LSDB

System Requirements Specifications

16th November 2023

London School of
DIGITAL BUSINESS

Table Of Contents

1.Introduction

1.1 Purpose:

The purpose of this website is to create a compelling online presence that reinforces the positive impression individuals have of Graham when they meet him in person or when referred to him. The website aims to be remarkable, encouraging visitors to engage with its content, share it, and generate positive word-of-mouth. It will serve as a warm, welcoming, and professional platform that reflects Graham's personality while providing clear information about his expertise and services.

1.2 Scope:

The website will consist of several key pages, each serving a specific purpose to enhance the user experience and provide comprehensive information about Graham and his offerings. The scope includes the creation of a simple home page, integration with social media platforms, a visual journey map highlighting Graham's signature workshop and coaching services, a professional page featuring Graham's background and personal story, a services page detailing his expertise, a testimonials page showcasing positive feedback, teaser content through videos and articles, and a connect page with an appointment booking system for easy communication.

1.3 Roles and Responsibilities:

1. **Content Creation:**
    - Develop engaging and informative content for each page, ensuring consistency in tone and messaging.
2. **Design and Visuals:**
    - Create visually appealing layouts, incorporating high-quality images that align with the warm and welcoming theme.
    - Ensure the website design is user-friendly and responsive across various devices.
3. **Integration with Social Media:**
    - Establish links to Graham's social media profiles and integrate social media feeds or widgets.
4. **Journey Map:**
    - Design a visual representation of the journey to Graham's signature workshop and coaching services.
5. **Professional Page and Personal Story:**
    - Craft a professional bio and personal story that resonate with visitors, providing insight into Graham's background and personality.
6. **Services Offered:**
    - Develop detailed descriptions of Graham's expertise and the services he provides.

7. **Testimonials:**
    - Collect and present testimonials from individuals who have experienced Graham's training and coaching.

8. **Teaser Content:**
    - Create and showcase teaser videos of training material, blog posts, and articles.
9. **Connect Page:**
    - Implement an appointment booking system for seamless communication.
    - Provide clear contact information and a user-friendly form for general inquiries.

2.SystemRequirements Specifications

2.1 System Description: The system is a professional website for Graham, incorporating Angular for the frontend, PHP for the backend, and MySQL as the database.

2.2 Key Features:

- Home page with a welcoming design.
- Integration with social media.
- Visual journey map for workshops and coaching.
- Professional and personal story page.
- Services page with detailed offerings.
- Testimonials section.
- Teaser content with videos and articles.
- Connect page with an appointment booking system.

3. Software Requirements

3. Frontend Technology:

3.1 **Framework:** Angular will be used for the frontend development.

3.2 **User Interface:** The UI will be designed to reflect warmth, professionalism, and engagement. It will be responsive across various devices.

3.3 **Interactivity:** Angular will be utilized to create dynamic and interactive components for an enhanced user experience.

4. Backend Technology:

4.1 **Language:** PHP will be used as the backend programming language.

4.2 **Server:** The backend will be hosted on a PHP-supported server.

4.3 **Functionality:** PHP will handle server-side processing, including data retrieval and business logic.

5. Database:

5.1 **Type:** MySQL will be used as the relational database management system.

5.2 **Data Storage:** MySQL will store user data, testimonials, service information, and other relevant content.

6. Functional Requirements:

6.1 **Home Page:**

- Display welcoming content.
- Responsive design.

6.2 **Integration with Social Media:**

- Links to social media profiles.
- Integration with social media feeds.

6.3 **Journey Map:**

- Visual representation of workshop and coaching journeys.

6.4 **Professional Page and Personal Story:**

- Professional biography.
- Personal story section.

6.5 **Services Offered:**

- Detailed service descriptions.

6.6 **Testimonials:**

- Display client testimonials.

6.7 **Teaser Content:**

- Video snippets and article excerpts.

6.8 **Connect Page:**

- Contact information.
- Appointment booking system.

7. Non-Functional Requirements:

7.1 **Performance:**

- Fast loading times.
- Optimized media files.

7.2 **Scalability:**

- Design for future content additions.
- Handle increased user traffic.

7.3 **Security:**

- SSL encryption.
- Regular security updates.

7.4 **Compatibility:**

- Cross-browser compatibility.
- Device compatibility.

7.5 **Accessibility:**

- Compliance with accessibility standards.

7.6 **User Experience:**

- Intuitive and user-friendly interface.

7.7 **Backup and Recovery:**

- Regular backups.
- Recovery plan in case of data loss.

8. Constraints:

- Budget constraints may impact the selection of certain technologies or features.

4.Hardware Specifications

1. Web Server:

1.1 **Type:**

- Dedicated or cloud-based server.

1.2 **Processor:**

- Multi-core processor for handling concurrent requests.

1.3 **RAM:**

- Adequate RAM to support the concurrent users and ensure fast response times.

1.4 **Storage:**

- SSD storage for faster data access.

2. Database Server:

2.1 **Type:**

- Dedicated or cloud-based server for hosting the MySQL database.

2.2 **Processor:**

- Multi-core processor for efficient database operations.

2.3 **RAM:**

- Sufficient RAM to handle database queries and optimize performance.

2.4 **Storage:**

- SSD storage for quick access to database records.

3. Load Balancer (Optional):

3.1 **Type:**

- If anticipating high traffic, consider a load balancer to distribute requests across multiple servers.

4. Network:

4.1 **Bandwidth:**

- High-speed internet connection to ensure quick data transfer.

4.2 **Firewall:**

- Implement a firewall for security.

