num"]; ?></td>
                      <td><?php echo $row["report_date"]; ?></td>
                      <td><?php echo $row["borough_name"]; ?></td>
                      <td><?php echo $row["offense_desc"]; ?></td>
                    <!-- End first row. Note this will repeat for each row in the $result variable-->
                    </tr>
                  <?php } ?>
                  <!-- End table body -->
                </tbody>
                <!-- End table -->
            </table>

        <?php } else { ?>
          <!-- IF query execution resulted in error display the following message-->
          <h3>Sorry, no results found for complaint_num <?php echo $_POST['complaint_num']; ?>. </h3>
        <?php }
    } ?>



  </body>
</html>
```

Have you implemented any DML components (INSERT, UPDATE, DELETE) yet in your project (Note that you must have DML components in the final version of your project).

As seen above we have included a delete statement in the crime database. We still need to do the insert and update. However, given that we have figured out how to delete that should be much simpler. We estimate we are 75% of the way done.

What is the status of front-end application? In terms of percentage how much do you think, you have completed on front-end side of the project? Describe what is completed and what is left. 2-3 sentences.

The status of the front end application is nearly completed. We are around 70% of the way through. We just have left to add an insert page for the front end.

Report and demo video

- Have you started working on your report? If yes, how much is completed (percentage)

  We have an outline of the various steps needed to create the report. We are around 15% complete.

- Which tool you are planning to use to create a demo video of your application?

  We will try to use imovie. However this is very much subject to change.

Final Submission

- The final submission of Project will be due on 4/26/2020 at 12 noon.

Submission:

Complete this document and save it as pdf. You must submit a PDF file named **p2-part3-lastname1-lastname2.pdf** (For example if I submit this document with John Smith, I would name it p2-part3-singh-smith.pdf). Submit your files on Brightspace.

You must include your name and your partner name in the Brightspace submission text box.

Each member of the team must make the submission of same file.

Grading:

This Assignment will be graded on the following criteria:

1. Completeness of document.
2. Completeness of required components at this stage of project.
3. Clear evidence of work completed.

NO grading will be done on file/s sent through email or not uploaded to Brightspace.