LAB MANUAL

WEB PROGRAMMMING LABORATORY

Submitted by

SHASHANK GURUNAGA (U03NM21T029054)

VI SEM, B. TECH (CSE)

Under the guidance of

Ms Amulya,
Guest Faculty,
Department of Computer Science,
UVCE

Department of Computer Science and Engineering

UNIVERSITY OF VISVESVARAYA COLLEGE OF ENGINEERING
K.R. Circle, Bangalore – 560001

BANGALORE UNIVERSITY

UNIVERSITY OF VISVESVARAYA COLLEGE OF ENGINEERING
K.R. Circle, Bangalore – 560001

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Shashank Gurunaga** (U03NM21T029054) of VI Semester, B. Tech, Computer Science and Engineering, have submitted the Lab Manual in fulfilment for the Web Programming Lab, prescribed by the Bangalore University for the academic year 2023-24.

Ms. Amulya,
Guest Faculty,
Dept. of CSE,
UVCE

Dr. Thriveni J,
Professor &
Chairperson,
Dept. of CSE,
UVCE

Examiners:

1. ……

2. ...……

ACKNOWLEDGEMENT

I take this opportunity to thank our institution University of Visvesvaraya College of Engineering for having given me an opportunity to carry out this project.

I would like to thank Dr. Paul Vizhian, Principal, UVCE, for providing us all the facilities to work on this project. I am indebted to him for being my pillar of strength

I would like to thank Dr. Thriveni J, Professor and Chairperson, Department of Computer Science and Engineering, UVCE, for providing us all the facilities to work on this project.

I am thankful to Ms Amulya , Guest Faculty(CSE Department), Department of Computer Science and Engineering, UVCE, for his constant support during the course of the project.

I express my sincere thanks to all teaching and non-teaching staff, Department of Computer Science and Engineering, UVCE for all the facilities that they have provided me for successfully completing this project.

I also thank my parents and friends for their continuous support and encouragement.

ABSTRACT

The **Web Programming Lab Manual** provides a comprehensive guide to developing dynamic and interactive websites using a variety of web technologies. The manual is divided into two parts, each focusing on different core aspects of web programming.

In **Part A**, students will explore foundational web technologies such as **HTML**, **PHP**, and **Perl**:

- **HTML (HyperText Markup Language)** introduces the structure and design of web pages. Students will learn how to create and style web pages using HTML5, enhancing their layout with elements, forms, multimedia, and responsive design techniques.
- **PHP (Hypertext Preprocessor)** is used for server-side scripting. This section covers building dynamic web pages that interact with databases, process forms, and handle user inputs, emphasizing the integration of PHP with HTML to create robust web applications.
- **Perl**, another server-side language, is explored for tasks such as text processing and CGI scripting. Students will gain practical experience in creating dynamic web content and handling HTTP requests with Perl.

In **Part B**, the manual focuses on advanced web technologies like **Java Servlets**:

- **Java Servlets** introduces students to server-side programming in Java. Students will learn how servlets handle HTTP requests and responses, manage sessions, and interact with databases to build scalable, secure, and efficient web applications.
- **HTML Integration** with Java Servlets will also be covered to illustrate how servlets can dynamically generate web content and interact with static HTML to deliver customized user experiences.

This lab manual aims to equip students with the skills to develop and deploy full-stack web applications, mastering both client-side and server-side technologies. By completing these labs, students will gain hands-on experience in creating interactive, responsive, and data-driven web solutions.

INDEX

INTRODUCTION

Web programming involves creating dynamic and interactive web applications. It encompasses various technologies such as HTML for structure, CSS for styling, JavaScript for interactivity, and server-side languages like PHP, Perl, and Java. Here's a brief overview of the programs mentioned:

These web programming tasks provide a comprehensive approach to server management, user interaction, and database manipulation using HTML, PERL/PHP, MySQL, and Java Servlets. Programs like Server Information Display and UNIX Command Execution focus on retrieving server details and executing commands, enabling administrators to monitor and manage server configurations. User Interaction Programs include greeting messages and visitor counters, which enhance user engagement by providing personalized experiences and tracking visitor counts. The Time-based Greeting and Login Check program introduces dynamic responses based on the server access time, while the Digital Clock Display offers real-time time tracking. The Database Content Display and Insert Data programs demonstrate fundamental database interactions, allowing users to view, add, and query data from MySQL databases directly through a web interface. The Book Information Management program exemplifies data collection and storage, a crucial feature for managing library or inventory systems, while Search Book by Title facilitates data retrieval by user-provided keywords.

The programs using Java Servlets extend functionality by incorporating dynamic page customization, cookie management, and session handling. User Interaction and Page Customization allows personalized greetings and enables users to change the page's background color, showcasing the flexibility of Java Servlets in client-side customization. Cookie Management and Session Management programs illustrate the importance of user data retention and stateful interactions, critical for

maintaining seamless user experiences across sessions. Request Server Information and User Information Display programs demonstrate data retrieval methods for both server details and user input, highlighting essential concepts in web client-server communication. Collectively, these programs provide a practical overview of essential web programming concepts, including server-side scripting, database management, session tracking, and user interface customization, forming a strong foundation for developing dynamic and interactive web applications.

1a) Program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.

