Answer of Web Application Development Review A3

1. What are differences between Frontend and Backend?

⇨ The differences between Frontend and Backend are below:

Frontend	Backend
ជាអ្វីដែល User មើលឃើញនៅលើ Web Browser	ជាអ្វីដែល User មើលមិនឃើញនៅលើ Web Brower
ប្រើប្រាស់ HTML, CSS, and JavaScript	ប្រើប្រាស់ PHP, Python, Ruby, or ASP.Net
ពាក់ព័ន្ធនឹង Images, Graphic, Content, and Structure	ពាក់ព័ន្ធនឹង Business Logic and Data
ដំណើរការនៅលើ Client-Side	ដំណើរការនៅលើ Server និងជាមួយ Database

2. What are differences between Static Website and Dynamic Website?

⇨ The differences between Static Website and Dynamic Website are below:

Static Website	Dynamic Website
មាន Web Page ជា Static	មាន Web Page ជា Dynamic
ងាយស្រួលក្នុងការ Build	ពិបាកក្នុងការ Build
ការ Hosting មានតម្លៃថូរថ្លៃ	ការ Hosting មានតម្លៃថ្លៃ
ការអភិវឌ្ឍន៍មានតម្លៃសមរម្យ	ការអភិវឌ្ឍន៍មានតម្លៃថ្លៃ
ពិបាកក្នុងការផ្លាស់ប្តូរ និង Update	ងាយស្រួលក្នុងការផ្លាស់ប្តូរ និង Update

3. Explain about the processes of Web Client and Web Server when we open the browser.

⇨ **The processes of web client and web server when we open the browser is below:**

- ➢ A Client requests a web page from a web browser
- ➢ The web browser sends a request to the web server
- ➢ The web server receives the request and parses it
- ➢ The web server then executes the code that generates the web page
- ➢ The web server then sends the generated web page back to web browser
- ➢ The web browser then displays the web page to the client or end-user

4. What is PHP? What are the advantages and disadvantages of using PHP?

⇨ **PHP** ជា Server-side script language ដែលត្រូវបានគេប្រើប្រាស់ក្នុងការ Develop Dynamic Web Page and Dynamic Web Application។ PHP code ត្រូវ run នៅលើ server។

➢ **The Advantages and Disadvantages of using PHP are below:**

Advantages of PHP	Disadvantages of PHP
Easy to learn and use	Lack security features
Cross-platform compatibility	Limited debugging tools
Free and Open-source	Complex syntax
Support various databases	Poor performance នៅពេលដែលប្រើ features ច្រើន
Powerful and Flexible	Not recommend for Enterprise project
Large Community of developers	

5. What are the different between Web Dynamic Client and Web Dynamic Server?

⇨ **The different between Web Dynamic Client and Web Dynamic Server are below:**

Web Dynamic Client	Web Dynamic Server
Code ត្រូវបាន run នៅលើ User Computer	Code ត្រូវបាន run នៅលើ Web Server
ប្រើប្រាស់ HTML, CSS, and JavaScript	ប្រើប្រាស់ PHP, Python, Ruby, or ASP.Net
Source Code អាចមើលបាន	Source Code មិនអាចមើលបាន
Less Security	More Security
More interactive and personalized	More scalable and performant

6. What is Model View Controller (MVC)? Explain each of them.

⇨ **MVC – Model View Controller** ជាគំរូស្ថាបត្យកម្មដែលប្រើក្នុងការអភិវឌ្ឍន៍កម្មវិធី ដើម្បីធ្វើរចនាសម្ព័ន្ធមូលដ្ឋានកូដនៃកម្មវិធី។ MVC បានបែងចែកចេញជា ៣ ផ្នែកដែលមានដូចជា Model, View និង Controller។

➢ **Explain of each of them (MVC):**

- **Model:** សំដៅលើ Data and Business logic of application។ Model ពាក់ព័ន្ធនឹងការគ្រប់គ្រងទិន្នន័យ ដំណើរការលក្ខខណ្ឌអាជីវកម្ម និងដើម្បីឆ្លើយតបទៅនឹង Controller។
- **View:** សំដៅលើការបង្ហាញទិន្នន័យទៅកាន់ User។ View ពាក់ព័ន្ធនឹងការបំប្លែងទិន្នន័យដែលទទួលបានពី Model ទៅជាទម្រង់ដែលអាចអោយ User ងាយមើលងាយយល់។
- **Controller:** សំដៅលើការ Handle នូវ User Input។ Controller ពាក់ព័ន្ធនឹងការ Implement នូវ User Input ដើម្បីធ្វើអន្តកម្មរវាង View និង Model។

