

SPIRITUAL DEVAL: FINAL REPORT

Eastern Washington University

CSCD 490

Submitted by Andree Ramirez, Kevin Flannery, and Andrea Diaz

March 17th, 2025

Github repository: <https://github.com/Sanmeet-EWU/cscd-488-490-project-spiritual-deval>

Table of Contents

1. Abstract	1
2. Problem Statement and Motivation	2
3. Functional Requirements	3 – 4
4. Non Functional Requirements	5
5. System Design and Architecture	6 – 11
6. Implementation Details	12 – 13
7. Testing and Evaluation	14 – 17
8. Deployment and Maintenance	18 – 19
9. Challenges and Resolutions	20 – 21
10. Future Work and Improvements	22 – 23
11. References	24
12. Appendices	25 – 35

List of Figures

1. Figure 1: System Architecture Diagram	7
2. Figure 2: Class Diagram	8
3. Figure 3: Entity Relationship Diagram	9
4. Figure 4: Login Sequence Diagram	10
5. Figure 5: Event Creation Sequence Diagram	10
6. Figure 6: Use Case Diagram	11

Abstract

Spiritual Deval is a web application designed to help the Dada Dev Deval temple manage events, volunteers, and donations on one platform. Developed using C#, Blazor, and MySQL, it provides an intuitive interface for administrators to organize events, track donations, and oversee volunteer participation. The system includes an event management module, allowing users to browse upcoming events, sign up based on availability and required skills, and receive real time updates. User authentication and role management are handled using Microsoft's Identity Framework, ensuring secure access for both administrators and regular users. The platform is designed to be scalable, supporting different temple needs while keeping operations efficient and organized. By simplifying administrative tasks and enhancing community engagement, Spiritual Deval aims to enhance how the temple interacts with their members and manage resources.

Problem Statement

The temple needs an efficient system to enhance communication with members, manage donations, track event attendance, and coordinate volunteers. Current methods, such as manual record keeping or using multiple disconnected tools, make it difficult to stay organized and engage effectively with the community. Without a centralized solution, tracking member participation, collecting donations securely, and managing volunteer efforts become time consuming and prone to errors.

Motivation

To better serve the community, the temple aims to modernize its management processes through a unified digital platform. A streamlined system would enable better communication, provide a secure way to accept online donations, and make it easier to track attendance and volunteer contributions. By improving efficiency and engagement, the temple can foster stronger connections with its members, encourage more participation in events, and ensure that resources are utilized effectively. By automating routine tasks and providing real time updates, the solution aims to reduce administrative burdens and empower temple staff to focus on building community engagement and ensuring that resources are used effectively.

Functional Requirements

User Management

Functional Requirement	Description / Specification
[FR-1(U1)] User Registration and Authentication	There will be a feature that allows users to register and log in using valid credentials.
Priority	Level 0
[FR-2(U8)] User Profile Management	There will be a feature which allows users to change their password if needed.
Priority	Level 0
[FR-3(U1, U3)] User Event Registration	There will be a feature which allows logged in users to register for events
Priority	Level 0

Admin Management

Functional Requirement	Description / Specification
[FR-4(U4)] Donation Statistic Viewing	There will be a feature which allows administrators to view donation statistics.
Priority	Level 0
[FR-5(U7)] Event Registration Statistics Viewing	There will be a feature which allows administrators to view event registration information.
Priority	Level 0

[FR-6(U6)] Event Creation	There will be a feature that allows administrators to create events that temple members logged in may register to.
Priority	Level 0

Donation Management

Functional Requirement	Description / Specification
[FR-7(U2, U2.1)] Donation Placement	There will be a feature where users can make donations (one-time or recurring).
Priority	Level 0

Volunteer Management

Functional Requirement	Description / Specification
[FR-8(U9)] Volunteer Management	There will be a feature to register for event volunteering opportunities based on availability and skillsets.
Priority	Level 0
[FR-9(U9)] Event Subscription and Updates	There will be a feature that allows volunteers when they register to volunteer to be updated about changes to that event and ask if they want to receive info about more volunteer opportunities.
Priority	Level 0

Non-Functional Requirements

Non-Functional Requirement	Description
[NFR-1] Security	The system shall ensure secure payment processing using encryption.
[NFR-2] Performance	The system should respond quickly to user interactions.
[NFR-3] Scalability	The system should be capable of accommodating an increasing number of users and transactions without compromising performance or functionality.
[NFR-4] Usability	The system will be easy to use for anyone visiting.
[NFR-5] Availability	The system will have an uptime of 99%.

System Design and Architecture

System Architecture

The System Architecture, as seen in Figure 1, follows a three-tier structure:

1. Presentation Layer (Frontend)

- Built with Blazor Server, which enables interactive UI updates using C# and Razor components.
- Handles event browsing, and volunteer registration.