1a.html

```
<html>
<!-- path to perl file -->
  <form action="http://localhost/cgi-bin/1a.pl">
    <center>
    <input type=submit value=Show_Server_Info />
    </center>
  </form>
</html>
```

1a.pl

```
#!/usr/bin/perl
#this is a here-document
print<<here;
Content-type:text/html\n\n
<html>
<center>
  <table border=1>
  <tr>
    <th>ENV_VARIABLES</th><th>Value</th>
  </tr>
here
#end of here document
#display values in a table
foreach $i(sort keys %ENV)
```

```
{
print "<tr><td>$i</td><td>$ENV{$i}</td></tr>";
}
print<<here
        </table>
    </center>
</html>";
here
```

Output:

ENV_VARIABLES	Value
DOCUMENT_ROOT	/var/www
GATEWAY_INTERFACE	CGI/1.1
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_CONNECTION	keep-alive
HTTP_DNT	1
HTTP_HOST	localhost
HTTP_REFERER	http://localhost/html/5a.html
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:17.0) Gecko/20131030 Firefox/17.0 Iceweasel/17.0.10
PATH	/usr/local/bin:/usr/bin:/bin
QUERY_STRING	
REMOTE_ADDR	::1
REMOTE_PORT	38557
REQUEST_METHOD	GET
REQUEST_URI	/cgi-bin/5a.pl
SCRIPT_FILENAME	/usr/lib/cgi-bin/5a.pl
SCRIPT_NAME	/cgi-bin/5a.pl
SERVER_ADDR	::1
SERVER_ADMIN	webmaster@localhost
SERVER_NAME	localhost
SERVER_PORT	80
SERVER_PROTOCOL	HTTP/1.1
SERVER_SIGNATURE	*Apache/2.2.22 (Debian) Server at localhost Port 80*
SERVER_SOFTWARE	Apache/2.2.22 (Debian)

1b) Program to accept UNIX command from a HTML form and to display the output of the command executed.

1b.html

```
<html>
   <!-- path to perl file -->
   <form action="http://localhost/cgi-bin/1b.pl">
      <!-- input command name -->
      Command:<input type=text name=com>
         <input type=submit value=submit />
   </form>
</html>
```

1b.pl

```
#!/usr/bin/perl
use CGI ':standard';
#the following line is used for displaying the output of the script in the browser
print "Content-type:text/html\n\n";

$c=param('com');
system($c);
exit(0);
```

Output:

Command: submit

Command: date submit

Sun Jan 19 22:30:07 IST 2014

2a) Program to accept the User Name and display a greeting message.

2a.html

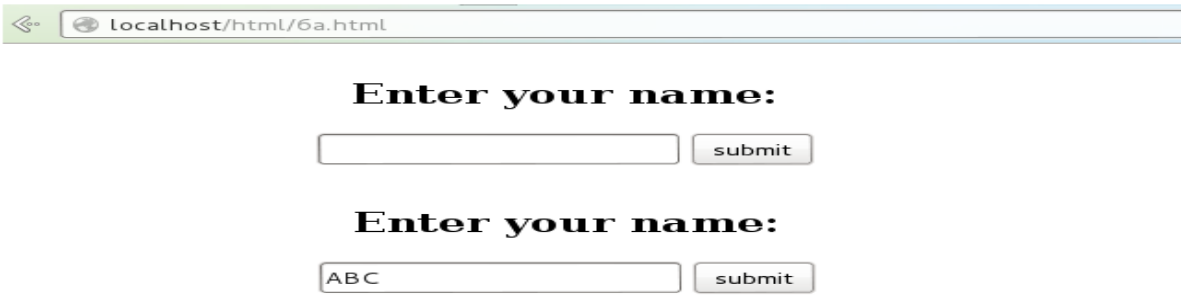
```
<html>
    <!-- path to perl file -->
    <form action="http://localhost/cgi-bin/6a.pl">
        <center>
            <!-- input -->
            <h2>Enter your name:</h2>
            <input type=text name=name />
            <input type=submit value=submit />
        </center>
    </form>
</html>
```

