Portfolio.gen

Submitted By -- Bhupendra Kumar Ravi

Date of Submission
Nov 14, 2024

Instructor
Dr. Biresh

Abstract

The Portfolio Generator is a web application developed on the WAMP (Windows, Apache, MySQL, PHP) stack, with HTML, CSS, and Bootstrap on the front end, and PHP and MySQL handling the backend. It allows users to create personalized portfolios and resumes, offering an intuitive, user-friendly interface. The application leverages the FPDF library to generate high-quality PDF portfolios from user inputs, covering key personal, educational, and professional information. Future scope includes enhancing this application with the MERN stack for a more robust, dynamic experience and adding a mobile app interface using React Native.

Introduction

With the rising demand for online resumes and digital portfolios, having a streamlined tool for portfolio creation is essential. This project, the Portfolio Generator, aims to simplify and automate the resume-building process by enabling users to input their details and download a well-structured PDF portfolio. Designed on the WAMP stack, this web application combines the simplicity of HTML and CSS with the interactivity of PHP and MySQL, presenting users with a clear, concise tool for self-promotion.

Use Case

The Portfolio Generator is ideal for:

1. Job seekers needing a professional, easily customizable portfolio.
2. Students and recent graduates looking to build their first resume.
3. Professionals wanting a quick and reliable way to update their resumes with new skills or experiences.

Goals and Objectives

Goals

- To create an interactive web application that generates downloadable PDF portfolios based on user input.

Objectives

- Develop a user-friendly interface for data entry.
- Store user data securely in a MySQL database.
- Generate a professionally formatted PDF using FPDF.
- Enable future scalability with a potential migration to the MERN stack.

Thesis Statement

The *Portfolio Generator* empowers users by simplifying the process of creating personalized, high-quality portfolios and resumes. This application combines robust backend functionality with an intuitive frontend interface, ensuring accessibility and professionalism in the digital age.

Review of Literature

The growing need for automated portfolio tools has been well-documented in recent years. Research highlights that users value tools offering quick, personalized resume creation. However, many existing solutions lack offline accessibility or customizable features. The *Portfolio Generator* bridges these gaps by providing tailored portfolios in a downloadable PDF format, giving users full control over their data and presentation. This combination of customization and convenience makes the *Portfolio Generator* a unique and valuable tool for professionals and students alike.

Methodology

1. **Frontend Development**: HTML and Bootstrap provide a responsive, accessible interface, while CSS styles the application for an aesthetically pleasing experience.
2. **Backend Processing**: PHP captures user inputs and stores them in a MySQL database, ensuring data persistence. Additionally, the FPDF library is used to generate the downloadable PDF portfolio.
3. **Database Management**: MySQL is utilized to store user information, which can later be fetched or updated, supporting a seamless user experience.
4. **PDF Generation**: The FPDF library is used to create visually consistent, easily downloadable PDF documents formatted with professional styling.

Technology Used

Frontend: HTML, CSS, Bootstrap (for responsive design).

Backend: PHP, FPDF library (for PDF generation).

Server: Apache (running on Windows).

Database: MySQL (for data storage)

Development Process

The *Portfolio Generator* was developed using a step-by-step approach to ensure a clear workflow, robust implementation, and user-friendly design. Below is a detailed account of the development process:

1. Requirement Analysis

The first step was to understand the purpose of the application and the needs of its users. This involved identifying the core features, such as an intuitive user interface, secure data storage, and the ability to generate downloadable PDF portfolios.

- Research was conducted to examine existing tools, highlighting their shortcomings, such as limited customization options and the inability to download offline versions. These insights shaped the application's goals and functionality.

2. Planning and Design

The design phase focused on creating a roadmap for development:

- **Architecture Selection:**
  The WAMP stack (Windows, Apache, MySQL, PHP) was chosen for its compatibility with the development environment and its ability to manage both the frontend and backend. The FPDF library was selected for its ability to create customizable and professional PDF documents.
- **User Interface (UI) Design:**
  Using HTML, CSS, and Bootstrap, wireframes were created to visualize the layout and flow of the application. The goal was to build a clean, responsive, and easy-to-use interface that works well across all devices.
- **Database Planning:**
  MySQL was used to design a relational database structure that could securely store user information, such as personal details, education, skills, and work experience.

