INTRODUCTION TO PHP PROGRAMMING

PHP, originally an acronym for ***"Personal Home Page" (Hypertext Preprocessor)*** is a widely-used open-source server-side scripting language that is particularly suited for web development and can be embedded into **HTML**. It was originally created by **Danish-Canadian** programmer ***Rasmus Lerdorf*** in 1994 and has since been continually developed by a large community of contributors.

The significance of PHP lies in its role as one of the most widely used languages for building dynamic web applications. Its simplicity, flexibility, and integration with various databases make it a preferred choice for developers worldwide.

PHP's history is marked by several milestones, including the release of PHP 3 in 1998, which introduced a full-fledged language with support for object-oriented programming. Subsequent versions, such as PHP 4 and PHP 5, brought significant improvements and enhancements, including better performance and security features.

In terms of functions, PHP offers a vast array of built-in functions for various tasks, such as string manipulation, file handling, database interaction, and more. Additionally, PHP supports user-defined functions, allowing developers to create custom functionality to suit their specific needs.

PHP supports various data types, including integers, floats, strings, booleans, arrays, and objects. These data types enable developers to work with different kinds of information efficiently.

Operations in PHP encompass a wide range of tasks, including arithmetic operations, string concatenation, comparison operations, logical operations, and more. These operations allow developers to manipulate data and control the flow of their applications effectively.

Applications of PHP span across numerous domains, including:

1. Web Development: PHP is extensively used for creating dynamic websites, content management systems (CMS), e-commerce platforms, and web applications.

2. Server-Side Scripting: PHP's server-side scripting capabilities enable developers to generate dynamic content, interact with databases, handle form submissions, and perform other server- side tasks.

3. Command-Line Scripting: PHP can also be used for command-line scripting, allowing developers to automate tasks and perform system administration tasks.

4. Frameworks and CMS: PHP frameworks like Laravel, Symfony, and CodeIgniter, as well as CMS platforms like WordPress, Drupal, and Joomla, leverage PHP's capabilities to streamline web development processes.

Overall, PHP's versatility, ease of use, and extensive community support have solidified its position as a cornerstone of web development for over two decades.

Introduction of Bus Ticket Generation System

A bus ticket generation system built with PHP typically involves the following :

1. User Interface: A web-based interface where passengers can input their travel details, such as departure location, destination, date, and number of passengers. This interface may also include options for selecting seats and purchasing tickets.
2. Ticket Generation Logic: PHP scripts responsible for processing user input, validating data, generating unique ticket identifiers, calculating fares, and formatting ticket information.
3. Database Integration: Integration with a database management system (e.g., MySQL, PostgreSQL) to store passenger details, ticket information, and transaction records. This enables the system to retrieve and update data as needed.
4. Payment Gateway Integration: Integration with payment gateways (e.g., PayPal, Stripe)

Key Features and Functionality

A robust bus ticket generation system developed with PHP may include the following features:

1. User Authentication: Registration and login functionality to manage user accounts and track ticket purchases.
2. Booking Management: Ability to search for available buses based on criteria such as departure location, destination, and date. Users should be able to select preferred seats and book tickets for desired routes.
3. Ticket Generation: Automated generation of unique ticket identifiers (e.g., QR codes) containing relevant travel information.

PHP BUS TICKET GENERATION CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Karnataka Government Bus Ticket</title>
    <style>
        .ticket {
            border: 1px solid #000;
            padding: 20px;
            width: 300px;
        }
        .title {
            text-align: center;
            font-size: 20px;
            margin-bottom: 20px;
        }
        .details {
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <?php
    // Function to calculate fare based on distance and number of persons
    function calculateFare($distance, $numPersons) {
        // Fare calculation logic (Example: Rs. 5 per km per person)
        $fare_per_km_per_person = 5;
        $total_fare = $distance * $numPersons * $fare_per_km_per_person;
        return $total_fare;
```

```
}

// Form submission handling
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get form inputs
    $from = $_POST["from"];
    $to = $_POST["to"];
    $distance = $_POST["distance"];
    $numPersons = $_POST["numPersons"];

    // Calculate fare
    $fare = calculateFare($distance, $numPersons);
?>
    <div class="ticket">
        <div class="title">Karnataka Government Bus Ticket</div>
        <div class="details">From: <?php echo $from; ?></div>
        <div class="details">To: <?php echo $to; ?></div>
        <div class="details">Distance: <?php echo $distance; ?> km</div>
        <div class="details">Number of Persons: <?php echo $numPersons; ?></div>
        <div class="details">Total Fare: Rs. <?php echo $fare; ?></div>
    </div>
<?php
}
<?
<?
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
    <label for="from">From:</label>
    <input type="text" id="from" name="from" required><br><br>
    <label for="to">To:</label>
    <input type="text" id="to" name="to" required><br><br>
    <label for="distance">Distance (in km):</label>
    <input type="number" id="distance" name="distance" required><br><br>
    <label for="numPersons">Number of Persons:</label>
```

```
        <input type="number" id="numPersons" name="numPersons" required><br><br>
        <input type="submit" value="Generate Ticket">
    </form>
</body>
</html>
```

OUTPUT 1:

Karnataka Government Bus Ticket

From Bangalore
To Mysore

Number of Persons 3
Distance (KMs) 150
Total Amount Rs 4500

OUTPUT 2:

Karnataka Government Bus Ticket

From Sakleshapura
To Mysore

Number of Persons 2
Distance (KMs) 250
Total Amount Rs 5000

LINE TO LINE EXPLANATION AND WORKING OF CODE:

- <!DOCTYPE html>: This declaration specifies the document type and version of HTML being used.
- <html lang="en">: Opening tag for the HTML document, specifying the language as English.
- <head>: This section contains meta-information about the document, such as the character encoding, viewport settings, and the document title.
- <meta charset="UTF-8">: Specifies the character encoding used in the document (UTF-8, which supports a wide range of characters).
- <meta name="viewport" content="width=device-width, initial-scale=1.0">: Sets the viewport properties for responsive design on various devices.
- <title>Bus Ticket</title>: Specifies the title of the HTML document, which appears in the browser's title bar or tab.
- <style>: This section contains CSS (Cascading Style Sheets) rules for styling the HTML elements within the document.
- body: CSS rule for styling the <body> element, setting the font family, margin, and padding to 0.
- .container: CSS rule for styling a container div, setting its maximum width, margin, padding, border, and border-radius.

o h2: CSS rule for styling <h2> elements, aligning text to the center.

o table: CSS rule for styling <table> elements, setting the width to 100% and collapsing borders.

- th, td: CSS rule for styling <th> and <td> elements within tables, setting border and padding properties.
- th: CSS rule for styling <th> elements within tables, setting a background color.
- </style>: Closing tag for the CSS rules.
- </head>: Closing tag for the <head> section.
- <body>: Opening tag for the document body, containing the content visible to users.
- : Opening tag for a element with the class "container", used for styling and organizing content.
- <h2>Karnataka Government Bus Ticket</h2>: Displays a level 2 heading with

the specified text.

- <table>: Opening tag for a table element to display ticket details.
- <tr>: Opening tag for a table row.
- <th>: Opening tag for a table header cell (bold and centered text).
- <td>: Opening tag for a table data cell (normal text).
- <?php echo isset($_POST['persons']) ? $_POST['persons'] :

Explanation:

This HTML document presents a form for users to input the number of persons traveling and the distance travelled. Upon submitting the form, PHP code processes the input, calculates the total amount based on the provided values and displays the ticket details along with the calculated amount in a structured table format.

CONCLUSION:

The provided HTML and PHP code implements a basic bus ticket generation system. It includes a form where users can input the number of persons traveling and the distance traveled. Upon submission, the PHP code calculates the total amount based on the provided inputs and displays the ticket details, including the source and destination, number of persons, distance, and the calculated amount. This simple yet functional system demonstrates how PHP can be used to generate dynamic content and handle user inputs effectively within a web application.