SCANDINAVIAN CERTIFICATION SERVICES

SC SERVICES STORE

Booking site with product management and multilingualism

Summary

Målet är att utveckla en mångsidig bokningssite för certifieringar och försäljning av relaterade produkter som utbildningsmaterial, övningstester och certifieringsvouchers. Webbplatsen ska vara flerspråkig, med stöd för svenska, danska, norska och finska. Maskinöversättning ska användas för att hantera översättningar, med svenska som huvudspråk. Projektet inkluderar integrationer med Stripe för betalningar och Freshdesk för helpdesk-funktioner.

Peter Robinson

peter.robinson@scservices.se

Specification of requirements for development projects: SC Services Store

1. Overview

Målet är att utveckla en mångsidig bokningssite för certifieringar och försäljning av relaterade produkter som utbildningsmaterial, övningstester och certifieringsvouchers. Webbplatsen ska vara flerspråkig, med stöd för svenska, danska, norska och finska. Maskinöversättning ska användas för att hantera översättningar, med svenska som huvudspråk. Projektet inkluderar integrationer med Stripe för betalningar och Freshdesk för helpdesk-funktioner.

2. Functional requirements

2.1. User Management

- **Admin login:** Admins should be able to log in to manage bookings, products, and multilingual content.
- **Admin privileges**: Only admins should be able to add, update, or remove available dates, products, and language versions.

2.2. Booking management

- **View available dates**: Users should be able to see a calendar with available dates for certifications.
- **Book dates**: Users should be able to select a date and fill out a booking form with the necessary details such as name, email, and phone number.
- **Booking confirmation**: After booking and payment, the user should receive a confirmation on screen and via email.

2.3. Product Handling

- **Product catalog**: The website should have a section to display and sell products such as training materials, practice tests, and certification vouchers.
- **Product information**: Each product should have a detailed description, price, and availability.
- **Cart and checkout**: Users should be able to add products to a cart and complete the purchase through a checkout.

2.4. Payment Processing

- **Stripe integration**: The website should use Stripe to process payments for bookings and products.
- **Payment confirmation**: After completing the payment, the payment status should be updated and the user should receive a payment confirmation.

2.5. Multilingualism

- **Language support**: The website will support Swedish, Danish, Norwegian and Finnish.
- **Machine translation**: Machine translation should be used to translate content from Swedish to other languages.
- **User language selection**: Users shall be able to select their preferred language via a language selector on the website.

2.6. Integrations

- **Freshdesk integration**: Integration with Freshdesk to offer help desk and support services.

3. Non-functional requirements

3.1. Performance

- **Response times**: The website should load pages within 2 seconds under normal load.
- **Scalability**: The solution should be scalable to handle a growing volume of bookings, products, and users.

3.2. Usability

- **Responsive design**: The website should be user-friendly and work well on various devices, including computers, tablets, and mobiles.
- **Intuitive interface**: The user interface should be easy to navigate and use for both administrators and users.

3.3. Accessibility

- **WCAG 2.1**: The website should follow the guidelines of WCAG 2.1 to ensure accessibility for all users.

4. Technical infrastructure

4.1. Hosting and server environment

- **Platform**: Any hosting platform
- **Web server**: Support for PHP and MySQL
- **Databas**: MySQL
- **Language**: PHP 7.4 or higher

4.2. Development tools

- **Payments**: Stripe for payment processing
- **Help Desk**: Freshdesk for Customer Support
- **Versioning**: Git

5. Security

5.1. Datasker

- **Encryption**: Use HTTPS for secure data transfer.
- **CSRF protection**: Implement CSRF tokens in all forms.
- **Remediation and validation**: Validate and sanitize all user inputs to prevent XSS and SQL injection.

5.2. Permissions

- **Admin management**: Implement a system for admin sign-in and admin role management.

6. Design och UX

6.1. Designprocess

- **UI/UX Design**: Create user-friendly interface and navigation flow.
- **Graphic Design**: Create graphics, logos, and other visual elements.

6.2. User tests

- **Test the user flow**: Conduct user testing to ensure that the website is user-friendly.

7. Documentation and Support

7.1. Technical documentation

- **Installation Guide**: Include step-by-step instructions for setting up and operating the system.
- **User documentation**: Include manuals for administrators and end users.

7.2. Support

- **Maintenance**: Plan for regular maintenance and updates.
- **Support**: Possibility of technical support in case of problems or questions.

8. Schedule and delivery

8.1. Timetable

- **Project start**: [Ange startdate]
- **Design and Prototype**: [Enter Date]
- **Development and Implementation**: [Enter Date]
- **Testing and Bug Fixing**: [Enter Date]
- **Delivery**: [Ange date]

8.2. Delivery

- **Deliverables**: Website source code, installation manuals, user manuals, and any graphics.

9. Quality requirements

9.1. Testing

- **Functional tests**: Test all functional requirements.
- **Security testing**: Conducting penetration tests and security briefings.
- **Performance tests**: Test the performance of the system under different loads.

9.2. Quality assurance

- **Code review**: Perform code reviews to ensure the quality and security of the code.

- **User testing**: Conducting user testing to ensure that the website meets the needs of users.

Questions answered:

1. Any dates we need to consider? Maybe for a demo or a mid-project progress report.

not really but i suggest that you create milstones and share with me

2. Hosting: Why not use Azure?

Much more expensive

3. Machine Translation? Are we going to use AI to translate the pages or is there something else? If use Azure, we can use Microsoft AI Translation Module

Yes AI, maybe Azure Cog. services can used anyway?

4. What kind of do you have a preference when it comes to the front end development language? HTML/CSS/Bootstrap, React, PHP?

your choice

5. Do we need to use PHP? It would cause delays in the project due that we are not trained in PHP.

.Net is fine

6. FreshDesk Integration / Account registration, or use current system?

Current

7. Intergration ready for CertiPort or PearsonVue?

no need

8. Does the current website contain all the information the user needs? Eg. Product description, price list etc etc.?

not really

9. Should all the products belong to one product category?

nope

10. Is there a database currently in use?

No

11. Are you already using Stripe as a payments system?

Yes

12. What is the budget for setting up and maintaining the webpage? E.g. server costs.

I will choose a plan at simply.com and that will be the budget
Pro or Standard it will be.