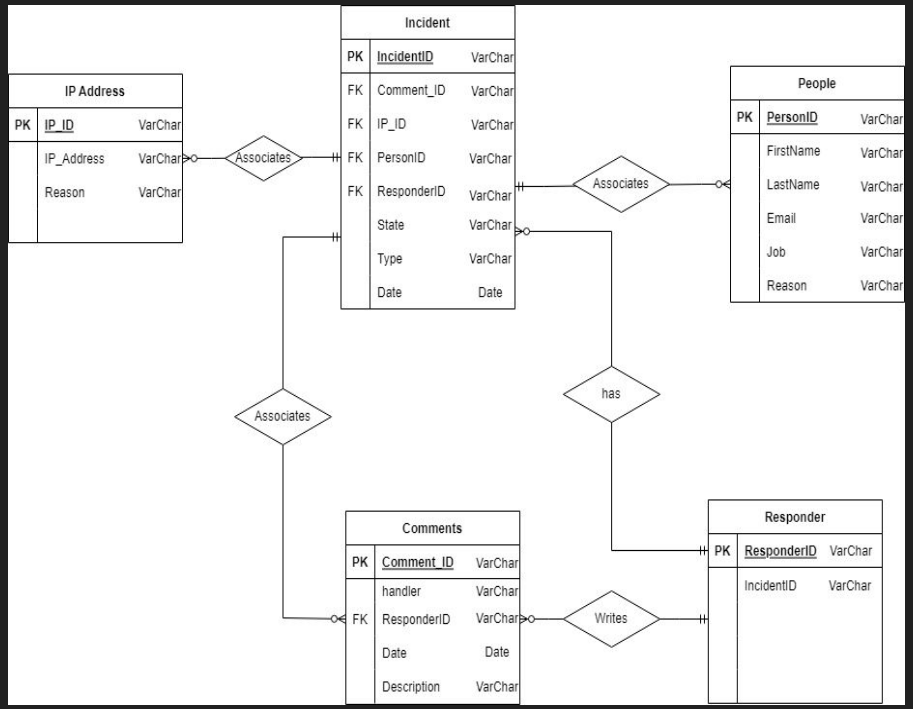
Database Management Systems Final Project

By: Nicholas Prashad, Larry Moreno, and Jobin John

ER Diagram

The ER Diagram displays the relationships between each entity available to us based on the case study. It helps us to identify attributes and requirements for building a proper database. The second diagram expands upon the basic ER diagram with labels for the foreign and primary keys. The addition of the labeled keys combined with the relationship shown between the entities help visualize the connection between the two. The Incident entity has attributes IncidentID, Comment_ID, IP_ID, PersonID, ResponderID, State, Type Date and IncidentID is the primary key which is used to search in the database. The IncidentID acts as the unique identifier that can be used to gather all other data linked to it with a single query. This works because of the connections between the Incident entity and every other entity. Each incident has comments which are written by responders who have their own ResponderID. Each incident also has an IP Address and Person associated with it represented by the People and IP Address entities.



Normalization UNF->2NF

UNF:

- A table that contains one or more repeating groups.

INCIDENT (IncidentID, IncidentType, IncidentState, IncidentDate, ResponderID, CommentID, Handler, Date, Description, IP_ID, IP_Address, IP_Reason, PersonID, FirstName, LastName, Email, Job, Person_Reason)

1NF:

- UNF - 1NF
  - Repeating groups are removed
  - Primary key is established

INCIDENT (IncidentID, IncidentType, IncidentState, IncidentDate, ResponderID, CommentID, Handler, Date, Description, IP_ID, IP_Address, IP_Reason, PersonID, FirstName, LastName, Email, Job, Person_Reason)

2NF:

- 1NF- 2NF
  - Is in First Normal Form
  - Non key attributes are dependant on primary key (No partial dependency)