3. Development

The application was built in three main layers:

1. **Frontend Development:**
   - The interface was created using HTML for structure, CSS for styling, and Bootstrap for mobile responsiveness.
   - Forms were designed to allow users to input their data efficiently, with clear labels and validation features.
2. **Backend Development:**
   - PHP was employed to handle data processing and ensure seamless interaction between the frontend and the MySQL database.
   - Essential operations like storing, retrieving, updating, and deleting user data were implemented.
3. **Database Management:**
   - A MySQL database was developed to securely store user inputs.
   - Security measures, such as input sanitization and prepared statements, were implemented to protect against SQL injection and ensure data integrity.
4. **PDF Generation:**
   - The FPDF library was integrated to create well-structured PDF documents based on user inputs.

    - The PDF layout was customized to ensure a professional look, including sections for personal details, education, skills, and work experience.

4. Testing and Debugging

- **Functional Testing:** Each module was tested individually to confirm that the data entry, storage, and PDF generation processes worked correctly.
- **UI Testing:** The application's user interface was tested for responsiveness and usability across different devices and screen sizes.
- **Performance Testing:** The speed and efficiency of the application, especially during PDF generation, were evaluated.
- **Bug Fixing:** Detected issues in both the frontend and backend were addressed to ensure a smooth and error-free experience.

5. Deployment

After thorough testing, the application was deployed on a local WAMP server for final validation. Future plans include deploying the application on a cloud-based server to make it accessible to a broader audience.

6. Iterative Improvement

Feedback from initial users was collected to identify areas for enhancement. Based on this feedback, features like advanced template customization and real-time updates are planned for future iterations. Additionally, the potential migration to the MERN stack and the development of a mobile app interface using React Native are under consideration to improve usability and scalability.

Code

Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Portfolio Generator</title>
  <style>
    h1 {
      color: white;
      padding-left: 10%;
    }
    .navbar {
      height: 150px;
    }
    .footer {
      background-color: #343a40;
      color: white;
```

```
            padding: 20px 0;
            text-align: center;
            height: 200px;
            margin-top: 30px;
        }
        .navbar-nav {
            font-size: 30px;
        }
        .navbar-nav .nav-link {
            color: white;
            text-shadow: 2px 2px #fdfbfb;
            margin-left: 20px;
        }
        #navbarNav{
            margin-left: 20%;
        }
    </style>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" integrity="sha512-pd39Z3Ss+5NbjG8/83w6dyUHvt+vXQIb0PAtfjNj9Xk5o5VioAFDT9s8NUQ7G6V0+eZ+Y5HQuQwYmuRd49tGg==" crossorigin="anonymous" referrerpolicy="no-referrer" />
</head>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand">
                <h1>Portfolio.Gen</h1>
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="index.html">Generator</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="about.html">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.html">Contact</a>
                    </li>
                    <li class="nav-item active">
                        <a class="nav-link" href="dev.html">Developer</a>
                    </li>
                </ul>
            </div>
    </div>
</nav>

<form action="index.php" method="POST">
<div class="container mt-4">
    <div class="row">
        <div class="col-md-6">
            <div class="card bg-dark text-white">
                <div class="card-body">
                    <h3 class="card-title">Personal Data</h3>

                        <div class="mb-3">
                            <label for="name" class="form-label">Name:</label>
                            <input type="text" class="form-control" id="name" name="name" required>
                        </div>
                        <div class="mb-3">
                            <label for="email" class="form-label">Email:</label>
                            <input type="email" class="form-control" id="email" name="email" required>
                        </div>
                        <div class="mb-3">
                            <label for="dob" class="form-label">Date of Birth:</label>
                            <input type="date" class="form-control" id="dob" name="dob" required>
                        </div>
                        <div class="mb-3">
                            <label for="phone" class="form-label">Phone Number:</label>
                            <input type="tel" class="form-control" id="phone" name="phone" required>
                        </div>
                        <div class="mb-3">
                            <label for="address" class="form-label">Address:</label>
                            <textarea class="form-control" id="address" name="address" rows="3"></textarea>
                        </div>