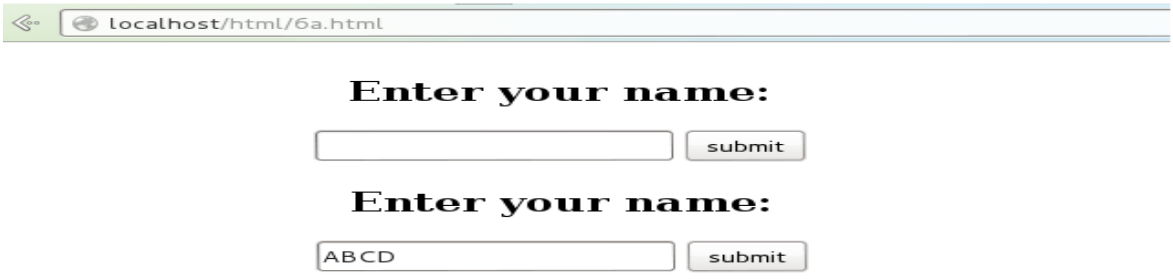
2a.pl

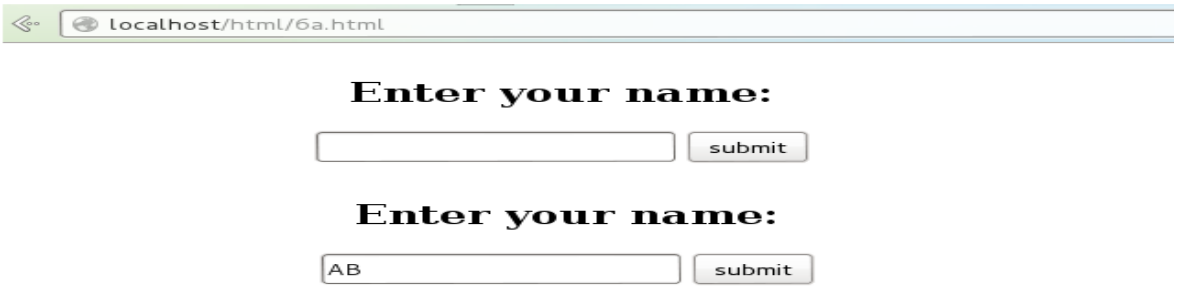
```
#!/usr/bin/perl
#load CGI standard routines
use CGI':standard';
#take input and store in local variable
$cmd=param('name');
#define various greeting messages
@greet=("Hello","Hai","Nice meeting you","Have a nice day");
#choose a message based on the length of the input
$index=int rand scalar @greet;
print<<here;
Content-type:text/html\n\n
<html>
```

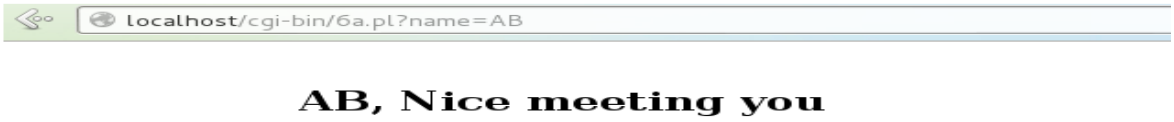
```
    <center>
        <h2>$cmd, $greet[$index]</h2>
    </center>
</html>
here
?>
```

Output:

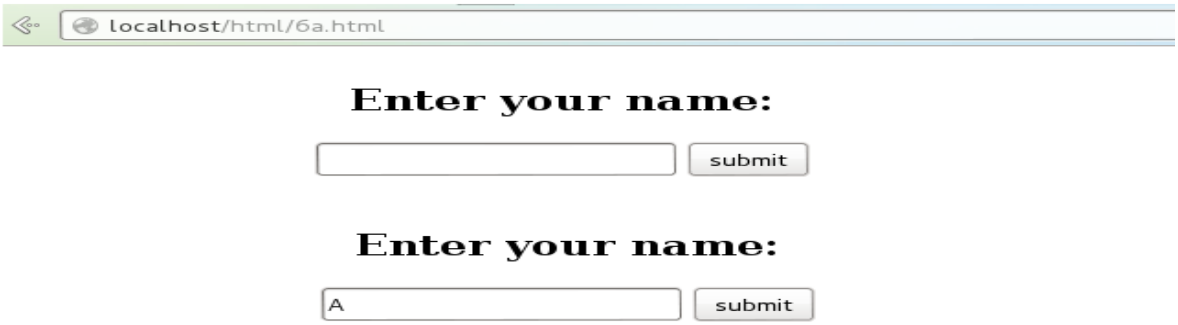








AB, Nice meeting you



A, Hai

2b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

2b.html

```
<html>
    <form action="http://localhost/cgi-bin/6b.pl">
        <center>
            <input type=submit value=Show_no_of_visitors />
        </center>
    </form>
</html>
```

2b.pl

```
#!/usr/bin/perl
#load CGI standard routines
use CGI':standard';
print "Content-type:text/html\n\n";
#open a file called count.txt to store values of number of views
open(FILE,'<count.txt');
#contents of file are copied to the variable
$count=<FILE>;
close(FILE);
$count++;
#append the new incremented value to the beginning of the file
open(FILE,'>count.txt');
print FILE "$count";
print "This page has been viewed $count times";
```

Output :

3. Program to display a greeting based on the access time of the Web server. Also to verify whether the webmaster is currently logged in.

3.html

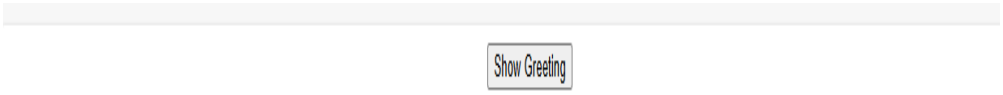
```
<!DOCTYPE html>
<html>
   <form action="http://localhost/cgi-bin/3.pl" method="post">
      <center>
         <input type="submit" value="Show Greeting" />
      </center>
   </form>
</html>
```

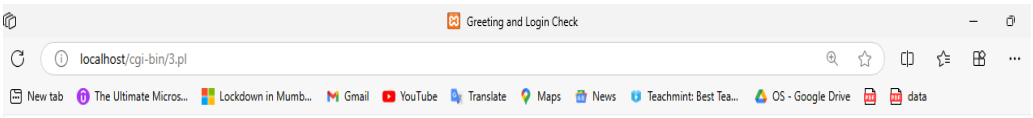
3.pl

```
#!/usr/bin/perl
use strict;
use warnings;
use CGI ':standard';
use POSIX 'strftime';

# Print HTTP header
print header;

# Get the current server time in hours
my $hour = strftime("%H", localtime);



# Determine greeting based on time
```

```
my $greeting;
if ($hour < 12) {
    $greeting = "Good Morning!";
} elsif ($hour < 18) {
    $greeting = "Good Afternoon!";
} else {
    $greeting = "Good Evening!";
}

# Check if the webmaster (admin) is logged in
# This is a basic simulation. Adjust it according to your actual login verification method.
# In a real scenario, you might check session or authentication data.
my $is_webmaster_logged_in = 1; # Set to 1 to simulate the webmaster being logged in, or 0 otherwise.
my $login_status = $is_webmaster_logged_in ? "The webmaster is currently logged in." : "The webmaster is not logged in.";
# Output HTML content
print start_html("Greeting and Login Check"),
    h1("Greeting Based on Access Time"),
    p($greeting),
    h2("Webmaster Login Status"),
    p($login_status),
    end_html;
```

Output:





Greeting Based on Access Time

Good Evening!