7. Explain the benefits of using MVC in Web Development.

⇨ **The benefits of using MVC in Web Development are below:**

➢ ការបែងចែករចនាសម្ព័ន្ធនៃកម្មវិធីជាប្លុកតូចៗ

➢ លិទ្ធភាពនៃការប្រើប្រាស់កូដឡើងវិញ

➢ លើកកម្ពស់គេហទំព័រអោយកាន់ស្តង់ដា

➢ ការថែទាំមិនប៉ះពាល់ដល់ផ្នែកផ្សេងទៀត

➢ សន្សំសំចៃពេលវេលា និងការចំណាយ

➢ ដំណើរការអភិវឌ្ឍន៍លឿន

➢ មានភាពបត់បែនខ្លាំង

➢ ការធ្វើផែនការ និងការថែទាំមានភាពងាយស្រួល

8. How many variables scope in PHP? List all of them with examples.

⇨ **There 3 variables scope in PHP including: local, global, and static.**

❖ **Example of 3 variables scope in PHP above:**

❖ **Local Scope:**

```
function my_function() {
  $local_variable = "This is a local variable";
}

my_function();

// This will not work because $local_variable is not defined
outside of my_function()
echo $local_variable;
```

- **Global Scope:**

```
$global_variable = "This is a global variable";

function my_function() {
  global $global_variable;
  echo $global_variable;
}

my_function();
```

- **Static Scope:**

```
function my_function() {
  static $static_variable = 0;

  $static_variable++;

  echo $static_variable;
}

my_function(); // 1
my_function(); // 2
my_function(); // 3
```

9. Explain about “Global” and “Static” keywords in PHP with examples.

⇨ **Explain is below:**

- **“Global” ជា keyword ដែលត្រូវបានគេប្រើក្នុងការ access global variable នៅក្នុង function។**

- **Example of Global:**

```
$x = 5; // global variable

function myFunction() {

    global $x;

    echo $x; // Output: 5

}

myFunction();
```

- ចំពោះការដែលយើងអាចធ្វើការ echo $x variable ដែលនៅក្នុង scope function បានយើងត្រូវប្រើ keyword global ភ្ជាប់ជាមួយនឹង $x variable ដែលនៅក្នុង scope function។

- “Static” ជា keyword ដែលត្រូវបានគេប្រើក្នុងការប្រកាស variable នៅក្នុង local function ។

- **Example of Static:**

```
function myFunction() {
    static $x = 0;
    echo $x; // Output: 0
    $x++;
}
myFunction(); // Output: 0
myFunction(); // Output: 1
myFunction(); // Output: 2
```

- ចំពោះ variable static $x ចាប់ផ្តើមជាមួយតម្លៃ 0 នៅមុនពេល function ត្រូវបាន call។ ក្រោយពេល function ត្រូវបាន call នោះតម្លៃនៃ variable static $x កើនតម្លៃ 1 រាល់ពេលដែល function ត្រូវបាន callម្តងទៀត។

10. What are the different between “echo” and “print”? Give examples about how to use them.

⇨ **Both “echo” and “print” are used to output the data. On the other hand, the different between “echo” and “print” are below:**

Echo	Print
No need parentheses (សញ្ញាវង់ក្រចក)	Need parentheses (សញ្ញាវង់ក្រចក)
No return value	Have return value 1
Output data ម្តងៗមួយ ឬ ច្រើន	Output data ម្តងតែមួយ
Output data លឿនជាងបន្តិច	Output data យឺតជាងបន្តិច

- **Example of “Echo” and “Print” are below:**

```
// Using echo
$name = "John";
echo "Hello, " . $name . "!"; // Output: Hello, John!

// Using print
$age = 25;
print("You are " . $age . " years old."); // Output: You are 25 years old.
```

11. What are the differences between $_POST and $_GET? Give example.

⇨ Both $_POST and $_GET are used to send the data to the server. The different between $_POST and $_GET are below:

$_POST	$_GET
Data មិនត្រូវបានបង្ហាញនៅលើ URLs	Data បង្ហាញនៅលើ URLs
Data send តាមរយៈ HTTP request body	Data send តាមរយៈ URL query string
Data មានភាព Secure	Data អត់មានភាព Secure
ការ send Data គ្មានដែនកំណត់	ការ send Data មានដែនកំណត់

❖ **Example of $_POST and $_GET:**

➢ **$_POST:**

```
<?php
    $username = $_POST[‘username’];
    $password = $_POST[‘password’];

    // process and validate the submitted data
?>
<form method="post" action=" ">

  <input type="text" name="username">

  <input type="password" name="password">

  <button type="submit">Submit</button>

</form>
```

➢ **$_GET:**

```
<a href="profile.php?id=123">View Profile</a>
```

In profile.php file:

```
$id = $_GET['id']; // 123

// Retrieve user profile information based on the ID
```

12. Write code PHP to allow upload multiple files (only image) to server.

⇨ Coding to upload multiple files (only images) to server:

❖ In index.php file:

```
<form action=”upload.php” method=”POST” enctype=”multipart/form-data”>
    <input type=”file” name=”file_upload[]” multiple/>
    <button type=”submit” name=”upload”>Upload</button>
</form>
```

❖ In upload.php file:

```
<?php
    If(isset($_POST[‘upload’])){
            $file = $_FILES[‘file_upload’];
            $allowed_file_types = array(‘jpg’, ‘jpeg’, ‘png’);
            foreach($file[‘tmp_name’] as $key => $value){
                    $tmp_name = $file[‘tmp_name’][$key];
                    $filename = $file[‘name’][$key];
                    $file_type = $file[‘type’][$key];
                    $extension = pathinfo($filename, PATHINFO_EXTENSION);
                    $destination = “uploads/ ” . $filename;

                    If(in_array($extension, $allowed_file_types)){
                            move_uploaded_file($tmp_name, $destination);
                            echo “File is uploaded successfully.”;
                    }else{
                            echo “File extension is invalid”;
                    }
            }
    }
?>
```

13. What are the differences between “Cookies” and “Session”? Give example.

⇨ **Both Cookies and Sessions are used to stored the data in web application. The different between Cookies and Sessions are below:**

COOKIES	SESSIONS
Data stored នៅលើ client-side	Data stored នៅលើ server-side
Data stored បានចំនួនតិច 4KB	Data stored បានចំនួនច្រើន
Data មានការកំនត់រយៈពេល Expires	Data នឹងបាត់បង់នៅពេល logout ឬ end session
Less secure	More secure
Store information ដូចជា user preferences, login information, and tracking data	Store information ដូចជា shopping cart items, user information, and login credentials

❖ **Example of Cookies and Sessions:**

- **Cookies:**

```
// Set a cookie
setcookie('username', 'John', time()+3600);  // will expire in 1H

// Get a Cookie

$result = $_COOKIE[‘username’];
// Destroy a Cookie
unset($_COOKIE[‘username’]);
```

- **Sessions:**

```
// Start a session
session_start();

// Set a session

$_SESSION['username'] = 'John';

// Get a session

$result = $_SESSION[‘username’];
// Destroy a Session
session_destroy();
```

14. Can Cookie and Session store the binary file? Explain.

⇨ **Cookie and Session** អាចធ្វើការ store ជា binary file បានត្រឹមចំនួនតិចតួចតែប៉ុណ្ណោះ។ ដូចនេះ Cookie and Session មិនត្រូវបានគេណែនាំអោយ store ជា binary file នោះទេដោយយោងក្រោមហេតុផលថា ភាពមានដែនកំណត់លើទំហំ ការដំណើរការ និងការពិចារណាសុវត្ថិភាពទិន្នន័យ។

15. Write PHP code to generate the random number with 8 digits (only number).

⇨ **Coding to generate the random number with 8 digits (only number):**

```
<?php
        // generate a random number with 8 digits
        $random_number = mt_rand(10000000, 99999999);

        // Display the random number
        echo $random_number;
?>
```

16. Write PHP code to write “Hello World!” 1000 line in a file named myfile.txt.

⇨ **Coding to write “Hello World!” 1000 line in a file named myfile.txt:**

```
<?php
        $file = fopen(‘myfile.txt’, ‘w’);

        for($i = 1; $i<=1000; $i++){
                fwrite($file , ‘Hello World!’);
        }
        fclose($file);
?>
```