                </div>
            </div>
        </div>

        <div class="col-md-6">
            <div class="card bg-dark text-white">
                <div class="card-body">
                    <h3 class="card-title">Skills</h3>
                    <div class="mb-3">
                        <label for="html">HTML :</label>
                        <input type="range" id="html" name="html" min="0" max="100" value="50" class="form-range">
                    </div>
                            <div class="mb-3">
                        <label for="css">CSS :</label>
                        <input type="range" id="css" name="css" min="0" max="100" value="50" class="form-range">
                    </div>

                    <div class="mb-3">
                        <label for="js">JavaScript :</label>
                        <input type="range" id="js" name="js" min="0" max="100" value="50" class="form-range">
                    </div>
```

```html
                    <div class="mb-3">
                      <label for="php">PHP :</label>
                      <input type="range" id="php" name="php" min="0" max="100" value="50" class="form-range">
                    </div>

                    <div class="mb-3">
                      <label for="sql">SQL :</label>
                       <input type="range" id="sql" name="sql" min="0" max="100" value="50" class="form-range">
                    </div>
                    <div class="form-group">
                      <label for="personal_url">Personal Website URL:</label>
                      <input type="url" class="form-control" id="personal_url" name="personal_url" >
                    </div>
                  </div>
                </div>
              </div>
            </div>

            <div class="row mt-4">
              <div class="col-md-6">
                <div class="card bg-dark text-white">
                  <div class="card-body">
                    <h3 class="card-title">Educational Background</h3>
                    <div class="mb-3">
                      <label for="high_school" class="form-label">High School:</label>
                      <input type="text" class="form-control" id="high_school" name="high_school" required>
                    </div>
                    <div class="mb-3">
                      <label for="diploma" class="form-label">Diploma:</label>
                      <input type="text" class="form-control" id="diploma" name="diploma">
                    </div>
                    <div class="mb-3">
                      <label for="bachelor_degree" class="form-label">Bachelor's Degree:</label>
                      <input type="text" class="form-control" id="bachelor_degree" name="bachelor_degree">
                    </div>
                    <div class="mb-3">
                      <label for="master_degree" class="form-label">Master's Degree:</label>
                      <input type="text" class="form-control" id="master_degree" name="master_degree">
                    </div>

                    <div class="mb-3">
                      <label for="max_education" class="form-label">Max Education:</label>
                      <select class="form-select" id="max_education" name="max_education" required>
                        <option value="High School">High School</option>
                        <option value="Diploma">Diploma</option>
                        <option value="Bachelor's Degree">Bachelor's Degree</option>
                        <option value="Master's Degree">Master's Degree</option>
                        <option value="Ph.D.">Ph.D.</option>
                      </select>
                    </div>
                    <div class="mb-3">
                      <button type="submit" class="btn btn-primary">Generate Portfolio</button>
                    </div>
                </div>
            </div>
        </div>

        <div class="col-md-6">
            <div class="card bg-dark text-white ">
                <div class="card-body">
                    <h3 class="card-title">Hobbies</h3>
                    <div class="mb-3">
                        <label for="Hobbies" class="form-label">Hobbie:</label>
                        <input type="text" class="form-control" id="Hobbies" name="Hobbies" required>
                    </div>

                    <h3 class="card-title">Languages</h3>
                    <!– Language –>

                     <div class="mb-3">
                        <label for="Language1" class="form-label">Language1:</label>
                         <input type="text" class="form-control" id="Language1" name="Language1" required>
                    </div>
                    <div class="mb-3">
                         <label for="Language2" class="form-label">Language2:</label>
                         <input type="text" class="form-control" id="Language2" name="Language2" required>
                    </div>