2. Business Logic Layer (Backend)

- Developed using .NET Core, responsible for handling application logic, authentication, and data processing.
- Uses Entity Framework Core to interact with the MySQL database.
- Implements Microsoft Identity Framework for secure user authentication and role based access control.

3. Data Layer (Database)

- Stores data in a MySQL database hosted on AWS RDS.
- Contains tables for Users, Events, Volunteers, and EventRegistrations, and VolunteerRegistrations.
- Supports scalability .

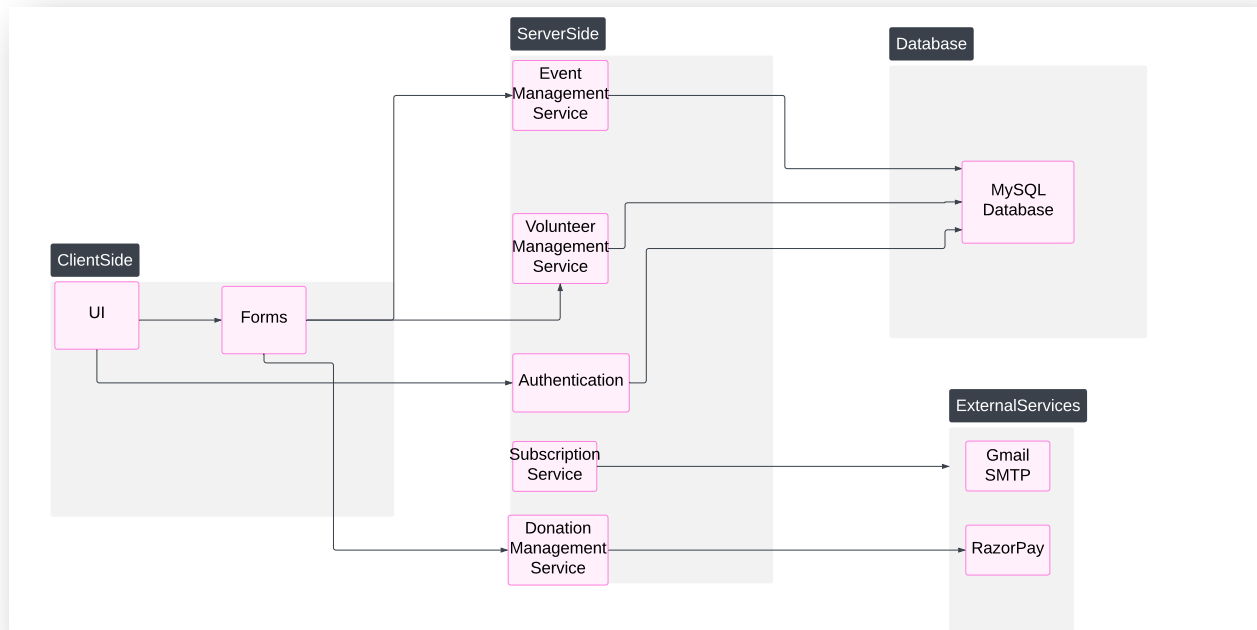


Figure 1: System Architecture Diagram

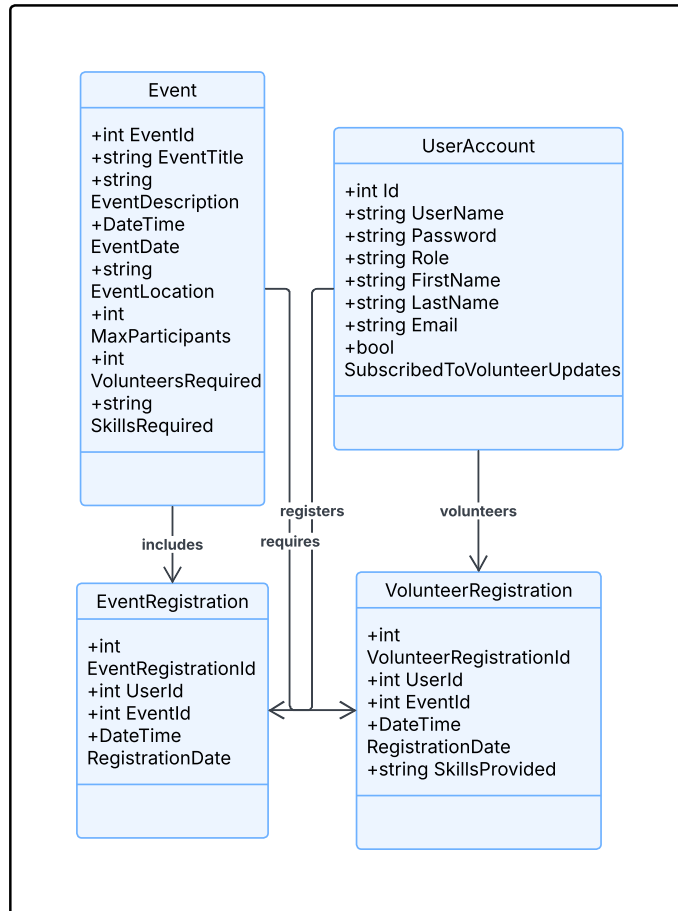


Figure 2: Class Diagram

Entity Relationships and Schema

As seen in Figure 3, the entity relationship diagram shows each entity and how they interact with each other:

- Events and EventRegistrations: A one-to-many relationship where an event can have multiple registrations.
- UserAccounts and EventRegistrations: A one-to-many relationship where a user can register for multiple events.
- Events and VolunteerRegistrations: A one-to-many relationship where an event can have multiple volunteer registrations.
- UserAccounts and VolunteerRegistrations: A one-to-many relationship where a user can volunteer for multiple events.

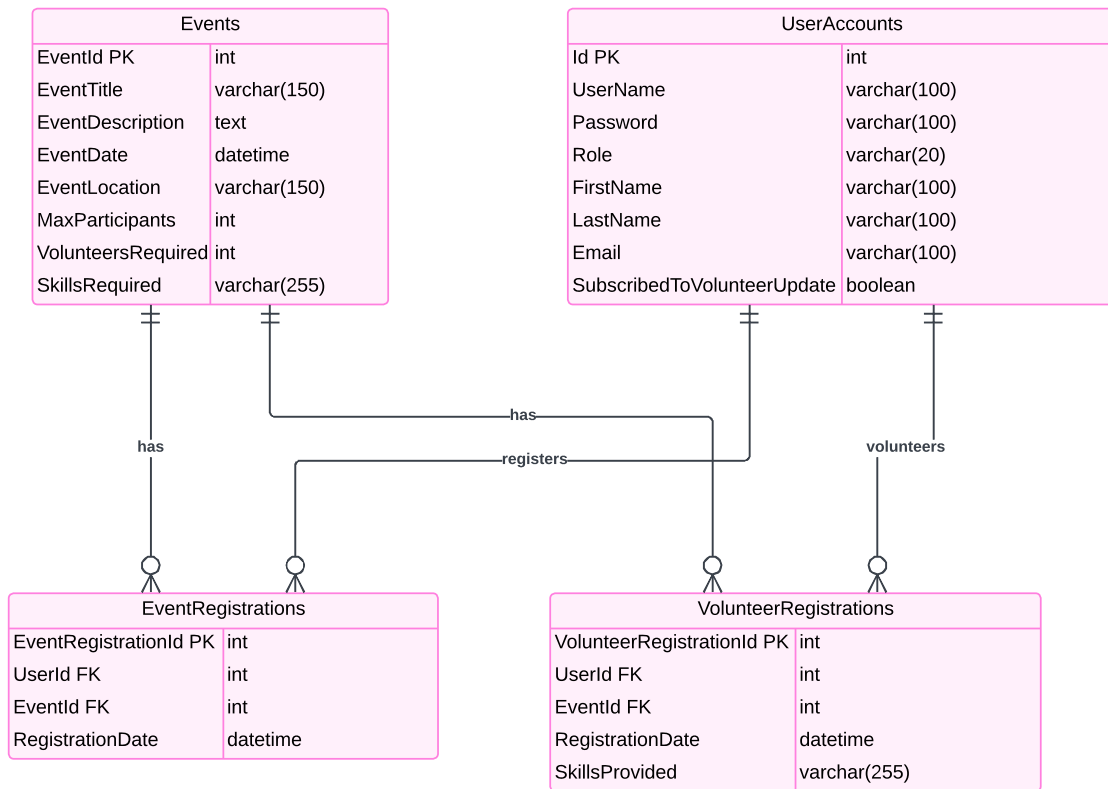


Figure 3: Entity Relationship Diagram

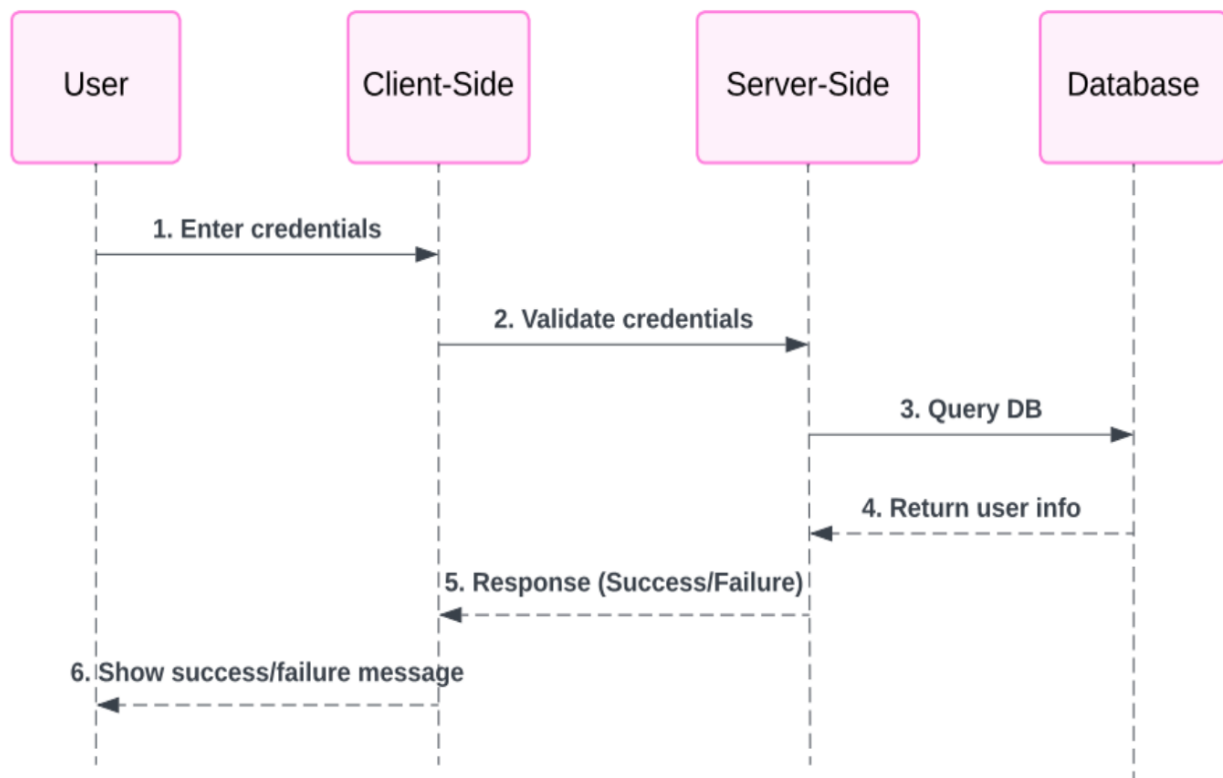


Figure 4: Login Sequence Diagram

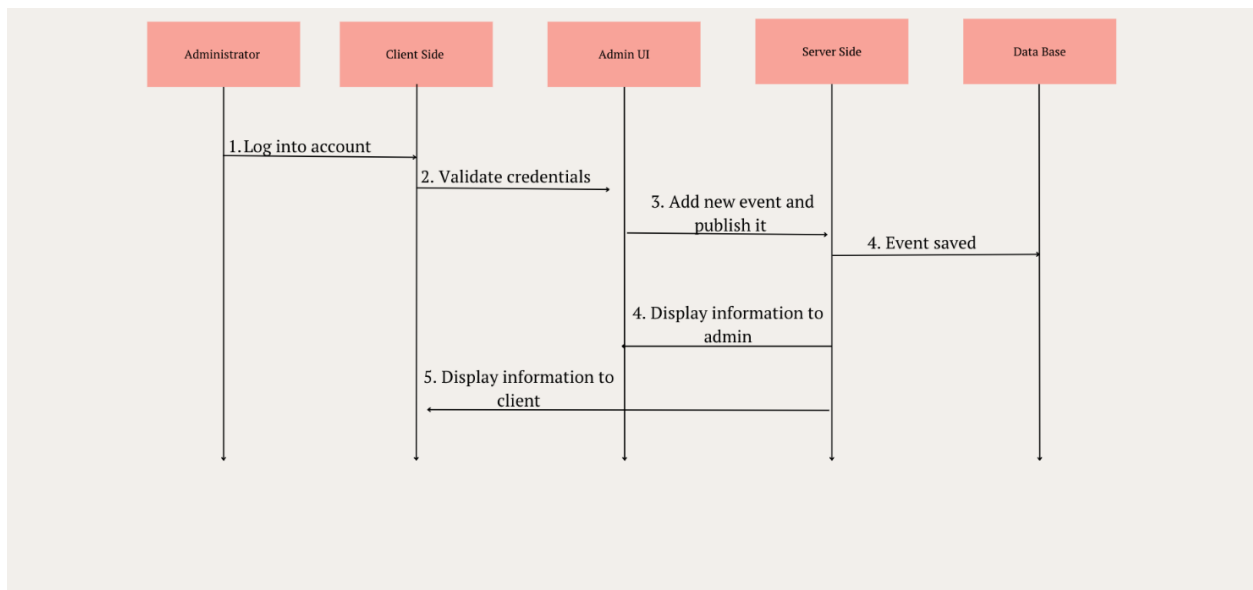


Figure 5: Event Creation Sequence Diagram

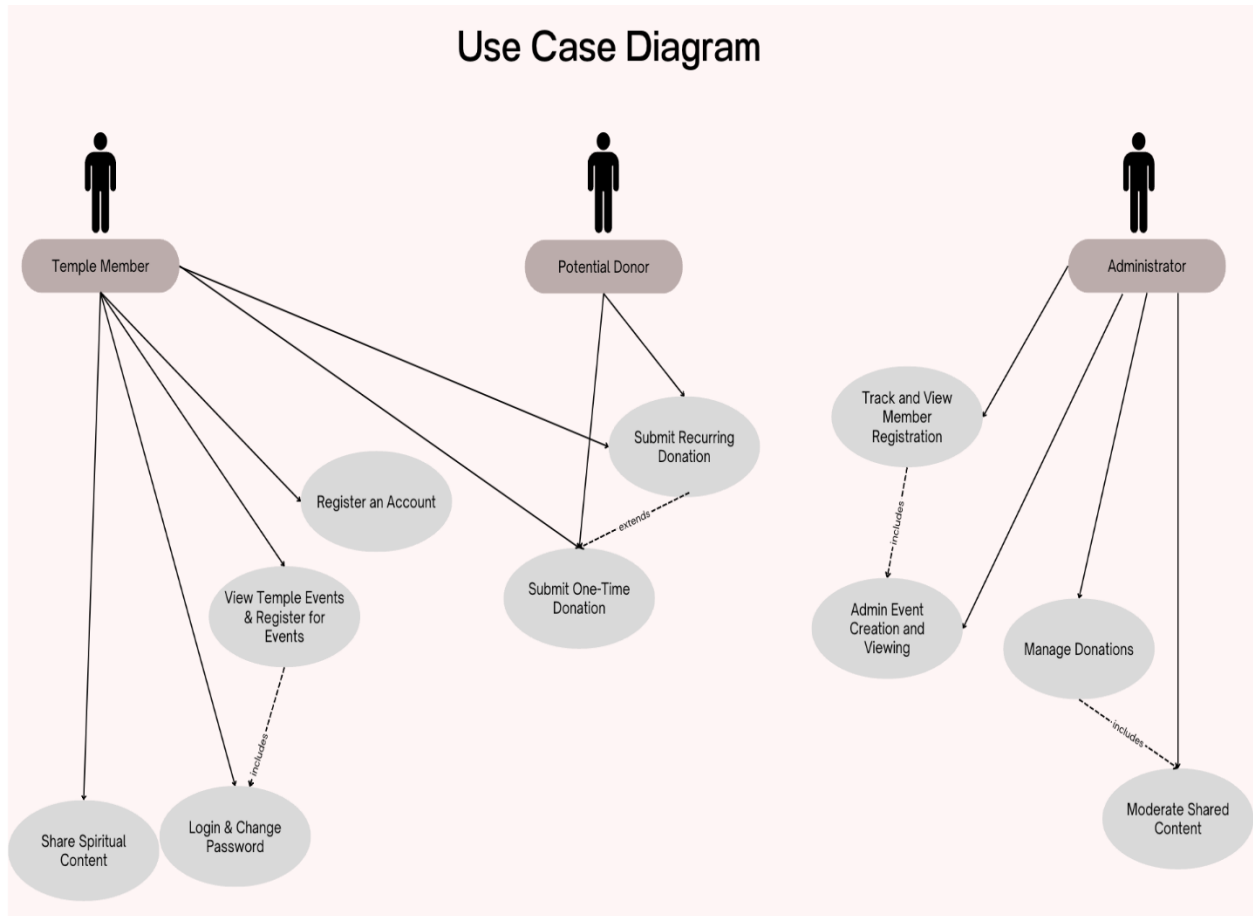


Figure 6: Use Case Diagram (Refer to Appendix A in “Appendices” for all Use Cases)