17. Write 3 functions to connect, insert, update, and delete data from MySQL database respectively.

⇨ **Coding to connect and 3 functions including: insert, update, and delete are below:**

❖ **Connection function**

```
function connect($hostname, $username, $password, $dbname){
    $conn = new mysqli($hostname, $username, $password, $dbname);
    If(!conn->connect_error){
            return $conn;
    }else{
            echo “Connection Failed” . $conn->connect_error;
            exit();
    }
```

❖ Insert function

```
function insert($table, $data = array()){
    $conn = connect();
    $fields = implode(“,”, array_keys($data));
    $values = implode(“’,’”, array_values($data));
    $sql = “INSERT INTO ” . $table . “ (” . $fields . “) VALUES (‘” . $values . “’)”;
    $result = $conn->query($sql);
    If(!result){
            echo “Error: “ . $conn->connect_error;
            return false;
    }
    $conn->close();
    return true;
}
```

❖ Update function

```
function update($table, $data = array(), $conditon = “”){
    $conn->connect();
    $field_value = “”;
    foreach($data as $field => $value){
            $field_value .= “ ”. $field . “=’” . $value . “’,”;
    }
    $field_value = substr($field_value, 0, strlen($field_value) - 1);
    $sql = “UPDATE ” . $table . “ SET “ . $field_value . “ WHERE ” . $condition;
    $result = $conn->query($sql);
    If(!$result){
            echo “Error: ” . $conn->connect_error;
            return false;
    }
    $conn->close();
    return true;
}
```

❖ Delete function

```
function delete($table, $condition){
    $conn = connect();
    $sql = “DELETE FROM ” . $table . “ WHERE ” . $condition;
    $result = $conn->query($sql);
```

```
        If(!result){
                echo “Error: ” . $conn->connect_error;
                return false;
        }
        $conn->close();
        return true;
    }
```

18. What is the difference between $var1 and $$var1?

⇨ **The different between $var1 and $$var1 are below:**

- ❖ **$var1** ជា variable ធម្មតានៅក្នុង PHP។ វាជា variable មួយដែលមានតម្លៃតែមួយគត់ ហើយ អាចចូលទៅប្រើប្រាស់ដោយផ្ទាល់តាមរយៈឈ្មោះរបស់ Variable ។
- ❖ **$$var1** នៅក្នុង PHP តេហៅថា variable variable។ វាជាអ្នកបង្កើតអោយមាន variable name dynamic ដោយអាចចូលទៅប្រើប្រាស់តម្លៃនៃ Variable ផ្សេងទៀតក្រោមឈ្មោះ variable ខ្លួនឯង។

19. What are the ways to define a constant in PHP?

⇨ **There are 2 ways to define a constant in PHP including: “define” function and ‘const’ keyword.**

- ❖ **Using define() function:**

```
define("PI", 3.14);
echo PI; // outputs 3.14
```

- ❖ **Using const keyword:**

```
const PI = 3.14;
echo PI; // outputs 3.14
```

20. What does isset() function?

⇨ **isset() function** ជា built-in-function របស់ PHP ដែលគេប្រើវាសម្រាប់ត្រួតពិនិត្យមើលថាតើ variable ត្រូវបានបង្កើតរួច ឬ null។ isset() function នឹងផ្តល់តម្លៃជា Boolean (true or false) ដែលអាស្រ័យលើ variable បានបង្កើតរួច ឬអត់។

21. What is $_SERVER? Explain about the benefit of using this variable.

⇨ **$_SERVER** ជា Superglobal variable ដែលផ្តល់នូវការប្រើប្រាស់ព័ត៌មានដែលទាក់ទងនឹង Server។ $_SERVER ជា Associative Array ដែលមានព័ត៌មានផ្សេងៗគ្នាដូចជា Request method, header, URL and Script paths។

- ❖ **The benefit of using $_SERVER variable is below:**
  - ➢ **$_SERVER** ផ្តល់នូវការចូលប្រើប្រាស់ Server variable ដែលមានព័ត៌មានអំពី user browser
  - ➢ **$_SERVER** ផ្តល់អោយព័ត៌មានអំពីការកំណត់ហេដ្ឋារចនាសម្ព័ន្ធ Server ដូចជា root directory ជាដើម
  - ➢ **$_SERVER** ផ្តល់នូវការបង្កើនសុវត្ថិភាពចំពោះការចូលទៅប្រើប្រាស់ IP Address របស់ Client ណាមួយ
  - ➢ **$_SERVER** ផ្តល់នូវវិធីស្តង់ដារនៃការចូលប្រើប្រាស់ server information តាមរយៈ web server និងនៅតាម platform ផ្សេងៗគ្នា។