Webmaster Login Status

The webmaster is currently logged in.

4. **Program to display a digital clock which displays the current time of the server.**

4.html

```
<html>
<form action="http://localhost/cgi-bin/4.pl">
<center>
    <input type=submit value=Click_to_display_time>
</center>
</form>
</html>
```

4.pl

```
#!/usr/bin/perl
use CGI':standard';
#refresh the display every second
print "Refresh:1\n";
print "Content-type:text/html\n\n";
($s,$m,$h)=localtime(time);
print br "The current server time is $h:$m:$s";
print br "In words,the time is- $h hours, $m minutes and $s seconds";
```

Output:

5. Program to display the current contents of the table in a database.

5.html

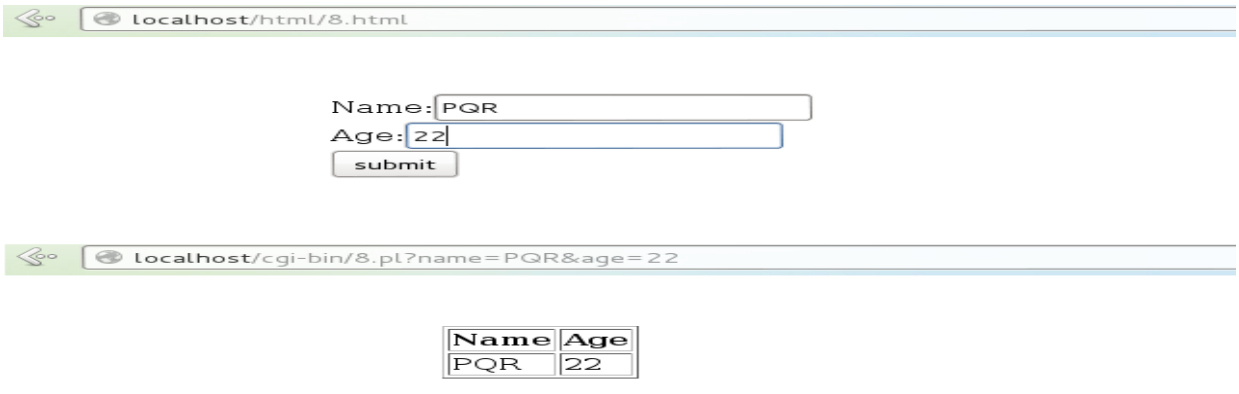
```
<html>
<form action="http://localhost/cgi-bin/5.pl">
        Name:<input type=text name=name><br>
        Age:<input type=text name=age><br>
        <input type=submit value=submit />
    </form>
</html>
```

5.pl

```
        #!/usr/bin/perl
        use CGI':standard';
        #import database module
        use DBI;
        #take inputs and store in local variables
        $nm=param('name');
        $age=param('age');
        #connect takes parameter driver_name:database_name
  #this creates a database handle & stores it in $db
        $db=DBI->connect("DBI:mysql:test");
        #query
         $str=("insert into info values('$nm',$age)");
         #compiles the query and returns an object reference which is called
  statement handle
        $q=$db->prepare($str);
        #execute the query using statement handle
        $q->execute();
```

```
$q=$db->prepare("select * from info");
$q->execute();
print<<1;
Content-type:text/html\n\n
<html>
<table border=1>
<tr>
        <th>Name</th>
        <th>Age</th>
</tr>

#obtain all the rows with matching patterns
while(($nm,$age)=$q->fetchrow()) {
        print "<tr><td>$nm</td><td>$age</td></tr>";
}
print "</table></html>";
#release the statement handle & database handle
$db->disconnect;
$q->finish;
```

Output:



6. Program, to insert new name and age information entered by the user into the database.

6.html

```
<html>
<form action="http://localhost/cgi-bin/5.pl">
    Name:<input type=text name=name><br>
    Age:<input type=text name=age><br>
    <input type=submit value=submit />
  </form>
 </html>
```

6.pl

```
#!/usr/bin/perl
use CGI':standard';
#import database module
use DBI;
#take inputs and store in local variables
$nm=param('name');
$age=param('age');
#connect takes parameter driver_name:database_name
#this creates a database handle & stores it in $db
$db=DBI->connect("DBI:mysql:test");
#query
$str=("insert into info values('$nm',$age)");
#compiles the query and returns an object reference which is called statement handle
$q=$db->prepare($str);
#execute the query using statement handle
```

```
$q->execute();
$q=$db->prepare("select * from info");
$q->execute();
print<<1;
Content-type:text/html\n\n
<html>
    <table border=1>
   <tr>
            <th>Name</th>
            <th>Age</th>
   </tr>
1
#obtain all the rows with matching patterns
while(($nm,$age)=$q->fetchrow()) {
print "<tr><td>$nm</td><td>$age</td></tr>";
}
print "</table></html>";
#release the statement handle & database handle
$db->disconnect;
$q->finish;
```

Output:

7. Program to query the data base and to display the result on a webpage.

7.html

```
<html>
<form action="http://localhost/cgi-bin/5.pl">
        Name:<input type=text name=name><br>
        Age:<input type=text name=age><br>
        <input type=submit value=submit />
  </form>
</html>
```

7.pl

```
#!/usr/bin/perl
use CGI':standard';
#import database module
use DBI;
#take inputs and store in local variables
$nm=param('name');
$age=param('age');
#connect takes parameter driver_name:database_name
#this creates a database handle & stores it in $db
$db=DBI->connect("DBI:mysql:test");
#query
$str=("insert into info values('$nm',$age)");
#compiles the query and returns an object reference which is called
statement handle
```

```
$q=$db->prepare($str);
#execute the query using statement handle
$q->execute();
$q=$db->prepare("select * from info");
$q->execute();
print<<1;
Content-type:text/html\n\n
<html>
    <table border=1>
   <tr>
            <th>Name</th>
            <th>Age</th>
   </tr>
1
#obtain all the rows with matching patterns
while(($nm,$age)=$q->fetchrow()) {
print "<tr><td>$nm</td><td>$age</td></tr>";
}
print "</table></html>";
#release the statement handle & database handle
$db->disconnect;
$q->finish;
```

Output:

8. **Program to accept book information viz. Accession number, title, authors, edition and publication from a webpage and to store those in a database.**

8.html

```
<html>
<head>
   <title>Book Information</title>
</head>
<body>
   <h1>Enter Book Information</h1>
   <form action="save_book.php" method="POST">
      <label>Accession Number:</label>
      <input type="text" name="accession_number"
required><br><br>

      <label>Title:</label>
      <input type="text" name="title" required><br><br>

      <label>Author:</label>
      <input type="text" name="author" required><br><br>

      <label>Edition:</label>
      <input type="text" name="edition" required><br><br>

      <label>Publication:</label>
      <input type="text" name="publication" required><br><br>

      <input type="submit" value="Save Book">
   </form>
</body>
</html>
```

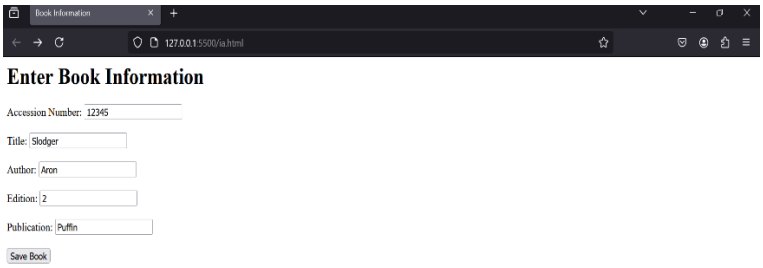
save_book.php

```
<?php
// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "library";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Get form data
$accession_number = $_POST['accession_number'];
$title = $_POST['title'];
$author = $_POST['author'];
$edition = $_POST['edition'];
$publication = $_POST['publication'];

// SQL query to insert data
$sql = "INSERT INTO books (accession_number, title, author,
edition, publication)
VALUES ('$accession_number', '$title', '$author', '$edition',
'$publication')";

if ($conn->query($sql) === TRUE) {
    echo "New book record created successfully";
}
else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

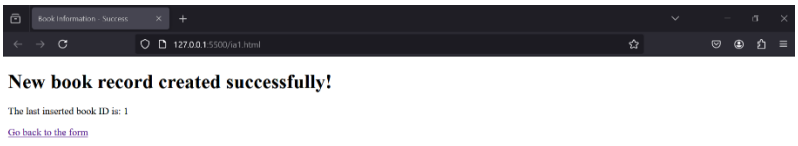
MySQL Database Table:

```
CREATE DATABASE library;

USE library;

CREATE TABLE books ( id INT(6) UNSIGNED
AUTO_INCREMENT PRIMARY KEY, accession_number
VARCHAR(30) NOT NULL, title VARCHAR(100) NOT NULL,
author VARCHAR(100) NOT NULL, edition VARCHAR(30) NOT
NULL, publication VARCHAR(100) NOT NULL );
```

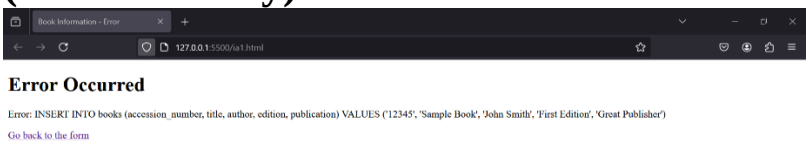
Output:



(on successful entry)



(on failed entry)



9. Program to search a book for a title given by the user on a webpage and display the search results with proper headings.

9. html

```
<!DOCTYPE html>
<html>
<head><title>Search Book</title></head>
<body>
        <h1>search book by title</h1>
       <form action="9.php" method="GET">
          <label> Enter Book Title</label>
          <input type="text" name="title" required><br><br>
          <input type="submit" value="search">
      </form>
</body>
</html>
```

9.php

```
<?php
$servername="localhost";
$username="root";
$password=" ";
$dbname="library";