Implementation Details (Technologies, Frameworks, Data structures)

This project utilizes a modern web development stack to ensure scalability, maintainability, and an interactive user experience. Below are the core technologies used in the implementation:

- **Blazor**: Used for building interactive web UIs with C# and .NET, allowing seamless integration between client-side and server-side logic.
- **.NET**: The backend framework responsible for handling business logic, API endpoints, and data processing.
- **HTML**: Provides the foundational structure and markup for the web application.
- **CSS**: Used for styling, layout design, and responsiveness to ensure a visually appealing and user-friendly interface.
- **MySQL**: The database management system that stores user data, event details, donation records, and volunteer registrations.
- **AWS**: Cloud hosting and deployment services to ensure scalability, security, and availability of the application.
- **Visual Studio Code (VS Code)**: The primary development environment for coding, debugging, and project management.
- **Entity Framework Core**: Used for Object Relational Mapping, enabling seamless database interactions.

- **Primary Data Models:**

- **UserAccount:** Stores user information and authentication details.
- **Event:** Represents events with details like title, date, location, and required volunteers.
- **EventRegistration:** Links users to events they sign up for.
- **VolunteerRegistration:** Tracks volunteers and their associated skills.
- **Gmail SMTP:** Used for sending volunteer updates.

These technologies work together to deliver a fully formed and efficient web application tailored to the needs of the spiritual temple.