22. What is the difference between “require”, “require_once”, “include”, “include_once”?

⇨ **The different is below:**

- ❖ **“require”** is used to pull in the file and die if it is missing
- ❖ **“require_once”** is used to pull in the file and die if it is missing unless it has already been pulled in before
- ❖ **“include”** is used to include the file every time it’s called
- ❖ **“include_once”** is used to include the file only once, even it’s called multiple times.

23. What is the difference between MySQLi and PDO, and when would you use one over the other?

⇨ **Both MySQLi and PDO are extension of PHP ដែលអាចធ្វើទំនាក់ទំនងជាមួយនឹង MySQL Database។ The different are below:**

MySQLi	PDO
Support on MySQL Database	Support on Multiple Databases
មានលក្ខណៈជា Procedural	មានលក្ខណៈជា Object-oriented
Less Secure	More Secure
Easier to learn	More difficult to learn
Features មានចំនួនតិច	Features មានចំនួនច្រើន

24. What is object-oriented programming (OOP), and how do you create classes and objects in PHP to organize code and improve code reuse?

⇨ OOP is a programming language paradigm that focuses on creating objects that contain both data and behavior.

- ❖ Example
  - ➢ creating class and object in PHP OOP using the class keyword:

```
Class Myclass{
    // body the class that contain properties and methods
}
```

  - ➢ Create an instance of the class using the new keyword:

```
$object = new MyClass();
```

25. How can we create a database named “mydatabase” using PHP and MySQl?

⇨ We can create like below:

```
<?php
        $hostname = “localhost”;
        $username = “root”;
        $password = “”;
        $database = “mydatabase”;
        $conn = new mysqli($hostname, $username, $password);

        If(!$conn->connect_error){
                $conn->query(“CREATE DATABASE ” . $database);
        }else{
                echo “Connection Failed” . $conn->connect_error;
        }
        $conn->close();
?>
```

26. Write a PHP code to send an email.

⇨ Coding PHP to send an email:

```
<?php
        $to = rupp@gmail.com;
        $subject = “This is the subject of the email”;
        $message = “This is the message of the email”;
        $headers = “From: sender@gmail.com”;

        mail($to, $subject, $message, $headers);
?>
```

27. How can you execute SQL Queries in PHP, and what are some best practices for handling user input to prevent SQL injection attacks?

⇨ We can execute SQL Queries in PHP by 2 ways including: Using the MySQLi and PDO extension.

- ❖ Some best practices for handling user input to prevent SQL injection attacks are below:
  - ➢ Always sanitize and validate User Input using “filter_var()” or “mysqli_real_escape_string()” in a SQL query.
  - ➢ Never use user input directly in a SQL query
  - ➢ User prepared statements to execute SQL queries

28. Write a code let user upload only image to server and the image must less than 3MB.

⇨ Coding let user upload only image with less 3 MB:

❖ **In index.php file:**

```
<form action= “upload.php” method=”POST” enctype=”multipart/form-data”>
  <input type=”file” name=”file_img” />
  <button type=”submit” name=”upload”>Upload</button>
</form>
```

❖ **In upload.php file**

```
<?php
  If(isset($_POST[‘upload’])){
      $file = $_FILES[‘file_img’];
      $file_max_size = 3 * 1024 *1024; // 3 MB

      If(file_exists($file[‘tmp_name’]) || is_upload_file($file[‘tmp_name’])){
          $filename = basename($file[‘name’]);
          $destination = “uploads/” . $filename;

          // validate the file size
          if($file[‘size’] > $file_max_size){
              echo “File size is too large”;
              return true;
          }
          else{
              move_uploaded_file($file[‘tmp_name’], $destination);
              echo “Uploaded Successfully”;
          }
      }
      else{
```