INCIDENT (IncidentID, IncidentType, IncidentState, IncidentDate, ResponderID, CommentID, IP_ID, PersonID)
PEOPLE (PersonID, FirstName, LastName, Email, Job, Reason)
COMMENTS (CommentID, Handler, Responder, Date, Description)
IP_ADDRESS (IP_ID, IP_Address, Reason)
RESPONDER (ResponderID)

Normalization 2NF->BCNF

3NF:

- 2NF- 3NF
  - Is in Second Normal Form
  - No transitive dependencies on primary key (A → B, B → C, A → C)

INCIDENT (IncidentID, IncidentType, IncidentState, IncidentDate, *ResponderID*, *CommentID*, *IP_ID*, *PersonID*)

Foreign Keys:

INCIDENT(CommentID) REFERENCES COMMENT(CommentID)
INCIDENT(ResponderID) REFERENCES RESPONDER(ResponderID)
INCIDENT(PersonID) REFERENCES PEOPLE(PersonID)
INCIDENT(IP_ID) REFERENCES IP_ADDRESS(IP_ID)

PEOPLE (PersonID, FirstName, LastName, Email, Job, Description)

COMMENTS (CommentID, handler, *ResponderID*, CommentDate, description)

Foreign Keys:

INCIDENT(ResponderID) REFERENCES RESPONDER(ResponderID)

IP_ADDRESS (IP_ID, IP_Address, IP_Reason)

RESPONDER (ResponderID, *IncidentID*)

Foreign Keys:

INCIDENT(IncidentID) REFERENCES INCIDENT(IncidentID)

BCNF:

- 3NF- BCNF
  - Is in Third Normal Form
  - Every determinant key is a candidate key

INCIDENT (IncidentID, IncidentType, IncidentState, IncidentDate, *ResponderID*, *CommentID*, *IP_ID*, *PersonID*)

Foreign Keys:

INCIDENT(CommentID) REFERENCES COMMENT(CommentID)
INCIDENT(ResponderID) REFERENCES RESPONDER(ResponderID)
INCIDENT(PersonID) REFERENCES PEOPLE(PersonID)
INCIDENT(IP_ID) REFERENCES IP_ADDRESS(IP_ID)

PEOPLE (PersonID, FirstName, LastName, Email, Job, Description)

COMMENTS (CommentID, handler, *ResponderID*, CommentDate, description)

Foreign Keys:

INCIDENT(ResponderID) REFERENCES RESPONDER(ResponderID)

IP_ADDRESS (IP_ID, IP_Address, IP_Reason)

RESPONDER (ResponderID, *IncidentID*)

Foreign Keys:

INCIDENT(IncidentID) REFERENCES INCIDENT(IncidentID)

Schema

```
CREATE TABLE People(
 personID CHAR(4) NOT NULL,
 firstName CHAR(25) NOT NULL,
 lastName CHAR(25) NOT NULL,
 email VARCHAR(100) NULL,
 job CHAR(25) NULL,
 description VARCHAR(255) NULL,
 CONSTRAINT PK_People PRIMARY KEY (personID)
);

CREATE TABLE Ip_Address(
 ipID VARCHAR(10) NOT NULL,
 ipaddress INT NULL,
 reason VARCHAR(255) NULL,
 CONSTRAINT PK_Ip_Address PRIMARY KEY (ipID)
);

CREATE TABLE Responder(
responderID CHAR(4) NOT NULL,
incidentNum CHAR(4) NOT NULL,
 CONSTRAINT PK_Responder PRIMARY KEY (responderID)
);
```

```
CREATE TABLE Comment(
commenID CHAR(4) NOT NULL,
handler CHAR(25) NULL,
description VARCHAR(255) NOT NULL,
date DATE,
responderID CHAR(4) NULL,
CONSTRAINT PK_Comment PRIMARY KEY (commenID),
CONSTRAINT FK_Responder FOREIGN KEY (responderID) REFERENCES Responder
(responderID)
);


CREATE TABLE Incident(
incidentID CHAR(4) NOT NULL,
type VARCHAR(25) NOT NULL,
state VARCHAR(25) NOT NULL,
date DATE,
responderID CHAR(4) NULL,
commenID CHAR(4) NULL,
ipID VARCHAR(10) NULL,
personID CHAR(4) NULL,
CONSTRAINT PK_Incident PRIMARY KEY (incidentID),
CONSTRAINT FK_Responded FOREIGN KEY (responderID) REFERENCES Responder
(responderID),
CONSTRAINT FK_Comment FOREIGN KEY (commenID) REFERENCES Comment
(commenID),
CONSTRAINT FK_Ip_Address FOREIGN KEY (ipID) REFERENCES Ip_Address (ipID),
CONSTRAINT FK_Person FOREIGN KEY (personID) REFERENCES People (personID),
CONSTRAINT CHK_State CHECK (state = 'open' OR state = 'closed' OR state = 'stalled')
);
```