Testing and Evaluation (Testing Strategies and Results)

Test Plan:

Our testing methodology has primarily been manual thus far. We have focused on verifying the basic functionality of core features by directly interacting with the application and observing outputs. We have tested the Create Account and Login forms, among various other Blazor components, by submitting user input and verifying that the record was either submitted successfully in MySQL Workbench, printed to the terminal, or showed the expected behavior of the component.

Test Cases and Results:

Below are some key test cases we have executed:

Test Case	Description	Expected Outcome	Actual Outcome	Status
Submit Create Account Form	Enter user details (name, email, password, address) and submit the form	Input data is captured and printed to the terminal	Input data was successfully captured and printed	Passed
Submit Login Form	Enter email and password and submit the form	Input data is captured and printed to the terminal	Input data was successfully captured and printed	Passed

Event Creation	Submit 'Create Event' form	New event is visible in database and shows up on the home page.	New event was visible in database and showed up on the home page.	Passed
Normal Registration	Click 'Register' button on event page	New EventRegistration record gets created in Database	New EventRegistration record was created	Passed
Volunteer Registration	Click on 'Register as Volunteer' on volunteer event page, filling out optional 'Skills Provided' section	New VolunteerRegistration record with skills provided gets created in Database	New VolunteerRegistration record with skills provided created in Database.	Passed
Volunteer Subscribe Button/Send	Click 'Subscribe for Updates' on volunteer event page. Admin	An email should be received.	An email was received.	Passed

Volunteer Update	uses 'Send Volunteer Update' to send update email			
Get in Touch Button (Logged in)	Click 'Get in Touch' button from Volunteer page	Send user to Home page	Sent user to Create Account page	Failed
Get in Touch Button (Not logged in)	Click 'Get in Touch' button from Volunteer page	Send user to Create Account page	Sent user to Create Account page	Passed
Promote User to Admin	Enter user's email that you would like to promote and click 'Promote'	User's role in database should be 'Administrator'	User's role in database was 'Administrator'	Passed
Try Register Volunteer (When max volunteers are already met)	Go to event volunteer page where max number of volunteers are met	'Volunteer limit reached for this event.' should be displayed	'Volunteer limit reached for this event.' is displayed	Passed

Try Register to Event (When max registrations are already met)	Go to event register page where max number of registrations are met	'This event is full.' should be displayed.	'This event is full.' is displayed.	Passed
--	---	--	-------------------------------------	--------

Deployment and Maintenance (Steps For Deployment and Maintenance)

The deployment process for Spiritual Deval involves setting up the necessary infrastructure, configuring the database, and deploying the Blazor Server application on AWS. Below are the key steps, with what happens at each step (More robust instructions are provided in the Deployment Instructions):

Deployment Steps

1. Fork and Clone the Repository – The project is forked from GitHub, and the main branch is cloned.
2. Set Up AWS RDS (MySQL) – A MySQL database is created on AWS RDS with necessary configurations for connection.
3. Configure Email Notifications – A Gmail account is set up with app passwords for sending volunteer updates.
4. Deploy to AWS EC2 – An EC2 instance is created and configured with security rules. The Blazor application is published and transferred to the server.
5. Run the Application – The app is started on the EC2 instance and made accessible via a public IP.
6. Automate Deployment – A system service is set up to ensure the application restarts automatically after reboots.

Maintenance Steps

1. Updating the Deployment – New updates are published and transferred to the EC2 instance, followed by restarting the service.
2. Database Backups & Recovery – AWS RDS snapshots are used to back up and restore the database if needed.
3. Performance Monitoring – AWS monitoring tools track CPU usage, network traffic, and database queries for optimization.
4. Security Checks – Regular audits ensure EC2 security group rules and database access settings remain secure.
5. Domain Management – The EC2 public IP is linked to a domain via DNS configuration for accessibility.

This structured approach ensures that Spiritual Deval remains stable, scalable, and secure in a production environment.

Challenges and Resolutions

Throughout the development of Spiritual Deval, we encountered several challenges that required problem solving and adaptability. Below are the key issues we faced and how we resolved them.

Email Service Integration

Integrating the email service with Gmail SMTP proved challenging due to outdated tutorials and documentation. Many available resources referenced deprecated methods, making implementation difficult. We resolved this by referring to the latest official Google documentation and configuring the app password correctly for secure authentication.

Database Configuration on Microsoft Azure

Initially, we attempted to configure the database on Microsoft Azure, but we faced persistent issues with setup and connectivity. After troubleshooting, we opted to use AWS RDS, which provided a more seamless and reliable database hosting solution.

Deployment Challenges

Deploying the project on AWS Elastic Beanstalk led to several unexpected errors that were difficult to debug. After evaluating alternatives, we successfully deployed the application using AWS EC2, which offered greater flexibility and control over the deployment process.

Payment Gateway Selection and Implementation

Choosing a suitable payment gateway was another hurdle. After considering multiple options, we decided on Razorpay. However, the account verification process took longer than expected, delaying implementation. Additionally, since an API key was required to test the system, we were unable to fully validate the payment functionality in advance. In hindsight, we should have prioritized payment gateway integration earlier in the project timeline to avoid last minute setbacks.

Staggered Communication with the Sponsor

At times, delayed responses from our project sponsor left us uncertain about the next steps. This lack of timely communication made it challenging to proceed efficiently. To mitigate this, we adjusted our workflow by making informed decisions based on the requirements we had, while ensuring we followed up regularly to keep the project aligned with its expectations.

By navigating these challenges, we gained valuable experience in troubleshooting, adapting to new technologies, and improving project management strategies for better efficiency in future development efforts.

Future Work and Improvements

While Spiritual Deval provides core functionality for event management, volunteering, and administrative features, there are several areas for future enhancements and bug fixes to improve usability and expand features.

Bug Fixes

- **Scrolling Issues:** The Home.razor and Volunteer.razor pages currently have two scroll bars, affecting user experience. This needs to be resolved for a smoother interface.
- **Volunteer Page Navigation:** The “Get in Touch” button on Volunteer.razor incorrectly redirects users to CreateAccount.razor, even when they are already logged in. Attempting to log in again results in an error. This should be fixed to properly direct logged in users to the homepage.

Planned Enhancements

- **Complete Payment Portal:** The integration of Razorpay needs to be finalized to allow secure donations.
- **Admin Payment Dashboard:** Enable admin users to view payment statistics and track donation trends.
- **Event Image Uploads:** Implement functionality to upload images when creating events to make event listings more visually engaging.
- **Event Sharing Feature:** Add a “Share” button on event pages to allow users to easily share events on social media or via direct links.

- Event Comments Section: Introduce a comment system under events to encourage discussions and engagement.
- Home Page Slideshow: Implement a dynamic slideshow on the homepage to showcase the temple's culture, history, and ongoing events.

By addressing these bugs and enhancements, Spiritual Deval can offer a more seamless, visually appealing, and feature-rich experience for users, volunteers, and administrators.

References

1. dotnet. (2024, January 25). *What is Blazor? [Pt 1] | Front-end Web Development with .NET for Beginners*. [YouTube video]. Retrieved from <https://www.youtube.com/watch?v=sIIfnGqElds&list=PLdo4fOcmZ0oXNZX1Q8rB-5xgTSKR8qA5k&index=2>
2. dotnet. (2024, January 25). *Build and run your first Blazor web app [Pt 2] | Front-end Web Development with .NET for Beginners*. [YouTube video]. Retrieved from <https://www.youtube.com/watch?v=llDc88XE-Q&list=PLdo4fOcmZ0oXNZX1Q8rB-5xgTSKR8qA5k&index=3>
3. dotnet. (2024, January 25). *Introducing Blazor Components [Pt 3] | Front-end Web Development with .NET for Beginners*. [YouTube video]. Retrieved from <https://www.youtube.com/watch?v=sbXzifvMKXE&list=PLdo4fOcmZ0oXNZX1Q8rB-5xgTSKR8qA5k&index=4>
4. Microsoft. (n.d.). *Blazor Documentation*. Retrieved December 5, 2024, from <https://learn.microsoft.com/en-us/aspnet/core/blazor/>

Appendices

Appendix A: Use Cases

Use Case 1: Make a Donation

- Actors: Temple Donor
- Preconditions: User is on the temple website
- Postconditions: Donation is processed, and a confirmation email is sent to the customer.
- Main Flow:
 - * User navigates to the Donation Page.
 - * Selects kind of donation (One-time or Recurring)
 - * Enters payment and details.
 - * Confirms the donation via email.
- Alternative Flow: If payment fails, the user retries payment with a different method.
- Related Requirements: FR-7, FR-8

Use Case 2: Register for Account

- Actors: Temple Member
- Preconditions: User is on the temple website
- Postconditions: User has valid login credentials.
- Main Flow:
 - * User navigates to the Log in page and clicks ‘Create Account’
 - * Enters name, email, and password and clicks ‘Create Account’
 - * Confirms account creation via email.
- Alternative Flow: If account creation fails, the user retries with a different email.
- Related Requirements: FR-1

Use Case 3: Register for Event

- Actors: Temple Member
- Preconditions: User is logged in on the temple website
- Postconditions: User is registered for event and a confirmation email is sent to the

customer.

– Main Flow:

- * User navigates to the Home Page.
- * Selects 'Register for Event' button
- * The button changes color and displays 'Registered!'

– Alternative Flow: If user isn't logged in, they are redirected to the login screen.

– Related Requirements: FR-1, FR-3

Use Case 4: Manage Event Registrations

– Actors: Administrator

– Preconditions: User is logged into the administrator account on the temple website.

– Postconditions: Administrator can view, approve, or cancel event registrations.

