**Functional Specification Document**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the functional requirements and design of the Online Bookstore project. This project aims to create an e-commerce platform for selling books online, providing users with a seamless and efficient shopping experience.

**1.2 Scope**

This document covers the development of the Online Bookstore, including user registration, book browsing, shopping cart functionality, order processing, and payment integration. The scope excludes backend inventory management and supplier interactions.

**1.3 Audience**

The intended audience for this document includes project stakeholders, development team members, quality assurance team, and system architects.

**1.4 Definitions and Acronyms**

* **User**: An individual who visits and interacts with the Online Bookstore.
* **Admin**: An individual with administrative privileges to manage the bookstore.
* **SKU**: Stock Keeping Unit, a unique identifier for each book.

**2. System Overview**

**2.1 System Description**

The Online Bookstore will be a web-based application allowing users to browse and purchase books. The system will include user authentication, book catalog management, shopping cart, order processing, and payment gateway integration.

**2.2 Use Cases**

* User Registration and Login
* Browsing Books by Category
* Adding Books to Shopping Cart
* Placing an Order
* Making a Payment

**3. Functional Requirements**

**3.1 Feature List**

1. User Registration
2. User Login
3. Book Browsing
4. Shopping Cart
5. Order Placement
6. Payment Processing

**3.2 Detailed Requirements**

**3.2.1 User Registration**

* **Description**: Allows new users to create an account.
* **Input**: Username, email, password.
* **Process**: Validate input, store user data in the database.
* **Output**: Confirmation message, new user entry in the database.
* **Dependencies**: Email validation service.
* **Assumptions**: User will provide a valid email address.

**3.2.2 Book Browsing**

* **Description**: Allows users to browse books by category or search by title/author.
* **Input**: Search query, category selection.
* **Process**: Query database for matching books.
* **Output**: List of books matching the search criteria.
* **Dependencies**: Book database.
* **Assumptions**: Book database is regularly updated.

**4. User Interface**

**4.1 Overview**

The user interface will be designed to be intuitive and easy to navigate. The main elements include a homepage, category pages, book detail pages, shopping cart, and checkout process.

**4.2 Screens**

**4.2.1 Homepage**

* **Elements**: Navigation bar, search bar, featured books, categories.
* **Interactions**: Clicking on a book takes the user to the book detail page.

**4.2.2 Book Detail Page**

* **Elements**: Book title, author, description, price, add to cart button.
* **Interactions**: Clicking 'Add to Cart' adds the book to the user's shopping cart.

**5. Data Requirements**

**5.1 Data Elements**

* **User Data**: Username, email, password (hashed), address.
* **Book Data**: Title, author, ISBN, price, description, category, SKU.
* **Order Data**: Order ID, user ID, book SKUs, quantity, total price, order date.

**5.2 Data Sources**

* **User Data**: User registration form.
* **Book Data**: Admin uploads via the admin panel.
* **Order Data**: Generated during order placement.

**5.3 Data Flows**

* User registers -> User data stored in database.
* User browses books -> System queries book database.
* User places order -> Order data stored in database.

**6. Non-Functional Requirements**

**6.1 Performance**

* The system should respond to user actions within 2 seconds.
* The system should handle up to 10,000 concurrent users.

**6.2 Security**

* User passwords must be hashed and stored securely.
* The system must use HTTPS for all communications.

**6.3 Usability**

* The interface should be accessible to users with disabilities.
* The system should provide clear error messages and guidance.

**6.4 Reliability**

* The system should have an uptime of 99.9%.
* The system should recover gracefully from failures.

**6.5 Scalability**

* The system should be able to scale horizontally to handle increased load.

**7. External Interfaces**

**7.1 System Interfaces**

* Integration with an external payment gateway for processing payments.
* Integration with an email service for sending confirmation emails.

**7.2 User Interfaces**

* Web-based user interface accessible from modern web browsers.

**7.3 API Interfaces**

* REST API for accessing book data and user orders.

**8. Constraints and Assumptions**

**8.1 Constraints**

* The project must be completed within six months.
* The budget for the project is $100,000.

**8.2 Assumptions**

* Users have access to the internet and a modern web browser.
* The book database will be provided by the client.

**9. Appendices**

**9.1 Glossary**

* **SKU**: Stock Keeping Unit
* **HTTPS**: HyperText Transfer Protocol Secure

**9.2 References**

* None.

**9.3 Change Log**

* **Version 1.0**: Initial document created on 2024-07-09 by John Doe.

**Functional Specification Document**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the functional requirements and design for creating a Power BI report and deploying it to a designated SharePoint site. The report will provide insights into sales performance, customer demographics, and product trends.

**1.2 Scope**

This document covers the development of the Power BI report, including data extraction, transformation, and loading (ETL) processes, report creation, and deployment. It excludes the creation of new data sources or changes to the underlying database schema.

**1.3 Audience**

The intended audience for this document includes project stakeholders, business analysts, data engineers, report developers, and IT administrators.

**1.4 Definitions and Acronyms**

* **ETL**: Extract, Transform, Load
* **KPI**: Key Performance Indicator
* **DAX**: Data Analysis Expressions
* **PBI**: Power BI
* **URL**: Uniform Resource Locator

**2. System Overview**

**2.1 System Description**

The system will extract sales and customer data from the company’s SQL Server database, transform it as necessary, and load it into Power BI. The Power BI report will be created to visualize key metrics and trends, and then published to a SharePoint site.