5.Security Specifications

1. Web Application Security:

1.1 **SSL Encryption:**

- Implement SSL/TLS encryption to secure data transmission between the client and the server.

1.2 **Secure Coding Practices:**

- Adhere to secure coding standards, such as input validation and output encoding, to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS).

1.3 **Content Security Policy (CSP):**

- Utilize CSP headers to mitigate the risks of content injection attacks.

2. Server Security:

2.1 **Regular Updates:**

- Keep the operating system, web server (e.g., Apache, Nginx), and other software up-to-date with the latest security patches.

2.2 **Firewall:**

- Implement a firewall to control incoming and outgoing traffic, blocking unauthorized access.

2.3 **Intrusion Detection System (IDS):**

- Deploy an IDS to monitor and detect any suspicious activities on the server.

2.4 **SSH Key Authentication:**

- Use SSH key-based authentication instead of password-based authentication for secure server access.

3. Database Security:

3.1 **Secure Database Configuration:**

- Configure the MySQL database securely, including strong authentication and access controls.

3.2 **Regular Database Audits:**

- Conduct regular audits of the database to identify and address potential security issues.

3.3 **Data Encryption:**

- Implement encryption for sensitive data stored in the database.

4. User Authentication and Authorization:

4.1 **Secure Authentication Mechanism:**

- Implement strong password policies and consider multi-factor authentication for added security.

4.2 **Role-Based Access Control (RBAC):**

- Use RBAC to control access to different parts of the application based on user roles.

5. Data Protection:

5.1 **Data Backups:**

- Regularly backup website data, and ensure backups are stored securely.

5.2 **Data Loss Prevention (DLP):**

- Implement DLP measures to prevent accidental or unauthorized data leaks.

6. Monitoring and Logging:

6.1 **Security Monitoring:**

- Set up continuous security monitoring to detect and respond to potential security incidents.

6.2 **Logging:**

- Implement logging mechanisms for tracking and analyzing system and application activities.

7. Third-Party Integrations:

7.1 **Vetted Libraries and Components:**

- Only use well-vetted and secure third-party libraries and components.

7.2 **Regular Updates:**

- Keep third-party components up-to-date with the latest security patches.

8. Compliance:

8.1 **Regulatory Compliance:**

- Ensure compliance with relevant data protection regulations and standards.

9. Incident Response Plan:

9.1 **Incident Response Team:**

- Establish an incident response team and define clear procedures for responding to security incidents.

9.2 **Communication Plan:**

- Develop a communication plan to notify stakeholders in the event of a security breach.

10. Training and Awareness:

10.1 **Security Training:**

- Provide security training for development and operational teams to raise awareness of potential threats and best practices.

6. Data Specifications

1. User Data:

1.1 **Types of User Data:**

- Personal information (name, contact details).
- Booking information for appointments.
- User preferences and settings.

1.2 **Data Storage:**

- Store user data securely in the MySQL database.

1.3 **Data Encryption:**

- Implement encryption for sensitive user information, especially personal details.

1.4 **Data Retention Policy:**

- Define a data retention policy to manage the storage duration of user data.

2. Workshop and Coaching Data:

2.1 **Journey Map Information:**

- Details of workshops, coaching sessions, and tailored solutions.

2.2 **Data Storage:**

- Organize and store journey map data in the database.

2.3 **Visual Representation:**

- Use appropriate data structures to represent the visual journey map.

3. Professional and Personal Story Data:

3.1 **Professional Bio:**

- Textual information about Graham's professional background.

3.2 **Personal Story:**

- Narrative content providing a personal touch.

3.3 **Data Storage:**

- Store this content in a structured format in the database.

4. Services Data:

4.1 **Service Descriptions:**

- Detailed information about Graham's expertise and services.

4.2 **Data Storage:**

- Maintain a database table for service-related information.

5. Testimonials:

5.1 **Testimonial Content:**

- Textual content of what clients say about Graham's training.

5.2 **Attribution Information:**

- Names, photos, or identifiers of clients providing testimonials.

5.3 **Data Storage:**

- Store testimonials in a structured format in the database.

6. Teaser Content:

6.1 **Video Snippets and Article Excerpts:**

- Short video clips and textual excerpts.

6.2 **Data Storage:**

- Store multimedia content and textual excerpts in a format suitable for efficient retrieval.

7. Connection and Booking Data:

7.1 **Contact Information:**

- User-provided contact details.

7.2 **Appointment Booking Information:**

- Details of scheduled appointments.

7.3 **Data Storage:**

- Securely store user contact information and appointment details in the database.

8. Security and Compliance Data:

8.1 **Security Logs:**

- Logs containing security-related events.

7.Network Impact Specifications

1. Bandwidth Requirements:

1.1 **Content Delivery:**

- Optimize images, videos, and other media for efficient content delivery.

1.2 **Caching Mechanisms:**

- Implement caching mechanisms to reduce the need for repetitive data downloads.

1.3 **Content Compression:**

- Use compression techniques for textual content to minimize bandwidth usage.

2. Latency Considerations:

2.1 **Content Delivery Networks (CDN):**

- Utilize CDNs to distribute content across servers geographically, reducing latency.

2.2 **Minimize HTTP Requests:**

- Limit the number of HTTP requests by optimizing frontend assets.

3. Server-Side Impact:

3.1 **Server Response Time:**

- Optimize server-side processing to ensure fast response times.

3.2 **Load Balancing (if applicable):**

- Implement load balancing to distribute incoming traffic across multiple servers.

4. Security Measures:

4.1 **SSL/TLS Overhead:**

- Consider the overhead introduced by SSL/TLS encryption for secure data transmission.