$conn = new mysqli($servername, $username, $password, $dbname);
```

```
if($conn->connect_error){
    die("connection failed:".$conn->connect_error);
}
$title=$_GET['title'];
$sql="SELECT accession_number,title,author,edition,publication FROM books where title LIKE %title%";
$result=$conn->query($sql);

if($result->num_rows>0){
    echo "<h2>Search Results</h2>";
    echo "<table border='1'><tr><th>Accession Number</th><th>Title</th><th>Author</th><th>Edition</th><th>Publication</th></tr>";
    while($row=$result->fetch_assoc()){
        echo "<tr><td>".$row["accession_number"]."</td><td>".$row["title"]."</td><td>".$row["author"]."</td><td>".$row["edition"]."</td><td>".$row["publication"]."</td></tr>";
    }
    echo "</table>";
}
else{
    echo "No books found with title '$title'";
}
$conn->close();
?>
```

Output:

(on success)

(on fail)

Step-by-Step Guide to Using Java Servlets on Eclipse:

1. Set Up Your Development Environment

1. Install Eclipse IDE
   Download and install Eclipse IDE for Java EE Developers from the official Eclipse website.
2. Install Apache Tomcat
   Download Apache Tomcat (e.g., version 9 or 10) from the official website and extract it to a directory on your computer.
3. Configure Tomcat in Eclipse
   - Open Eclipse and navigate to Window > Preferences > Server > Runtime Environments.
   - Click Add, select Apache Tomcat, and specify the directory where Tomcat is installed.

---

2. Create a Dynamic Web Project

1. Go to File > New > Project and select Dynamic Web Project.
2. Name the project (e.g., "ServletExample") and choose a target runtime (e.g., Apache Tomcat).
3. Click Finish.

---

3. Write Your First Servlet

1. Right-click the project and navigate to New > Servlet.
2. Enter a package name (e.g., com.example) and a servlet name (e.g., HelloServlet).
3. Eclipse will generate a basic servlet structure. Update the doGet or doPost method as needed.

---

4. Deploy and Run the Project

1. Right-click on the project and choose Run As > Run on Server.
2. Select Apache Tomcat as the server and click Finish.
3. Your servlet should now be accessible

---

10. a) Program to accept username and display a greeting message.

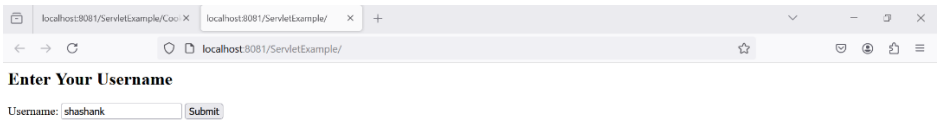
index.html

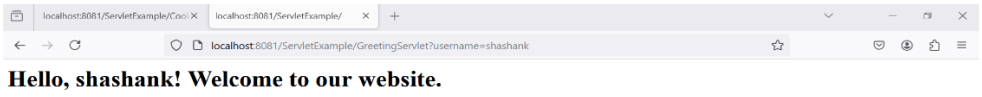
```
<html>
<body>
   <h2>Enter Your Username</h2>
   <form action="GreetingServlet" method="GET">
      Username: <input type="text" name="username">
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

GreetingServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/GreetingServlet")
public class GreetingServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;

   protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
      String username = request.getParameter("username");

      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<h1>Hello, " + username + "! Welcome to our
website.</h1>");
      out.println("</body></html>");
   }
}
```

Output:


localhost:8081/ServletExample/
localhost:8081/ServletExample/
localhost:8081/ServletExample/
Enter Your Username
Username: shashank
Submit


localhost:8081/ServletExample/
localhost:8081/ServletExample/
localhost:8081/ServletExample/GreetingServlet?username=shashank
Hello, shashank! Welcome to our website.

10b) Program to change the background colour of the page based on the colour selected by the user.

color.html

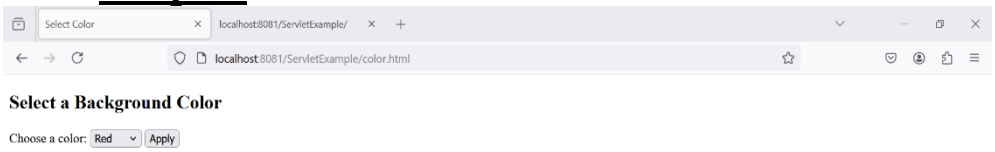
```
<html>
<head>
   <title>Select Color</title>
</head>
<body>
   <h2>Select a Background Color</h2>
   <form action="ColorServlet" method="GET">
      <label for="color">Choose a color:</label>
      <select id="color" name="color">
         <option value="red">Red</option>
         <option value="blue">Blue</option>
         <option value="green">Green</option>
         <option value="yellow">Yellow</option>
      </select>
      <input type="submit" value="Apply">
   </form>
</body>
</html>
```

ColorServlet.java

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ColorServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;

   protected void doGet(HttpServletRequest request,
```

```
HttpServletResponse response) throws ServletException,
IOException {
        String color = request.getParameter("color");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><body style='background-color:" + color +
"'>");
        out.println("<h1>The background color is " + color + "</h1>");
        out.println("</body></html>");
    }
}
```

Output:



11. **Program to display a greeting based on the access time of the server**.