Dbconnect.php

- File is used to establish connection to the mysql database
- Hostname, Database username, Database password, and Database name are all left blank to allow the current user to input their information.
- The connection is made using the provided information and then returned for use in other files.

```php
<!DOCTYPE html>
<html>
<head>
    <title>PHP Connect MySQL Database</title>
</head>
<body>
    <p><?php
        function connect_db()
        {
            $db_host = "localhost";   // Hostname
            $db_user = "root";  // Database username
            $db_pass = "Pr@shAd3lpH1";  // Database Password
            $db_name = "db_263_project";  // Database name

            $conn = new mysqli($db_host, $db_user, $db_pass, $db_name) or die("Connection failed: %s\n" . $conn->error);

            return $conn;
        }
    ?></p>
</body>
</html>
```

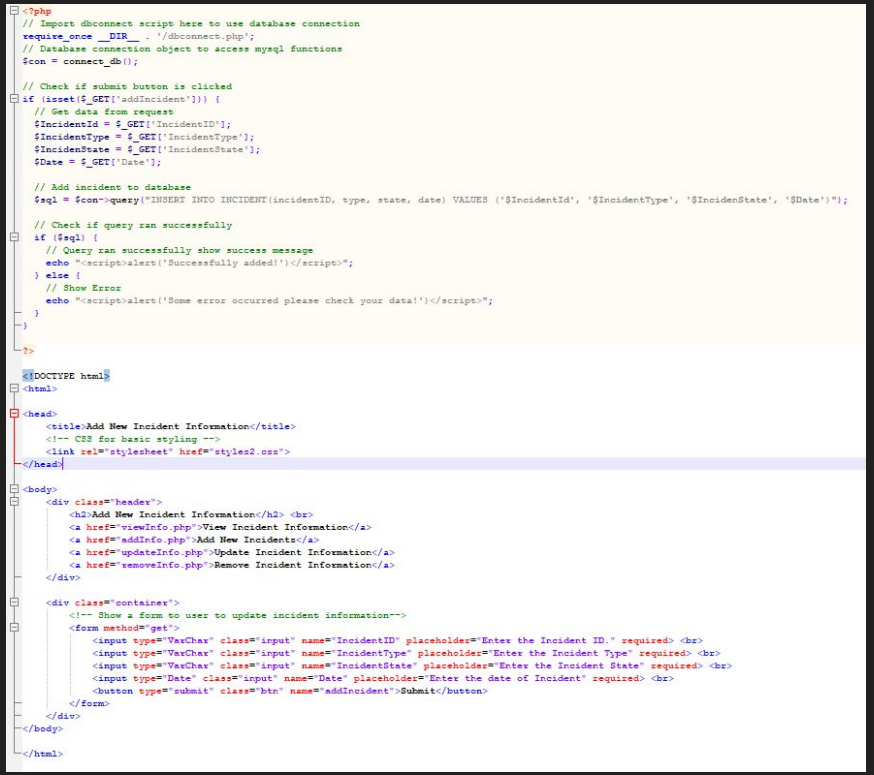
Main.php

```
<html>
  <title> Computer Security Incident Response Team </title>
  <body style="background-color:white">
  <center>
    <h1> Computer Security Incident Response Team </h1>
    <h3>  </h3>
    <p> Click on the action you would like to perform: </p>

    <p> <h4> <a href='addIncident.php'> Add New Incident Information </a> </h4> </p>

    <p> <h4> <a href='updateInfo.php'> Update Incident Information </a> </h4> </p>

    <p> <h4> <a href='RemoveInfo.php'> View Incident Information </a> </h4> </p>

    <p> <h4> <a href='viewInfo.php'> View Incident Information </a> </h4> </p>

  </center>
  </body>
</html>
```