              <div class="form-group">
                <label for="about">About:</label>
               <textarea class="form-control" id="about" name="about" rows="4" required></textarea>
            </div>

                </div>
            </div>
        </div>
    </div>

</form>

    </div>
  </div>
  <div class="footer ">

        <p>
            Developer: Bhupendra Kumar Ravi |
            <a href="https://github.com/your-github-username" target="_blank"><i class="fa-brands fa-github"></i></a>
            <a href="https://www.linkedin.com/in/your-linkedin-username" target="_blank"><i class="fa-brands fa-linkedin"></i></a>
            <a href="https://plus.google.com/your-google-username" target="_blank"><i class="fab fa-google"></i></a>
            | Contact: ravikumar898911@gmail.com
        </p>
</div>
```

```
</body>
</html>

```

Index.php

```
<?php

require_once('fpdf186/fpdf.php');

// Retrieving form data

$name = $_POST['name'];

$email = $_POST['email'];

$dob = $_POST["dob"];

    $phone = $_POST["phone"];

    $html = $_POST['html'];

    $css = $_POST['css'];

    $js = $_POST['js'];

    $php = $_POST['php'];

    $sql = $_POST['sql'];

     $highSchool = $_POST['high_school'];

     $diploma = $_POST['diploma'];

     $maxEducation = $_POST['max_education'];

     $bachelor_degree = isset($_POST["bachelor_degree"]) ? $_POST["bachelor_degree"] : '';

     $master_degree = isset($_POST["master_degree"]) ? $_POST["master_degree"] : '';

     $hobbies = $_POST["Hobbies"];

$language1 = $_POST['Language1'];

$language2 = $_POST['Language2'];

$about = $_POST['about'];

$personal_url = $_POST['personal_url'];


// Generating PDF
```

```
$pdf = new FPDF();
   $pdf->AddPage();


    // Setting background color
    $pdf->SetFillColor(255, 255, 255);


    // Filling the entire page with black color
    $pdf->Rect(0, 0, $pdf->GetPageWidth(), $pdf->GetPageHeight(), 'F');



    // Seting the background area
    $pdf->SetFillColor(0,0,0);
    $pdf->Rect(0, 0, $pdf->GetPageWidth() * 0.3, $pdf->GetPageHeight(), 'F');
$pdf->SetTextColor(255, 255, 255);



$pdf->SetX(5);
$pdf->SetFont('Arial', 'B', 14);
$pdf->Cell(0, 10, 'Skills Section', 0, 1);



function displayRangeBar($rating)
{
   global $pdf;
   $barWidth = $rating * 0.5;
   $barHeight = 6;
   $pdf->SetDrawColor(0, 0, 0);
   $pdf->SetFillColor(135, 206, 250);
   $pdf->Rect($pdf->GetX(), $pdf->GetY(), $barWidth, $barHeight, 'F');
}
```

```
$pdf->SetX(5);

$pdf->SetFont('Arial', '', 12);

$pdf->Cell(0, 10, "HTML: $html", 0, 1);

$pdf->SetX(5);

displayRangeBar($html);


$pdf->Ln(10);


$pdf->SetX(5);

$pdf->Cell(0, 10, "CSS: $css", 0, 1);

$pdf->SetX(5);

displayRangeBar($css);


$pdf->Ln(10);


$pdf->SetX(5);

$pdf->Cell(0, 10, "JavaScript: $js", 0, 1);

$pdf->SetX(5);

displayRangeBar($js);


$pdf->Ln(10);


$pdf->SetX(5);

$pdf->Cell(0, 10, "PHP: $php", 0, 1);

$pdf->SetX(5);

displayRangeBar($php);


$pdf->Ln(10);


$pdf->SetX(5);
```

```
$pdf->Cell(0, 10, "SQL: $sql", 0, 1);

$pdf->SetX(5);

displayRangeBar($sql);


$pdf->Ln(10);

$pdf->Ln(20);