Index.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Greeting Page</title>
</head>
<body>
   <h1>Welcome to the Greeting Page</h1>
   <form action="GreetingTimeServlet" method="get">
      <button type="submit">Get Greeting</button>
   </form>
</body>
</html>
```

GreetingTimeServlet.java

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalTime;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class GreetingTimeServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request,
HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      LocalTime now = LocalTime.now();
      String greeting;

      if (now.isBefore(LocalTime.NOON)) {
```

```
            greeting = "Good Morning!";
        } else if (now.isBefore(LocalTime.of(18, 0))) {
            greeting = "Good Afternoon!";
        } else {
            greeting = "Good Evening!";
        }

        out.println("<html><body>");
        out.println("<h1>" + greeting + "</h1>");
        out.println("</body></html>");
    }
}
```

Output:

12. Program to create and display a cookie.

cookie.html

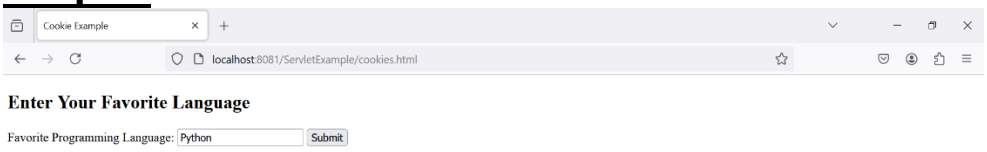
```
<html>
<head>
   <title>Cookie Example</title>
</head>
<body>
   <h2>Enter Your Favorite Language</h2>
   <form action="CookieServlet" method="POST">
   <label for="lang">Favorite Programming Language:</label>
      <input type="text" id="lang" name="lang">
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

CookieServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/CookieServlet")
public class CookieServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;

   protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
      String lang = request.getParameter("lang");

      Cookie cookie = new Cookie("language", lang);
      cookie.setMaxAge(60 * 60 * 24); // 1 day expiry
```

```
            response.addCookie(cookie);
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<html><body>");
            out.println("<h1>Favorite Language Cookie Created: " + lang +
"</h1>");
            out.println("</body></html>");
        }
    }
```

Output:



---

13. Program to create a session and display session information viz, session ID, creation time and last accessed.

Sessioninfo.html

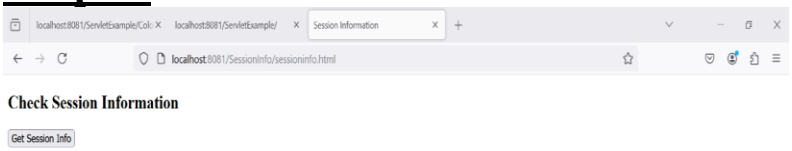
```
<html>
<head>
   <title>Session Information</title>
</head>
<body>
   <h2>Check Session Information</h2>
   <form action="SessionInfoServlet" method="get">
      <input type="submit" value="Get Session Info">
   </form>
</body>
</html>
```

SessionInfoServlet.java

```
package com.example;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Date;

public class SessionInfoServlet extends HttpServlet {
protected void doGet(HttpServletRequest request,
HttpServletResponse response)throws ServletException, IOException {
    HttpSession session = request.getSession();
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String sessionId = session.getId();
    Date creationTime = new Date(session.getCreationTime());
    Date lastAccessTime = new Date(session.getLastAccessedTime());

   out.println("<html><body>");
   out.println("<h2>Session Information</h2>");
   out.println("<p>Session ID: " + sessionId + "</p>");
   out.println("<p>Creation Time: " + creationTime + "</p>");
   out.println("<p>Last Accessed Time: " + lastAccessTime + "</p>");
```

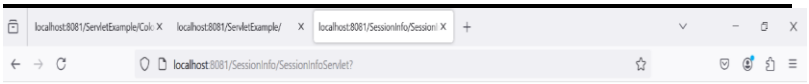
```
        out.println("</body></html>");
            }
        }
```

Output:



Check Session Information

Get Session Info



Session Information

Session ID: 6F97D46B68020B10DF56C6C3BBA74E38

Creation Time: Wed Oct 30 18:59:33 IST 2024

Last Accessed Time: Wed Oct 30 18:59:33 IST 2024

14. Program to request server information viz. Request Method, URI, Protocol and Remote address

serverinfo.html

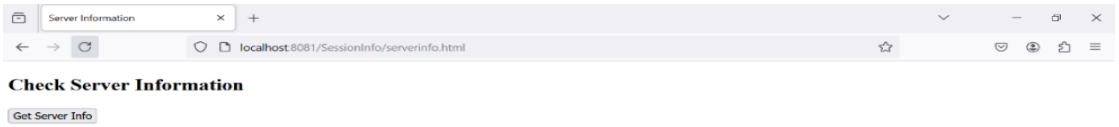
```
<html>
<head>
   <title>Server Information</title>
</head>
<body>
   <h2>Check Server Information</h2>
   <form action="ServerInfoServlet" method="get">
      <input type="submit" value="Get Server Info">
   </form>
</body>
</html>
```