**2.2 Use Cases**

* Data Extraction from SQL Server
* Data Transformation and Loading into Power BI
* Report Creation and Visualization
* Report Deployment to SharePoint

**3. Functional Requirements**

**3.1 Feature List**

1. Data Extraction
2. Data Transformation
3. Data Loading
4. Report Creation
5. Report Deployment

**3.2 Detailed Requirements**

**3.2.1 Data Extraction**

* **Description**: Extract sales and customer data from the SQL Server database.
* **Input**: SQL queries.
* **Process**: Execute SQL queries to retrieve data.
* **Output**: Data files (e.g., CSV) or direct data connections.
* **Dependencies**: Access to SQL Server.
* **Assumptions**: Database schema will remain unchanged during the project.

**3.2.2 Data Transformation**

* **Description**: Clean and transform the extracted data for reporting.
* **Input**: Raw data from SQL Server.
* **Process**: Apply data cleaning, aggregation, and transformation rules.
* **Output**: Cleaned and transformed data ready for loading into Power BI.
* **Dependencies**: Data cleaning scripts/tools.
* **Assumptions**: Data quality is consistent and requires minimal cleaning.

**3.2.3 Data Loading**

* **Description**: Load the transformed data into Power BI.
* **Input**: Transformed data.
* **Process**: Import data into Power BI datasets.
* **Output**: Power BI datasets.
* **Dependencies**: Power BI Desktop/Service.
* **Assumptions**: Sufficient Power BI workspace capacity.

**3.2.4 Report Creation**

* **Description**: Create a Power BI report with visualizations and KPIs.
* **Input**: Power BI datasets.
* **Process**: Develop visuals, charts, and dashboards using Power BI Desktop.
* **Output**: Power BI report (.pbix file).
* **Dependencies**: Power BI Desktop.
* **Assumptions**: Report requirements are clearly defined.

**3.2.5 Report Deployment**

* **Description**: Deploy the Power BI report to a SharePoint site.
* **Input**: Power BI report (.pbix file).
* **Process**: Publish the report to Power BI Service and embed it in SharePoint.
* **Output**: Accessible Power BI report on SharePoint.
* **Dependencies**: Power BI Service, SharePoint.
* **Assumptions**: SharePoint site is configured to host Power BI reports.

**4. User Interface**

**4.1 Overview**

The user interface will include the Power BI report embedded in a SharePoint page. The report will feature an intuitive layout with interactive visuals and filters.

**4.2 Screens**

**4.2.1 Power BI Report on SharePoint**

* **Elements**: Interactive charts, graphs, slicers for filtering data, and KPIs.
* **Interactions**: Users can interact with visuals, apply filters, and drill down into data.

**5. Data Requirements**

**5.1 Data Elements**

* **Sales Data**: Transaction ID, date, product ID, quantity, revenue.
* **Customer Data**: Customer ID, name, age, gender, location.
* **Product Data**: Product ID, name, category, price.

**5.2 Data Sources**

* **Sales Data**: SQL Server database.
* **Customer Data**: SQL Server database.
* **Product Data**: SQL Server database.

**5.3 Data Flows**

* Extract sales, customer, and product data from SQL Server.
* Transform data as per business rules.
* Load data into Power BI datasets for report creation.

**6. Non-Functional Requirements**

**6.1 Performance**

* Report should load within 5 seconds.
* Data refresh should complete within 30 minutes.

**6.2 Security**

* Data access should be restricted based on user roles.
* Data should be encrypted in transit and at rest.

**6.3 Usability**

* Report should be accessible on desktop and mobile devices.
* The interface should be user-friendly and intuitive.

**6.4 Reliability**

* Report should be available 99.9% of the time.
* System should handle concurrent access by up to 100 users.

**6.5 Scalability**

* System should accommodate future data growth without performance degradation.

**7. External Interfaces**

**7.1 System Interfaces**

* **Power BI Service**: For publishing and managing the report.
* **SharePoint**: For embedding and accessing the report.

**7.2 User Interfaces**

* **Web-based UI**: Accessible via modern web browsers on desktops and mobile devices.

**7.3 API Interfaces**

* **Power BI REST API**: For automated deployment and management tasks.

**8. Constraints and Assumptions**

**8.1 Constraints**

* The project must be completed within three months.
* The budget for the project is $50,000.

**8.2 Assumptions**

* Users have access to the necessary licenses for Power BI and SharePoint.
* Data sources are reliable and updated regularly.

**9. Appendices**

**9.1 Glossary**

* **ETL**: Extract, Transform, Load
* **KPI**: Key Performance Indicator
* **DAX**: Data Analysis Expressions
* **PBI**: Power BI
* **URL**: Uniform Resource Locator

**9.2 References**

* Power BI Documentation: [Power BI Documentation](https://docs.microsoft.com/en-us/power-bi/)
* SharePoint Documentation: [SharePoint Documentation](https://docs.microsoft.com/en-us/sharepoint/)

**9.3 Change Log**

* **Version 1.0**: Initial document created on 2024-07-09 by Jane Smith.