$pdf->SetX(5);

$pdf->SetFont('Arial', 'B', 14);

$pdf->Cell(0, 10, 'Hobbies', 0, 1);

$pdf->SetX(5);

$pdf->SetFont('Arial', 'B', 12);

$pdf->Cell(0, 10, "$hobbies", 0, 1);


if (!empty($language1) || !empty($language2)) {

    $pdf->Ln(10);

    $pdf->SetX(5);

    $pdf->SetFont('Arial', 'B', 14);

    $pdf->Cell(0, 10, 'Languages', 0, 1);

    $pdf->SetX(5);

    $pdf->SetFont('Arial', '', 12);

    $pdf->Cell(0, 10, "Language1: $language1", 0, 1);

    if (!empty($language2)) {

        $pdf->SetX(5);

        $pdf->Cell(0, 10, "Language2: $language2", 0, 1);

    }

}

 $rightSideX = $pdf->GetPageWidth() * 0.35;
 $rightSideY = 5; // Set the Y position to 15

 $pdf->SetXY($rightSideX,$rightSideY);

 $pdf->SetTextColor(0, 0, 0);

 $pdf->SetFont('Arial', 'B', 16);

$pdf->Cell(0, 10, 'Portfolio', 0, 1, 'C');

$pdf->Ln(10);

$pdf->SetFont('Arial', '', 12);

$pdf->SetX($rightSideX);

 $pdf->Cell(0, 10, "Name: $name", 0, 1);



 $pdf->SetX($rightSideX);

 $pdf->Cell(0, 10, "Email: $email", 0, 1);



 $pdf->SetX($rightSideX);

 $pdf->Cell(0, 10, "Date of Birth: $dob", 0, 1);



 $pdf->SetX($rightSideX);

 $pdf->Cell(0, 10, "Phone Number: $phone", 0, 1);


 $pdf->SetX($rightSideX);

 $pdf->SetFont('Arial', 'B', 14);

 $pdf->Cell(0, 10, 'Educational Background', 0, 1);



 $pdf->SetX($rightSideX);

 $pdf->SetFont('Arial', '', 12);

 $pdf->Cell(0, 10, "High School: $highSchool", 0, 1);
```

```
// Addind some space between High School and Diploma

if (!empty($diploma)) {

    $pdf->Ln(10);

    $pdf->SetX($rightSideX);

    $pdf->Cell(0, 10, "Diploma: $diploma", 0, 1);

}

$pdf->Ln(10);

$pdf->SetX($rightSideX);

$pdf->Cell(0, 10, "Max Education: $maxEducation", 0, 1);

$pdf->Ln(10);

$pdf->SetX($rightSideX);

$pdf->Cell(0, 10, "Bachelor's Degree: $bachelor_degree", 0, 1);

$pdf->Ln(10);

$pdf->SetX($rightSideX);

$pdf->Cell(0, 10, "Master's Degree: $master_degree", 0, 1);
$pdf->Ln(10);

$pdf->SetX($pdf->GetPageWidth() * 0.35);

$pdf->SetFont('Arial', 'B', 12);

$pdf->Cell(0, 10, "About MySelf: $about", 0, 1,);
$pdf->Ln(10);

$pdf->SetX($pdf->GetPageWidth() * 0.35);

$pdf->Cell(0, 10, "Personal Website URL: $personal_url", 0, 1);

// Storing user data in MySQL database

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "portifolio_data";
```

```
 // Createing a connection to the database

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


 // Checking connection

 if ($conn->connect_error) {

     die("Connection failed: " . $conn->connect_error);

 }


 // Inserting user data into the database

 $sql = "INSERT INTO user_data (name, email, PhoneNo) VALUES ('$name', '$email', '$phone')";




 if ($conn->query($sql) === TRUE) {

    echo "User data stored successfully.";

} else {

    echo "Error: " . $sql . "<br>" . $conn->error;

}



// Output of PDF as a downloadable file

$pdf->Output('portfolio.pdf', 'F');