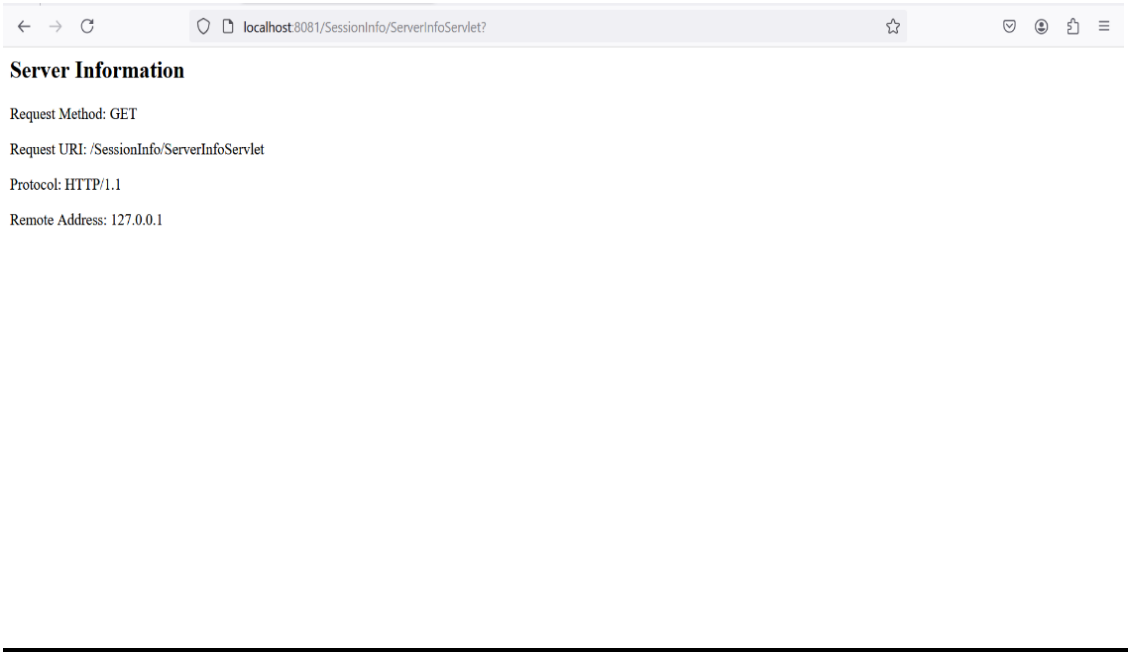
ServerInfoServlet.java

```
package com.example;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ServerInfoServlet extends HttpServlet {

protected void doGet(HttpServletRequest request,
HttpServletResponse response)throws ServletException, IOException {


      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      String method = request.getMethod();
      String uri = request.getRequestURI();
      String protocol = request.getProtocol();
      String remoteAddr = request.getRemoteAddr();

      out.println("<html><body>");
```

```
            out.println("<h2>Server Information</h2>");
            out.println("<p>Request Method: " + method + "</p>");
            out.println("<p>Request URI: " + uri + "</p>");
            out.println("<p>Protocol: " + protocol + "</p>");
            out.println("<p>Remote Address: " + remoteAddr + "</p>");
            out.println("</body></html>");
        }
    }
```

Output:





15. Program to accept Username and Address and display the webpage by passing parameters

userinfo.html

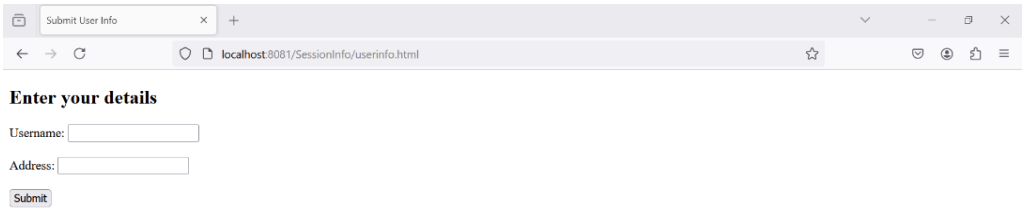
```
<html>
<head>
   <title>Submit User Info</title>
</head>
<body>
   <h2>Enter your details</h2>
   <form action="UserInfoServlet" method="post">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username"><br><br>

      <label for="address">Address:</label>
      <input type="text" id="address" name="address"><br><br>

      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

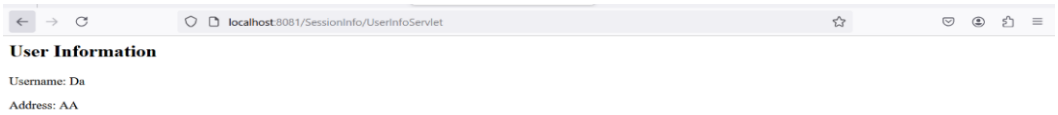
UserInfoServlet.java

```
package com.example;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class UserInfoServlet extends HttpServlet {

   protected void doPost(HttpServletRequest request,
HttpServletResponse response)
         throws ServletException, IOException {

      // Retrieve parameters from the HTML form
      String username = request.getParameter("username");
      String address = request.getParameter("address");
```

```
            // Set response content type
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            // Display user info on the webpage
            out.println("<html><body>");
            out.println("<h2>User Information</h2>");
            out.println("<p>Username: " + username + "</p>");
            out.println("<p>Address: " + address + "</p>");
            out.println("</body></html>");
        }
    }
```

Output:





display_query.pl

```
#!/usr/bin/perl
use strict;
use warnings;
use DBI;
use CGI qw(:standard);

# Database connection parameters
my $dsn = "DBI:mysql:database=yourdbname;host=localhost";
my $db_user = 'root';
my $db_password = 'password';

# Connect to the database
my $dbh = DBI->connect($dsn, $db_user, $db_password, {
RaiseError => 1, AutoCommit => 1 })
   or die "Couldn't connect to database: " . DBI->errstr;

# Prepare the SQL query
my $sth = $dbh->prepare("SELECT accession_number, title, author,
edition FROM books");
$sth->execute();

# Output HTML content
print header, start_html("Query Result"), h1("Book Information");


# Display the query result
print "<table border='1'><tr><th>Accession
Number</th><th>Title</th><th>Author</th><th>Edition</th></tr>";
while (my @row = $sth->fetchrow_array) {
   print
"<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row
[3]</td></tr>";
}
print "</table>";

# End the HTML page
print end_html;

# Clean up
```

```
$sth->finish();
$dbh->disconnect();
```

Output: