

Criterion C: Development

Word count excluding code, bulleted lists, titles and works cited: 1128

Note: code taken from external sources will have a commented link above the section of code.

Introduction:

The program I have written is called the Golfasian Inquiry Form. This program allows potential customers of Golfasian (the user) to choose their preferences and create a holiday inquiry. This data is then, inserted into a database, emailed to the user, emailed to an email account accessible by a Golfasian employee, and be available to download by a Golfasian employee.

Summary List of All Techniques:

- Form submission (\$_POST)
- Usage of APCu function
- User-defined objects made from an OOP "template" class
- Autoloading classes
- Encapsulation of private methods that work on public attribute of a "template" class
- Storing data in a MySQL database
- Fetching data from a MySQL database based on user inputs
- Using prepared SQL statements for queries and inserting data
- Usage of OOP to enhance security in terms of the MySQL database connection
- Usage of PHPMailer library
- Usage of IPInfoDB's IP Geolocation API
- Use of PhpSpreadsheet library
- For-loops and foreach-loops
- Nested for-loops and foreach-loops
- While-loops
- Arrays, 2d arrays
- Input checking
- Error checking
- Dynamically defining required inputs
- Dynamic GUI
- GUI Pop-up alerts
- Simple and compound selection (if/else)
- Switch statement
- Making an array of objects
- Usage of PHP to construct HTML code
- Exploding and imploding for arrays
- Finding the number of days between two dates
- Usage of defining NULL variables

Structure of the program:

The Golfasian Inquiry Form inquiry form works in a linear progression where the intended user, goes through three different pages to input the inquiry preferences. A summary page will then show all of the inputted information. Once the user hits submit, the program will then send an email to the user with the same summary information. Alongside this, the program will send an email to an email address intended for a Golfasian employee containing the same summary information along with two links, one leading to a download to an excel file containing the individual customer's inquiry, and the other to a download for an excel file containing every customer's inquiry and information (same as the database).

Class files are indicated by the file name ending with *.class.php* and are included in other files where needed using the PHP class autoload register function. ("PHP: Autoloading Classes - Manual.")

```
spl_autoload_register(function ($class_name) {  
    include $class_name . '.class.php';  
});
```

Complexities:

Form submissions (\$_POST)

"The \$_POST variable is an array of variable names and values sent by the HTTP POST method." from an HTML form ("PHP POST"). The HTTP method is defined in the HTML <form> tag where the `method="POST"` for the customer info page (index.html), the destination selection page (destinationpage.html), and the hotel and golf course selection page (hotelandgolfselection.html). The POST method allows for the sending of the form data from the user input to be collected in the next page using the \$_POST variable in storeCustomerInfo.html, storeDestination.html and storeHotelAndGolfSelection.html pages respectively. This data invisible to users.

Index.html

```
<!DOCTYPE html>  
<head>  
    <link href="style.css" rel="stylesheet">  
</head>  
<h1>Golfasian Inquiry Form</h1>  
<body>  
<form action="storeCustomerInfo.html" method="POST">  
    <div class="container">  
        <!--name-->  
        Name:<br>  
        <label for="fname"></label>  
        <input placeholder="First name" type="text" id="fname"
```

```

name="fname" required>
  <label for="mname"></label>
  <input placeholder="Middle name" type="text" id="mname"
name="mname">
  <label for="lname"></label>
  <input placeholder="Last name" type="text" id="lname"
name="lname" required><br>
  <p>
    <!--email-->
    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" required><br>
  </p><p>
    <label for="phone">Phone number:</label><br>
    <label for="countryCode"></label>
    +
    <input type="number" id="countryCode" name="countryCode"
min=1 max=999 required>
    <input type="tel" id="phone" name="phone" required><br>
  </p><p>
    <!--countries taken from:
https://www.technicalkeeda.com/html-tutorials/all-countries-drop-down-li
st-in-html-->
    <label for="nationality">Nationality:</label><br>
    <select id="nationality" name="nationality" required>...
    </select>
  </p><p>
    <label for="arrivalDate">Arrival Date:</label><br>
    <input type="date" id="arrivalDate" name="arrivalDate"
required>
  </p><p>
    <label for="departureDate">Departure Date:</label><br>
    <input type="date" id="departureDate" name="departureDate"
required>
  </p><p>
    <label for="hotelStandard">Hotel Standard:</label>
    <select id="hotelStandard" name="hotelStandard" required>
      <option value="any">any</option>
      <option value="3">3 star</option>
      <option value="4">4 star</option>
      <option value="5">5 star</option>
    </select>

```

```
</p>
  <input type="submit" name="submit" value="Next Page">
  <input type="reset">
</div>
</form>
</body>
</html>
```

Enforcing required inputs

Certain inputs that are required in order to proceed to the next page can be forced to be required to be input by adding the `required` attribute as seen above in `index.html`.

For the hotel and golf selection page, the inputs that have the required attribute needs to be dynamically changed. More specifically, if a user were to choose a golf course from the drop-down, the input teetime would need to be set to required. Upon clicking on the golf course selection box, the function `setVisible()` is called.

hotelandgolfselection.html

```
<script type="text/javascript">
  function setVisible(golfCourse, teeTime, teeTimeLabel, caddie, cart){
    var golfCourse = document.getElementById(golfCourse);
    var teeTime = document.getElementById(teeTime);
    var teeTimeLabel = document.getElementById(teeTimeLabel);
    var caddie = document.getElementById(caddie);
    var cart = document.getElementById(cart);
    if(golfCourse.value != "")
    {
      teeTimeLabel.style.display = '';
      teeTime.style.display = '';
      caddie.style.display = '';
      cart.style.display = '';
      teeTime.setAttribute('required', '');
    }else{
      teeTimeLabel.style.display = 'none';
      teeTime.style.display = 'none';
      caddie.style.display = 'none';
      cart.style.display = 'none';
      teeTime.removeAttribute('required');
    }
  }
}
```

```
</script>
```

Incorrect input checking

As a part of the testing plan, each input needs to be the correct type in order for the user to go to the next page. There are different ways that this is done:

Firstly checking the name and the phone number input using `preg_match()`. This function performs regular expression match that searches the string for a pattern and returns true if the pattern exists, otherwise it returns false. (“PHP Preg_match() Function”)

storeCustomerInfo.html

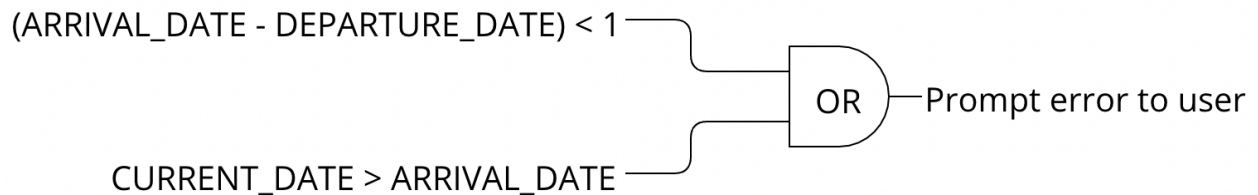
```
//concatonating all the names into one string in order to do only one
check for all the names
$name = $firstName.' '.$middleName.' '.$lastName;
//https://www.w3schools.com/php/php_form_url_email.asp
if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
    exit('<script>alert("Invalid Name, please try again");
history.back()</script>');
}
```

```
//taken from
https://www.abstractapi.com/guides/php-validate-phone-number
//preg_match /^[0-9]{8,12}$/ will check if all the characters are
integers (between 0 and 9) and if there are between 8 and 12 characters
if(!preg_match('/^[0-9]{8,12}$/', $_POST['phone'])) {
    exit('<script>alert("Phone number must only contain numbers and be
bewteen 8 to 12 digits"); history.back()</script>');
}
```

Secondly, using `filter_var()` function to filter and validate the email (checking for an @ symbol followed by a website address).

```
$email = filter_var($email, FILTER_SANITIZE_EMAIL);
if(!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    exit('<script>alert("Invalid Email, please try again");
history.back()</script>');
}
```

Thirdly, in order to compare and check the dates input, two functions must be used. First, parsing the time input to a standardized unix time stamp using `strtotime()`. Then, comparing the dates by checking two things in a conditional statement represented in the following diagram. (“PHP: Strtotime - Manual”)



```
//first or: departure minus the arrival date, if it equates to 0 days or  
a negative number (less than 1), it is invalid  
//second or: checking if the current date is greater than (after) the  
departure date, meaning it is invalid  
if((strtotime($_POST['departureDate']) -  
strtotime($_POST['arrivalDate'])) < 1 || (strtotime(date("Y-m-d")) >  
strtotime($_POST['arrivalDate']))) {  
    exit('<script>alert("Invalid dates, please try again");  
    history.back()</script>');  
}
```

The departure date should not be before the arrival date and the arrival date should not be before the current date. The current date is also retrieved from the `date()` function and also converted with `strtotime()`.

User-defined objects made from an OOP "template" class

From these inputs, the data is then used to instantiate objects from OOP “template” classes. The “template” classes are Customer, DayPlan, Inquiry and Destination. The object properties are initialized upon creation of the object using the constructor method `public function __construct()`.

storeCustomerInfo.html

```
$customer = new Customer($firstName, $middleName, $lastName, $email,  
$phoneNumber, $nationality, $IPAddress);
```

Customer.class.php

```
public function __construct($firstName, $middleName, $lastName, $email,  
$phoneNumber, $nationality, $IPAddress){  
    $this->firstName = $firstName;
```

```

$this->middleName = $middleName;
$this->lastName = $lastName;
$this->email = $email;
$this->phoneNumber = $phoneNumber;
$this->nationality = $nationality;
$this->location = $this->setLocation($IPAddress);
}

```

Moreover, inquiry **has a** customer, destination and multiple dayplan classes (**aggregation**). Multiple dayplan classes are stored into an array of objects `$dayPlans`.

Overall, using OOP allows for the program to be more easily debugged, reusable, visualized, managed, and extensible, with relationships between the classes making sense in a logical manner.

Usage of APCu function

As seen with the structure of the program, many different files are used, which makes it difficult to access the user-defined objects instantiated on another file. To solve this issue, the objects are stored in the user cache using the APCu function. Objects are stored with a key value, with the function `apcu_store()`. The key-value is then used as a parameter in `apcu_fetch()` to retrieve the object. (“PHP: APCu - Manual”)

storeCustomerInfo.html

```

//storing the user-defined objects in the CPU to be persistent
apcu_store('customer', $customer);
apcu_store('inquiry', $inquiry);

```

storeDestination.html

```

apcu_store('destination', $destination);

```

storeHotelAndGolfSelection.html

```

apcu_store('dayplans', $dayPlans);

```

submitpage.html

```

$destination = apcu_fetch('destination');
$inquiry = apcu_fetch('inquiry');
$customer = apcu_fetch('customer');
$dayPlans = apcu_fetch('dayplans');

```

To prevent a memory leak, the objects stored in the user cache are deleted at the end of the program using `apcu_delete();`.

thankyou.html

```
apcu_delete('inquiry');
apcu_delete('customer');
apcu_delete('destination');
apcu_delete('dayplans');
```

Using OOP to enhance security with MySQL connection

As a part of Criteria 1 and following criteria, a MySQL database is created and connected to. OOP logic is used with the MySQL connection `mysqli` object. OOP is used as the log-in information such as the log-in and password to the MySQL database can be private attributes, and the only way to access the `mysqli` object is by using a `protected function` that only allows the class itself and child classes to access the function. Child classes **inherit** the `MySQLConn` and are identified by `extends MySQLConn`.

MySQLConn.class.php

```
class MySQLConn {
    private $dns = 'localhost';
    private $user = ' ';
    private $pass = ' ';
    private $dbname = 'Golfasian';
    //only this class and child classes can use this method to connect to
    the mysql database
    protected function connect() {
        $conn = new mysqli($this->dns,$this->user,$this->pass,$this->dbname);

        if ($conn->connect_error) {
            return "Connection failed: ".$conn->connect_error;
        }else{
            return $conn;
        }
    }
}
```

Using prepared statements to make SQL queries.

As a part of criteria 8, data is uploaded/logged to the database. Prepared statements are used opposed to other methods such as concatenating a string for the MySQL query to prevent MySQL injection. ("Mysqli SELECT Query with Prepared Statements")

GetMySQLData.class.php

```
require_once 'MySQLConn.class.php';
class GetMySQLData extends MySQLConn {
    public function getHotelData($destination, $rating){
        //prepared sql statements
        //from https://phpdelusions.net/mysqli_examples/prepared_select
        $conn = $this->connect();
        if($rating == "any"){ //creating template
            $sql = "SELECT HotelName FROM hotel_data_2022 WHERE City=?
ORDER BY HotelName ASC;";
            $stmt = $conn->prepare($sql);
            $stmt->bind_param("s", $destination);
        }else{
            $sql = "SELECT HotelName FROM hotel_data_2022 WHERE City=?
AND Stars=? ORDER BY HotelName ASC;";
            $stmt = $conn->prepare($sql);
            $stmt->bind_param("ss", $destination, $rating);
        }
        $stmt->execute();
        $result = $stmt->get_result();
        $data = $result->fetch_all(MYSQLI_ASSOC);
        $numRows = $result->num_rows;
        if ($numRows > 0){
            return $data;
        }else{
            return "No results";
        }
    }
    public function getGolfCourseData($destination){
        //prepared sql statements
        //from https://phpdelusions.net/mysqli_examples/prepared_select
        $conn = $this->connect();
        $sql = "SELECT GolfCourseName FROM golf_course_data_2022 WHERE
City=? ORDER BY GolfCourseName ASC;";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("s", $destination);
        $stmt->execute();
        $result = $stmt->get_result();
        $data = $result->fetch_all(MYSQLI_ASSOC);
        $numRows = $result->num_rows;
        if ($numRows > 0){
```

```

        return $data;
    }else{
        return "No results";
    }
}
public function getTableData($table){
    $conn = $this->connect();
    $query = $conn->query("SELECT * FROM ".$table);
    return $query;
}

public function getTableDataFromInquiryID($table, $inquiryID){
    $conn = $this->connect();
    $sql = "SELECT * FROM ".$table." WHERE InquiryID = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $inquiryID);
    $stmt->execute();
    $result = $stmt->get_result();
    return $result;
}
public function getColumns($table){
    $conn = $this->connect();
    $sqlColumns = "SHOW COLUMNS FROM ".$table;
    $columnsQuery = $conn->query($sqlColumns);
    return $columnsQuery;
}
}
}