- This is a mage page that will act as an intermediate between all other pages.
- This home page will allow a viewer to access any page needed.

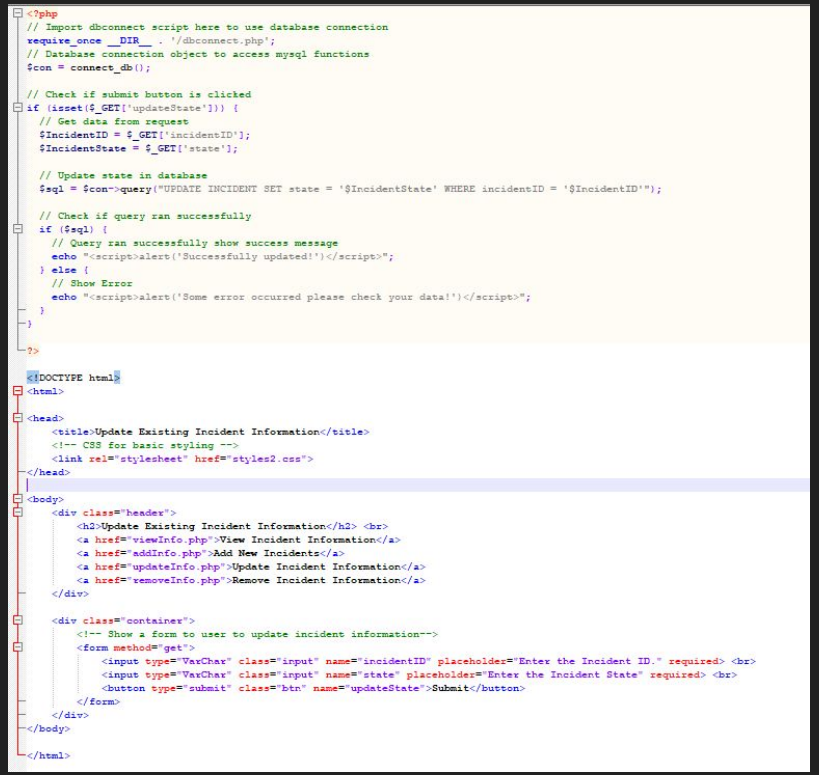
addInfo.php

- This pages allows users to create new incidents and add them to the database



```php
<?php
// Import dbconnect script here to use database connection
require_once __DIR__ . '/dbconnect.php';
// Database connection object to access mysql functions
$con = connect_db();

// Check if submit button is clicked
if (isset($_GET['addIncident'])) {
  // Get data from request
  $IncidentId = $_GET['IncidentID'];
  $IncidentType = $_GET['IncidentType'];
  $IncidentState = $_GET['IncidentState'];
  $Date = $_GET['Date'];

  // Add incident to database
  $sql = $con->query("INSERT INTO INCIDENT(incidentID, type, state, date) VALUES ('$IncidentId', '$IncidentType', '$IncidenState', '$Date')");

  // Check if query ran successfully
  if ($sql) {
    // Query ran successfully show success message
    echo "<script>alert('Successfully added!')</script>";
  } else {
    // Show Error
    echo "<script>alert('Some error occurred please check your data!')</script>";
  }
}

?>

<!DOCTYPE html>
<html>

<head>
    <title>Add New Incident Information</title>
    <!-- CSS for basic styling -->
    <link rel="stylesheet" href="styles2.css">
</head>

<body>
    <div class="header">
        <h2>Add New Incident Information</h2> <br>
        <a href="viewInfo.php">View Incident Information</a>
        <a href="addInfo.php">Add New Incidents</a>
        <a href="updateInfo.php">Update Incident Information</a>
        <a href="removeInfo.php">Remove Incident Information</a>
    </div>

    <div class="container">
        <!-- Show a form to user to update incident information-->
        <form method="get">
            <input type="VarChar" class="input" name="IncidentID" placeholder="Enter the Incident ID." required> <br>
            <input type="VarChar" class="input" name="IncidentType" placeholder="Enter the Incident Type" required> <br>
            <input type="VarChar" class="input" name="IncidentState" placeholder="Enter the Incident State" required> <br>