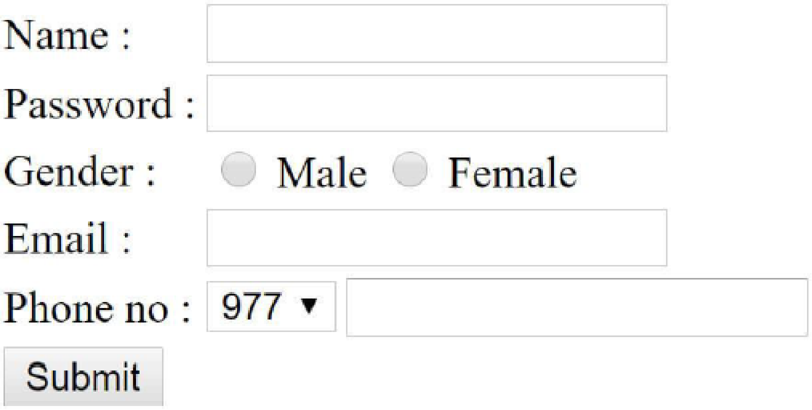
```
                    echo “No file is upload”;
            }
    }
    $conn->close();
?>
```

29. Write a PHP to connect to database and insert a data to database below:

a) Database name: mydatabase [id (auto number), Name (varchar (100)), sex (boolean), email (varchar(200)), phone (varchar(100)), password (varchar(100))

b) Password :123

c) Host: localhost



Name :

Password :

Gender : ○ Male ○ Female

Email :

Phone no : 977 ▾

Submit

Note: the password is auto generated (8-digit (only number))

⇨ **Coding to insert a data to database:**

- **In index.php file:**

```
// PHP code
<?php
    $hostname = “localhost”;
    $username = “root”;
    $password = “123”;
    $database = “mydatabase”;
    $conn = new mysqli($hostname, $username, $password, $database);

    If(isset($_POST[‘submit’])){
            $name = $_POST[‘name’];
            $email = $_POST[‘email’];
            $password = $_POST[‘password’];
            $sex = $_POST[‘sex’];
            $country_code = $_POST[‘country_code’];
            $phone = $_POST[‘phone’];
            $full_phone_number = “+” . $country_code . “ ” . $phone;
            $password = rand(10000000, 99999999); // auto generate 8-digits

                $sql = “INSERT INTO user (Name, sex, email, phone, password)
                VALUES (‘$name’, ‘$sex’, ‘$email’, ‘$full_phone_number’,
                ‘$password’)”;
                $insert_user = $conn->query($sql);
            If($insert_user){
                    header(“Location: index.php”);
                }
        }
        $conn->close();
?>


// html code
<html>
<body>
    <form action=”” method=”POST”>
            <div>
                    <label>Name</label>
                    <input type=”text” name=”name”/>
                </div>
                <div>
                    <label>Password</label>
                    <input type=”password” name=”password” />
                </div>
                <div>
                    <label>Gender</label>
                    <input type=”radio” value=”M” name=”sex” /> Male
                    <input type=”radio” value=”F” name=”sex” /> Female
                </div>
                <div>
                    <label>Email</label>
                    <input type=”email” name=”email” />
                </div>
                <div>
                    <label>Phone no</label>
                    <select name=”country_code”>
```

```
              <option value=”855”>855</option>
              <option value=”977”>977</option>
          </select>
          <input type=”tel” name=”phone” />
        </div>
        <button type=”submit” name=”submit”>Submit</button>
    </form>
</body>
</html>
```

30. Write to search for a student based on their name and date of birth in a single php file. If the student is found, “Found!” message will be display while if not, “Not Found!” message is shown in the page.

⇨ Coding to search student:

- In Index.php file

// PHP code

```
<?php
  $conn = new mysqli(‘localhost’, ‘root’, ‘123’, ‘mydatabase’);
  If(isset($_POST[‘search’])){
      $name = $_POST[‘name’];
      $date_of_birth = $_POST[‘date_of_birth’];
        $sql = “SELECT * FROM student WHERE name LIKE ’%$name%’ OR
        date_of_birth = ‘$date_of_birth’ ”;
        $result = $conn->query($sql);
        If($result->num_rows > 0){
          echo “Found!”;
        }
        else{
          Echo “Not Found !”;
        }
    }
    $conn->close();
```

```
?>



// html code
<html>
<body>
    <form action=”” method=”POST”>
            <div>
                    <label>Name</label>
                    <input type=”text” name=”name” />
                </div>
                <div>
                    <label>Date of birth</label>
                    <input type=”text” name=”date_of_birth” />
                </div>
                <button type=”submit” name=”search”>Search</button>
        </form>
</body>
</html>
```