// Displaying portfolio on the webpage

echo "<h2>Your Portfolio:</h2>";

echo "<embed src='portfolio.pdf' type='application/pdf' width='100%' height='100%'>";



echo "<div id='download-button-container'>";

echo "<a href='portfolio.pdf' download='portfolio.pdf' id='download-button'>Download Portfolio PDF</a>";
echo "</div>";

?>
```

Key Findings

1. A significant gap exists in the availability of tools that offer a balance between customization and offline accessibility for portfolio creation.
2. The WAMP stack effectively provides a solid foundation for creating lightweight, functional applications.
3. FPDF proves to be a reliable library for generating high-quality PDFs.
4. Expanding the application to the MERN stack and mobile platforms could significantly enhance usability and performance.

Conclusion

The *Portfolio Generator* is a practical, efficient tool for creating personalized resumes and portfolios. It simplifies the often-complex process of resume building, catering to a wide range of users from students to seasoned professionals. With its robust functionality and planned future enhancements, the application has the potential to become a comprehensive solution for portfolio creation in both web and mobile environments.

Future Scope

1. **MERN Stack Implementation**: Migrating the Portfolio Generator to the MERN (MongoDB, Express, React, Node.js) stack can significantly enhance performance, enabling real-time updates and a more dynamic user interface.
2. **Mobile Application Development**: Developing a mobile application using React Native would provide greater accessibility, allowing users to create and download portfolios on their mobile devices.
3. **Advanced Customization**: Adding features such as customizable templates and layout options would provide users with greater flexibility in designing their portfolios.

Resources

The development of the **Portfolio Generator** relied on various tools, technologies, and libraries to achieve its objectives. This section outlines the critical resources used in the project and their contributions.

1. Development Tools

WAMP Stack (Windows, Apache, MySQL, PHP):
The WAMP stack was chosen as the development platform, providing an integrated environment for web development.

Windows: Operating system used to host and manage the development environment.

Apache: Web server used to host the application and manage HTTP requests.

MySQL: Relational database management system (RDBMS) for storing user data securely.

PHP: Server-side scripting language for backend logic, form handling, and PDF generation.

Text Editors and IDEs:

Tools such as **Visual Studio Code** and **Sublime Text** were used for efficient coding, syntax highlighting, and debugging.

2. Frontend Technologies

HTML (Hypertext Markup Language):
Used for structuring the content and user interface.

CSS (Cascading Style Sheets):
Applied for styling the interface and ensuring a responsive design.

Bootstrap:
Integrated for creating a mobile-friendly, responsive layout with pre-designed components.

3. Backend Technologies

PHP:
Key scripting language for server-side processing, including form handling, database operations, and dynamic content generation.

MySQL:
Database used for secure storage and retrieval of user data, ensuring persistence and scalability.

FPDF Library:
A PHP library utilized for creating customized, downloadable PDF documents programmatically. It facilitated features such as:

Dynamic text placement.

Skill-based range bars.

Structuring and formatting the portfolio content.

4. Database Resources

MySQL Database:
Schema design focused on storing user information efficiently. Tables included fields for personal details, educational background, technical skills, and hobbies.

5. Web Hosting and Testing

Localhost Testing:
During development, the application was hosted on a local Apache server to test functionality in a controlled environment.

Browser-Based Testing:
Modern web browsers, such as Chrome, Firefox, and Edge, were used to test the application's user interface and functionality across platforms.

6. Research and Documentation

Official Documentation:

FPDF: http://www.fpdf.org/

Bootstrap: https://getbootstrap.com/

PHP: https://www.php.net/

Online Resources:
Blogs, forums, and tutorials, such as Stack Overflow and GitHub repositories, were consulted to troubleshoot challenges and enhance functionality.

7. Version Control

Git:
Git was used for version control to track code changes and collaborate during the development process.

8. Testing Tools

PHPUnit:
For testing backend functionalities.

Browser Developer Tools:
Used for debugging CSS and JavaScript issues and optimizing the user experience.