            <input type="Date" class="input" name="Date" placeholder="Enter the date of Incident" required> <br>
            <button type="submit" class="btn" name="addIncident">Submit</button>
        </form>
    </div>
</body>

</html>
```

updateInfo.php

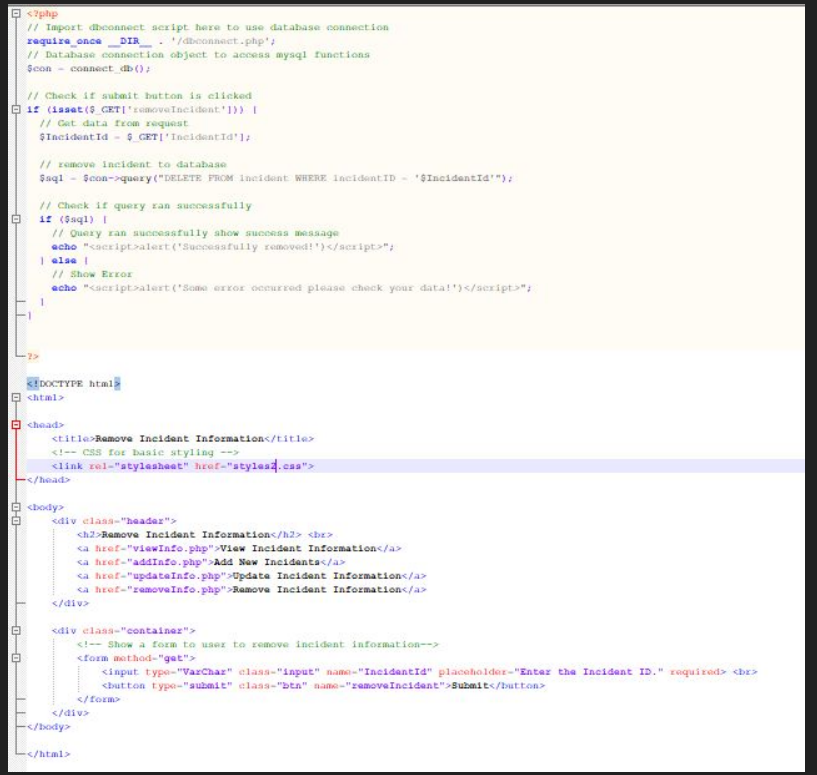


```php
<?php
// Import dbconnect script here to use database connection
require_once __DIR__ . '/dbconnect.php';
// Database connection object to access mysql functions
$con = connect_db();

// Check if submit button is clicked
if (isset($_GET['updateState'])) {
    // Get data from request
    $IncidentID = $_GET['incidentID'];
    $IncidentState = $_GET['state'];

    // Update state in database
    $sql = $con->query("UPDATE INCIDENT SET state = '$IncidentState' WHERE incidentID = '$IncidentID'");

    // Check if query ran successfully
    if ($sql) {
        // Query ran successfully show success message
        echo "<script>alert('Successfully updated!')</script>";
    } else {
        // Show Error
        echo "<script>alert('Some error occurred please check your data!')</script>";
    }
}

?>

<!DOCTYPE html>
<html>

<head>
    <title>Update Existing Incident Information</title>
    <!-- CSS for basic styling -->
    <link rel="stylesheet" href="styles2.css">
</head>

<body>
    <div class="header">
        <h2>Update Existing Incident Information</h2> <br>
        <a href="viewInfo.php">View Incident Information</a>
        <a href="addInfo.php">Add New Incidents</a>
        <a href="updateInfo.php">Update Incident Information</a>
        <a href="removeInfo.php">Remove Incident Information</a>
    </div>

    <div class="container">
        <!-- Show a form to user to update incident information-->
        <form method="get">
            <input type="VarChar" class="input" name="incidentID" placeholder="Enter the Incident ID." required> <br>
            <input type="VarChar" class="input" name="state" placeholder="Enter the Incident State" required> <br>
            <button type="submit" class="btn" name="updateState">Submit</button>
        </form>
    </div>
</body>

</html>
```

- This page allows users to edit information of current or past incidents
- Users can add new comments or change the state of the incident

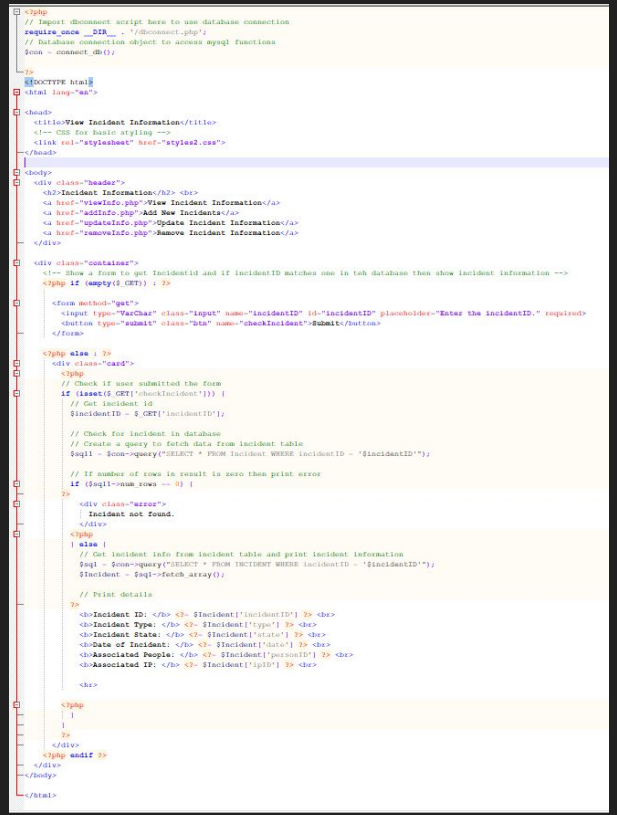
removeInfo.php

- This page allows users to remove traces of a specified incident



```php
<?php
// Import dbconnect script here to use database connection
require_once __DIR__ . '/dbconnect.php';
// Database connection object to access mysql functions
$con = connect_db();

// Check if submit button is clicked
if (isset($_GET['removeIncident'])) {
  // Get data from request
  $IncidentId = $_GET['IncidentId'];

  // remove incident to database
  $sql = $con->query("DELETE FROM incident WHERE incidentID = '$IncidentId'");

  // Check if query ran successfully
  if ($sql) {
    // Query ran successfully show success message
    echo "<script>alert('Successfully removed!')</script>";
  } else {
    // Show Error
    echo "<script>alert('Some error occurred please check your data!')</script>";
  }
}


?>

<!DOCTYPE html>
<html>

<head>
    <title>Remove Incident Information</title>
    <!-- CSS for basic styling -->
    <link rel="stylesheet" href="styles2.css">
</head>

<body>
    <div class="header">
        <h2>Remove Incident Information</h2> <br>
        <a href="viewInfo.php">View Incident Information</a>
        <a href="addInfo.php">Add New Incidents</a>
        <a href="updateInfo.php">Update Incident Information</a>
        <a href="removeInfo.php">Remove Incident Information</a>
    </div>

    <div class="container">
        <!-- Show a form to user to remove incident information-->
        <form method="get">
            <input type="VarChar" class="input" name="IncidentId" placeholder="Enter the Incident ID." required> <br>
            <button type="submit" class="btn" name="removeIncident">Submit</button>
        </form>
    </div>
</body>

</html>
```

viewInfo.php



```php
<?php
// Import dBconnect script here to use database connection
require_once __DIR__ . '/dbconnect.php';
// Database connection object to access mysql functions
$con = connect_db();

?>
<!DOCTYPE html>
<html lang="en">

<head>
  <title>View Incident Information</title>
  <!-- CSS for basic styling -->
  <link rel="stylesheet" href="styles2.css">
</head>

<body>
  <div class="header">
    <h2>Incident Information</h2> <br>
    <a href="viewInfo.php">View Incident Information</a>
    <a href="addInfo.php">Add New Incidents</a>
    <a href="updateInfo.php">Update Incident Information</a>
    <a href="removeInfo.php">Remove Incident Information</a>
  </div>

  <div class="container">
    <!-- Show a form to get IncidentId and if incidentID matches one in teh database then show incident information -->
    <?php if (empty($_GET)) : ?>

      <form method="get">
        <input type="VarChar" class="input" name="incidentID" id="incidentID" placeholder="Enter the incidentID." required>
        <button type="submit" class="btn" name="checkIncident">Submit</button>
      </form>

    <?php else : ?>
      <div class="card">
        <?php
        // Check if user submitted the form
        if (isset($_GET['checkIncident'])) {
          // Get incident id
          $incidentID = $_GET['incidentID'];

          // Check for incident in database
          // Create a query to fetch data from incident table
          $sql1 = $con->query("SELECT * FROM Incident WHERE incidentID = '$incidentID'");

          // If number of rows in result is zero then print error
          if ($sql1->num_rows == 0) {
        ?>
            <div class="error">
              Incident not found.
            </div>
          <?php
          } else {
            // Get incident info from incident table and print incident information
            $sql = $con->query("SELECT * FROM INCIDENT WHERE IncidentID = '$incidentID'");
            $Incident = $sql->fetch_array();

            // Print details
          ?>
            <b>Incident ID: </b> <?= $Incident['incidentID'] ?> <br>
            <b>Incident Type: </b> <?= $Incident['type'] ?> <br>
            <b>Incident State: </b> <?= $Incident['state'] ?> <br>
            <b>Date of Incident: </b> <?= $Incident['date'] ?> <br>
            <b>Associated People: </b> <?= $Incident['personID'] ?> <br>
            <b>Associated IP: </b> <?= $Incident['ipID'] ?> <br>

            <hr>

          <?php
          }
        }
        ?>
      </div>
    <?php endif ?>
  </div>
</body>

</html>
```

- The page allows users to query the database and retrieve all data around the chosen incident

styles.css

- Provides the colors and font for all parts of the website

```css
body {
  margin: 0px;
  font-family: Helvetica;
}

.header {
  background-color: gray;
  padding: 8px;
  color: black;
  align-self: center;
  overflow: hidden;
  text-align: center;
}

.container {
  width: 80%;
  margin: auto;
  padding: 20px;
  text-align: center;
}

.card {
  margin: auto;
  border: 1px gray;
  border-radius: 4px;
  padding: 20px 10px;
  text-align: center;
}

.input {
  margin-top: 4px;
  padding: 8px;
}

.btn {
  margin-top: 4px;
  padding: 8px;
  background-color: gray;
  color: black;
  border: 2px black;
  border-radius: 4px;
  cursor: pointer;
}

table {
  margin-left: auto;
  margin-right: auto;
}
```