```

InsertData.class.php

```

require_once 'MySQLConn.class.php';

class InsertData extends MySQLConn {
    private $conn;
    public function __construct(){
        $this->conn = $this->connect();
    }
    public function insertInquiryData($inquiry, $destination){
        $destinations = $destination->getChosenDestination();
    }
}

```

```

$arrivalDate = $inquiry->getArrivalDate();
$departureDate = $inquiry->getDepartureDate();
$days = $inquiry->getNumberOfDays();
$night = $inquiry->getNumberOfNights();
$sql = "INSERT INTO inquiry (Destinations, ArrivalDate,
DepartureDate, Days, Nights) VALUES (?, ?, ?, ?, ?)";
$stmt = $this->conn->prepare($sql);
$stmt->bind_param("sssss", $destinations, $arrivalDate,
$departureDate, $days, $night);
if($stmt->execute()){
    return $stmt->insert_id;
}else{
    return false;
}
}

public function insertCustomerData($inquiryID, $customer){
    $firstName = $customer->getFirstName();
    $middleName = $customer->getMiddleName();
    $lastName = $customer->getLastName();
    $email = $customer->getEmail();
    $phoneNumber = $customer->getPhoneNumber();
    $nationality = $customer->getNationality();
    $location = $customer->getLocation();
    $sql = "INSERT INTO customers (InquiryID, FirstName, MiddleName,
LastName, Email, PhoneNumber, Nationality, Location) VALUES (?, ?, ?, ?,
?, ?, ?, ?)";
    $stmt = $this->conn->prepare($sql);
    $stmt->bind_param("sssssss", $inquiryID, $firstName, $middleName,
$lastName, $email, $phoneNumber, $nationality, $location);
    if($stmt->execute()){
        return true;
    }else{
        return false;
    }
}

public function insertDailyPlanData($inquiryID, $dailyPlan,
$dayNumber){
    $golfCourse = $dailyPlan->getGolfCourse();
    $teeTime = $dailyPlan->getTeeTime();
    $caddie = $dailyPlan->getCaddie();
    $cart = $dailyPlan->getCart();
}

```

```

    $hotel = $dailyPlan->getHotel();
    $hotelBreakfast = $dailyPlan->getHotelBreakfast();
    $transportation = $dailyPlan->getTransportation();
    $sql = "INSERT INTO day_plans (InquiryID, DayNumber, GolfCourse,
TeeTime, Caddie, Cart, Hotel, HotelBreakfast, Transportation) VALUES
((?), ?, ?, ?, ?, ?, ?, ?, ?)";
    $stmt = $this->conn->prepare($sql);
    $stmt->bind_param("ssssssss", $inquiryID, $dayNumber,
$golfCourse, $teeTime, $caddie, $cart, $hotel, $hotelBreakfast,
$transportation);
    if($stmt->execute()){
        return true;
    }else{
        return false;
    }
}
}
}

```

Use of PHPMailer Library

PHPMailer was essential in order to send an email to the user's email and to the golfasian's representative email as described by criteria 8. PHPMailer is most useful due to its ease of use various types of email hosting through a SMTP server. The code below was adapted from the PHP Mailer GitHub page and the netcorecloud tutorial. (PHPMailer)

thankyou.html

```

//first object/iteration is sending mail to the customer, second
object/iteration is sending to the golfasian employee email
for($j = 0; $j < 2; $j++){
//php mailer: https://github.com/PHPMailer/PHPMailer
    //Mailing host info

    $phpmailer = new PHPMailer();
    $phpmailer->isSMTP();
    $phpmailer->Mailer = "smtp";
    $phpmailer->SMTPAuth = true;

    //from
https://netcorecloud.com/tutorials/send-an-email-via-gmail-smtp-server-u
sing-php/
    //Change info below to golfasian host email

```

```

    $phpmailer->SMTPSecure = "tls";
    $phpmailer->Port = 587;
    $phpmailer->Host = "smtp.gmail.com";
    $phpmailer->Username = "ibcomputerscienceia14@gmail.com"; //example
    email for prototype demonstration
    $phpmailer->Password = "Computersciencepass1";

    $phpmailer->setFrom('ibcomputerscienceia14@gmail.com', 'Golfasian');
    $phpmailer->addReplyTo('info@golfasian.com', 'Golfasian');
    if($j == 0){
        $phpmailer->addAddress($customer->getEmail(),
    $customer->getFullName()); //customer email
    }else{
        $phpmailer->addAddress('ibcomputerscienceia14@gmail.com');
    //golfasian employee email
    }

    $phpmailer->isHTML(true);
    $phpmailer->Subject = 'Golfasian Inquiry for
    '.$customer->getFullName(). ' ID: '.$lastInquiryID;

    //storing the HTML as an object to be sent as the email body
    ob_start(); ?>
    <!DOCTYPE html>
    <div class="container">
        <?php
        if($j == 1){ ?>
            <p>
                <a
    href="localhost/CS_IA/inquiryExcelDownload.html?inquiryid=<?php echo
    $lastInquiryID; ?>" target="_blank" style="font-size: 22px">Download
    inquiry to excel file</a> <br>
                <a href="localhost/CS_IA/entireExcelDownload.html"
    target="_blank" style="font-size: 22px">Download entire database to
    excel file</a>
            </p>

            <?php } ?>
            <p>
                <h4>Customer and Contact Information:</h4>
                <hr>

```

```

<table style="border-spacing: auto; width: 100%; text-align:
center;">
  <tr>
    <th>InquiryID</th>
    <th>Name</th>
    <th>Email</th>
    <th>Phone Number</th>
    <th>Nationality</th>
  </tr>
  <tr>
    <td> <?php echo $lastInquiryID; ?> </td>
    <td> <?php echo $customer->getFirstName()."
".$customer->getMiddlename()." ".$customer->getLastName(); ?> </td>
    <td> <?php echo $customer->getEmail(); ?> </td>
    <td> <?php echo $customer->getPhoneNumber(); ?> </td>
    <td> <?php echo $customer->getNationality(); ?> </td>
  </tr>
</table>
<hr>
</p><p>
<h4>Trip Details:</h4>
<hr>
<table style="border-spacing: auto; width: 100%; text-align:
center;">
  <tr>
    <th>Destination(s)</th>
    <th>Arrival</th>
    <th>Departure</th>
    <th>Period</th>
    <th>Hotel Standard</th>
  </tr>
  <tr>
    <td> <?php echo $destination->getChosenDestination(); ?>
</td>
    <td> <?php echo $inquiry->getArrivalDate(); ?> </td>
    <td> <?php echo $inquiry->getDepartureDate(); ?> </td>
    <td> <?php echo $inquiry->getNumberOfDays(); ?> days,
<?php echo $inquiry->getNumberOfNights() ?> nights</td>
    <td> <?php echo $inquiry->getHotelRating()." stars"; ?>
</td>
  </tr>
</table>

```

```

</table>
<hr>
</p><p>
<h4>Day by Day Details:</h4>
<table style="border-spacing: auto; width: 100%; text-align:
center;">
<hr>
<tr>
<th>Day Number</th>
<th>Hotel</th>
<th>Hotel Breakfast</th>
<th>Golf</th>
<th>Requested Tee Time</th>
<th>Cart</th>
<th>Caddie</th>
<th>Transportation Required</th>
</tr>
<?php for($i = 0; $i < $inquiry->getNumberOfDays(); $i++){
    if($dayPlans[$i]->getHotelBreakfast()){
        $hotelBreakfast = "Yes";
    }else{
        $hotelBreakfast = "None";
    }
    if($dayPlans[$i]->getGolfCourse() != "No golf day"){
        if($dayPlans[$i]->getCart()){
            $cart = "Yes";
        }else{
            $cart = "None";
        }
        if($dayPlans[$i]->getCaddie()){
            $caddie = "Yes";
        }else{
            $caddie = "None";
        }
    }else{
        $cart = NULL;
        $caddie = NULL;
    }
    if($dayPlans[$i]->getTransportation()){
        $transportation = "Yes";
    }else{

```

```

        $transportation = "No";
    }
    ?>
    <tr>
    <td> <?php echo $i+1; ?> </td>
    <td> <?php echo $dayPlans[$i]->getHotel(); ?> </td>
    <td> <?php echo $hotelBreakfast; ?> </td>
    <td> <?php echo $dayPlans[$i]->getGolfCourse(); ?> </td>
    <td> <?php echo $dayPlans[$i]->getTeeTime(); ?> </td>
    <td> <?php echo $cart; ?> </td>
    <td> <?php echo $caddie; ?> </td>
    <td> <?php echo $transportation; ?> </td>
    </tr>
    <?php } ?>
    </table>
    </p>
    <hr>
</html>
<?php $customerMailContent = ob_get_clean();

$mailer->Body = $customerMailContent;

if(!$mailer->send()){
    exit('<script>alert("Invalid Email, please try again. \r\n Mailer
Error: ' . $mailer->ErrorInfo.'"); history.back()</script>');
}
}

```

Use of IPInfoDB's IP Geolocation API

In order to convert an IP address to a country or location, as outlined in criteria 6, an external API was used. IPInfoDB's IP Geolocation API was selected due to its compatibility with IPv4 and IPv6 and being a free to use API. A PHP class for the API which was installed. The argument sent when instantiating a class is a specific API key which was obtained when creating an account to use the API. (IPInfoDB)

Customer.class.php

```

public function setLocation($IPAddress) { //obtaining the location from
the IP address
    $ipinfodb = new
    \IPInfoDB\Api('c9244e50668e8e65273036bdb95362187a2db11e9cb9eda27e3d4f5a7
1b4d48f');
}

```



```

$result = $ipinfodb->getCity($IPAdress);

    if ($result) {
        return $result['cityName'].", ".$result['regionName'].",
".$result['countryName'];
    }
}

```

Use of PhpSpreadsheet library

For criteria 9, the PhpSpreadsheet library was used in order to easily create a downloadable spreadsheet from the MySQL result data. ("PhpSpreadsheet Documentation.")

entireExcelDownload.html

```

<?php
require_once "vendor/autoload.php";

//importing phpspreadsheet library
use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

spl_autoload_register(function ($class_name) { //
https://www.php.net/manual/en/language.oop5.autoload.php
    include $class_name . '.class.php';
});

//adapted from the phpspreadsheet documentation
https://github.com/PHPOffice/PhpSpreadsheet
$sqlData = new GetMySQLData();
$spreadsheet = new Spreadsheet();
$Excel_writer = new Xlsx($spreadsheet);

//iterating 3 times to create 3 worksheets to represent the 3 tables in
MySQL
for($l = 0; $l < 3; $l++){
    //setting the active spreadsheet, starting at the first one at index
    0
    $spreadsheet->setActiveSheetIndex($l);
    $activeSheet = $spreadsheet->getActiveSheet();

```

```

switch($1){
  case 0:
    $table = 'inquiry';
    break;
  case 1:
    $table = 'customers';
    break;
  case 2:
    $table = 'day_plans';
    break;
}
//setting the title
$activeSheet->setTitle($table);
$columnsQuery = $sqlData->getColumns($table);
//the MySQL cols result is turned into an associative array and is
used to populate the first row cells for the excel worksheet
if($columnsQuery->num_rows > 0){
  $column = 'A';
  while($row = $columnsQuery->fetch_assoc()) {
    $activeSheet->setCellValue($column."1" , $row['Field']);
    $fields[] = $row['Field'];
    $column++;
  }
}
//the data within the MySQL table is returned as an associative array
and is then used to populate cells in the worksheet
$query = $sqlData->getTableData($table);
if($query->num_rows > 0){
  $i = 2;
  while($row = $query->fetch_assoc()){
    $column = 'A';
    for($j = 0; $j < count($fields); $j++){
      $activeSheet->setCellValue($column.$i ,
$row[$fields[$j]]);
      $column++;
    }
    $i++;
  }
  unset($fields);
}
//this if statement ensures that an extra worksheet is not made since

```

when creating the spreadsheet, a worksheet is already made.

```
    if($1 != 2){
        $spreadsheet->createSheet();
    }
}
//naming the file
$filename = 'entire_database_'.date("Y-m-d").'.xlsx';

//making this html file to start a download when opened
header('Content-Type: application/vnd.ms-excel');
header('Content-Disposition: attachment;filename='. $filename);
header('Cache-Control: max-age=0');
$Excel_writer->save('php://output');

?>
```

Graphical User Interface (GUI) Work

The majority of the GUI was constructed using HTML and CSS. This was done through CSS and Javascript were used to make the web program dynamic (the web page does not need to reload) and provides the user with a modern interactive experience.

Elements and techniques used:

- Headings
- HTML form
- Various types of input
- Div to create a container
- Using class and id identifiers on elements in order to target and style elements with CSS
- Fieldsets
- Tables
- Radio boxes
- Buttons

index.html

Golfasian Inquiry Form

The image shows a web form titled "Golfasian Inquiry Form". It contains several input fields: "Name:" with sub-fields for "First name", "Middle name", and "Last name"; "Email:"; "Phone number:" with a "+" sign; "Nationality:" with a dropdown menu labeled "Select a country..."; "Arrival Date:" with a date picker showing "mm/dd/yyyy"; "Departure Date:" with a date picker showing "mm/dd/yyyy"; and "Hotel Standard:" with a dropdown menu showing "any". At the bottom, there are two buttons: a red "Reset" button and a green "Next Page" button.

When clicking into one the input fields, a light blue background will be cast in the field along with a blue border as seen in the image above.

stylesheet.css

```
input:focus {  
  outline-color: #015fcc;  
  background-color: lightblue;  
}
```

Furthermore, when hovering over the buttons, they will change color to a slightly darker color.

```
input[type=submit]:hover{
  background-color: #45a049;
}

input[type=reset]:hover{
  background-color: #cc0404;
}
```

Fieldsets are used to distinguish between different parts of input/information in destinationpage.html and hotelandgolfselection.html.

destinationpage.html

Golfasian Inquiry Form

Destinations:						
Thailand						
<input type="checkbox"/> Bangkok	<input type="checkbox"/> Hua Hin	<input type="checkbox"/> Pattaya	<input type="checkbox"/> Chiang Mai	<input type="checkbox"/> Chiang Rai	<input type="checkbox"/> Kanchanaburi	<input type="checkbox"/> Khao Yai
<input type="checkbox"/> Khon Kaen	<input type="checkbox"/> Phang Nga	<input type="checkbox"/> Phuket	<input type="checkbox"/> Samui			
Vietnam						
<input type="checkbox"/> Danang / Hoi An	<input type="checkbox"/> Dalat	<input type="checkbox"/> Hai Phone / Halong Bay	<input type="checkbox"/> Hanoi	<input type="checkbox"/> Nha Trang	<input type="checkbox"/> Ninh Binh	<input type="checkbox"/> Phan Thiet
<input type="checkbox"/> Quang Binh	<input type="checkbox"/> Quy Nhon	<input type="checkbox"/> Saigon	<input type="checkbox"/> Sapa	<input type="checkbox"/> Vinh	<input type="checkbox"/> Vung Tau	
Malaysia						
<input type="checkbox"/> Johor	<input type="checkbox"/> Ipoh Cameron Highlands	<input type="checkbox"/> Kuching	<input type="checkbox"/> Desaru	<input type="checkbox"/> Langkawi	<input type="checkbox"/> Malacca	<input type="checkbox"/> Kuala Lumpur
<input type="checkbox"/> Penang						
Cambodia						
<input type="checkbox"/> Phnom Penh	<input type="checkbox"/> Siem Reap	<input type="checkbox"/> Sihanoukville				
Japan						
<input type="checkbox"/> Chiba	<input type="checkbox"/> Hokkaido	<input type="checkbox"/> Hyogo	<input type="checkbox"/> Ibaraki	<input type="checkbox"/> Kagoshima	<input type="checkbox"/> Kanagawa	<input type="checkbox"/> Mie
<input type="checkbox"/> Nagano	<input type="checkbox"/> Tochigi	<input type="checkbox"/> Tokyo	<input type="checkbox"/> Saitama	<input type="checkbox"/> Shizuoka		
Myanmar						
<input type="checkbox"/> Taunggyi	<input type="checkbox"/> Mandalay	<input type="checkbox"/> Inle	<input type="checkbox"/> Lake Bagan	<input type="checkbox"/> Yangon		
Laos						
<input type="checkbox"/> Luang Prabang	<input type="checkbox"/> Vientiane					
Sri Lanka						
<input type="checkbox"/> Colombo	<input type="checkbox"/> Hambantota	<input type="checkbox"/> Kandy	<input type="checkbox"/> Nuwara	<input type="checkbox"/> Eliya		
UAE						
<input type="checkbox"/> Abu Dhabi	<input type="checkbox"/> Dubai					
Other Countries						
<input type="checkbox"/> China	<input type="checkbox"/> Singapore					

Back Next Page

hotelandgolfselection.html

Golf and Hotel Daily Planner

Destination(s)	Arrival	Departure	Period	Hotel Standard
Bangkok	2022-04-27	2022-04-29	3 days, 2 nights	any stars
Day 1 Note: Arrival Day Golfcourse: No golf day Hotel: Select hotel <input type="checkbox"/> Hotel Breakfast Transportation required <input type="checkbox"/>				
Day 2 Golfcourse: No golf day Hotel: Select hotel <input type="checkbox"/> Hotel Breakfast Transportation required <input type="checkbox"/>				
Day 3 Note: Departure Day Golfcourse: No golf day Transportation required <input type="checkbox"/>				

[Back](#) [Next Page](#)

Tables are used to organize data (the user input) to present to the user in hotelandgolfselection.html, submitpage.html and in the emails.

submitpage.html

Golfasian Inquiry Form



Customer and Contact Information:			
Name	Email	Phone Number	Nationality
John Doe	example@example.com	+1123456789	Austria





Trip Details:				
Destination(s)	Arrival	Departure	Period	Hotel Standard
Bangkok, Hua Hin	2022-04-27	2022-04-29	3 days, 2 nights	any

Day by Day Details:							
Day Number	Hotel	Hotel Breakfast	Golf	Requested Tee Time	Cart	Caddie	Transportation Required
1	A-One Bangkok Hotel	None	No golf day				No
2	A-One Bangkok Hotel	None	No golf day				No
3	A-One Bangkok Hotel	None	No golf day				No

[Back](#) [Submit Inquiry](#)

Example of email received by the user

Golfasian Inquiry for John Doe ID: 3 Inbox x  

 **Golfasian** <ibcomputerscienceia14@gmail.com> Sun, Apr 24, 1:03 AM (1 day ago)   
to me ▾

Customer and Contact Information:

InquiryID	Name	Email	Phone Number	Nationality
3	John Doe	[REDACTED]	+155667788	Albania



Trip Details:





Destination(s)	Arrival	Departure	Period	Hotel Standard
Hua Hin	2022-04-28	2022-04-30	3 days, 2 nights	any stars

Day by Day Details:

Day Number	Hotel	Hotel Breakfast	Golf	Requested Tee Time	Cart	Caddie	Transportation Required
1	Ace of Hua Hin Resort	None	No golf day				No
2	Ace of Hua Hin Resort	None	No golf day				No
3	Ace of Hua Hin Resort	None	No golf day				No

Example of email received by the Golfasian representative

Golfasian Inquiry for John Doe ID: 3 Inbox x  

 **Golfasian** <ibcomputerscienceia14@gmail.com> Sun, Apr 24, 1:03 AM (1 day ago)   
to me ▾

[Download inquiry to excel file](#)
[Download entire database to excel file](#)

Customer and Contact Information:

InquiryID	Name	Email	Phone Number	Nationality
3	John Doe	[REDACTED]	+155667788	Albania

Trip Details:

Destination(s)	Arrival	Departure	Period	Hotel Standard
Hua Hin	2022-04-28	2022-04-30	3 days, 2 nights	any stars

Day by Day Details:

Day Number	Hotel	Hotel Breakfast	Golf	Requested Tee Time	Cart	Caddie	Transportation Required
1	Ace of Hua Hin Resort	None	No golf day				No
2	Ace of Hua Hin Resort	None	No golf day				No
3	Ace of Hua Hin Resort	None	No golf day				No

The hyperlinks also lead to the html file that starts the download for the respective excel files.

Works Cited

IPInfoDB. "IP Info, IP Geolocation Tools and API." *IPInfoDB*, www.ipinfodb.com/. Accessed 25 Apr. 2022.

"Mysqli SELECT Query with Prepared Statements." *Treating PHP Delusions*, phpdelusions.net/mysqli_examples/prepared_select. Accessed 25 Apr. 2022.

"PHP POST." *Sinsixx.com*, 2022, w3schools.sinsixx.com/php/php_post.asp.htm. Accessed 11 Apr. 2022.

"PHP Preg_match() Function." *Www.javatpoint.com*, 2021, [www.javatpoint.com/php-preg_match-function#:~:text=The%20preg_match\(\)%20function%20is, pattern%20exists%20otherwise%20returns%20false..](http://www.javatpoint.com/php-preg_match-function#:~:text=The%20preg_match()%20function%20is, pattern%20exists%20otherwise%20returns%20false..) Accessed 20 Apr. 2022.

"PHP: APCu - Manual." *Www.php.net*, www.php.net/manual/en/book.apcu.php. Accessed 25 Apr. 2022.

"PHP: Autoloading Classes - Manual." *Www.php.net*, www.php.net/manual/en/language.oop5.autoload.php.

"PHP: Strtotime - Manual." *Php.net*, 2018, www.php.net/manual/en/function.strptime.php. Accessed 20 Apr. 2022.

PHPMailer. "GitHub - PHPMailer/PHPMailer: The Classic Email Sending Library for PHP." *GitHub*, 12 Apr. 2022, github.com/PHPMailer/PHPMailer. Accessed 20 Apr. 2022.

"PhpSpreadsheet Documentation." *Phpspreadsheet.readthedocs.io*, phpspreadsheet.readthedocs.io/en/latest/. Accessed 25 Apr. 2022.