– Main Flow:

- * Administrator clicks on the 'View Registrations' tab.
- * Views a list of all upcoming events with their registration counts
- * Selects an event to view detailed registration information.
- * Sees a list of links to all member profiles that have registered for that event.

– Alternative Flow: If the administrator cannot access the registrations, an error message is displayed.

– Related Requirements: FR-3, FR-5, FR-6

Use Case 5: View Donation Information

– Actors: Administrator

– Preconditions: User is logged into the administrator account on the temple website.

– Postconditions: Admin can view donation statistics

– Main Flow:

- * User navigates to the 'Donation Information' tab.
- * Sees list of one-time donations accompanied by user profiles and list of active recurring donors.
- * There are different ways of viewing the donation data (inquire with sponsor about specifics)

- Alternative Flow: If user isn't logged in as an admin, they can't find this page.
- Related Requirements: FR-7, FR-8, FR-4

Use Case 6: Create Event

- Actors: Administrator
- Preconditions: User is logged into the administrator account on the temple website.
- Postconditions: Event is created and displayed on the home page/calendar page of the site
- Main Flow:
 - * The administrator navigates to the “Event Management” section of the platform.
 - * The administrator selects the option to “Add New Event.”
 - * The administrator fills in the required fields, including the event title, description, date, and optional image.
 - * The administrator clicks the “Publish” button.
 - * The system confirms the event creation and displays the event on the calendar and home page.
- Alternative Flow: If the administrator fails to provide required details (e.g., missing title or description), an error message prompts the administrator to fill in the missing information.
- Related Requirements: FR-6

Use Case 7: Change Password

- Actors: User
- Preconditions: User has forgotten their password.
- Postconditions: User has a new password and can log in.
- Main Flow:
 - * The user clicks ‘Forgot Password’
 - * The user enters their email address, and an email is sent to them
 - * The user clicks the reset password link in the email and enters their new password twice, ensuring that they match
 - * The user clicks the “Save New Password” button and can login with their new password.
- Alternative Flow: If the passwords don't match, a message displays in red stating so until they do match
- Related Requirements: FR-2

Use Case 8: Register for Volunteer Event

- Actors: Volunteer (User)
- Preconditions: Volunteer is logged in and can access a list of upcoming events.
- Postconditions: Volunteer is registered for the selected event and has the option to subscribe to event updates.
- Main Flow:
 - *The volunteer navigates to the list of upcoming events.
 - *The system displays the list of events with full details.
 - *The volunteer selects an event and clicks the “Volunteer” button.
 - *The system registers the volunteer for the event.
 - *The system sends a confirmation of successful registration.
 - *The system asks if the volunteer wants to receive updates about the event and future opportunities.
- Alternative Flow: If the event is already full or the volunteer doesn’t meet the required skills, the system notifies the volunteer and provides alternative events.
- Related Requirements: FR-8

Appendix B. User stories and scenarios

User Story 1: Member Registration

U1: As a temple member, I want to register for an account so that I can access temple updates and make donations.

Feature: Member Registration

Scenario: Successful registration

- Given I am on the registration page
- When I enter my name, email, and a valid password
- And I click the "Register" button
- Then I should receive a confirmation email
- And I should be redirected to the login page

Scenario: Unsuccessful registration due to missing information

- Given I am on the registration page
- When I leave the password field blank
- And I click the "Register" button
- Then I should see an error message "Password is required"
- And I should remain on the registration page

User Story 2: Donation Submission

U2: As a potential donor, I want to submit a one-time donation so that I can contribute to the temple's cause.

Feature: Donation Submission

Scenario: Successful one-time donation

- Given I am on the donation page
- When I select "One-time donation" and enter my payment details
- And I click the "Donate" button
- Then I should see a confirmation message "Thank you for your donation!"
- And I should receive a receipt in my email

Scenario: Unsuccessful donation due to invalid payment details

- Given I am on the donation page
- When I enter an invalid credit card number
- And I click the "Donate" button
- Then I should see an error message "Invalid payment information"
- And the payment should not be processed

User Story 2.1: Recurring Donation Submission

U2.1: As a potential donor, I want to submit a recurring donation so that I can contribute to the temple's cause periodically.

Feature: Recurring Donation Submission

Scenario: Successful recurring donation

- Given I am on the donation page
- When I select "Recurring donation", enter my payment details, and select how frequently I would like to donate (Monthly, Quarterly, Yearly)
- And I click the "Set Recurring Donation" button
- Then I should see a confirmation message "Thank you for your donation!"
- And I should receive a receipt in my email

Scenario: Unsuccessful recurring donation due to invalid payment details

- Given I am on the donation page

- When I select "Recurring donation", enter invalid payment details, and select how frequently I would like to donate
- And I click the "Set Recurring Donation" button
- Then I should see an error message "Invalid payment information"
- And the payment should not be processed

Scenario: Unsuccessful recurring donation due to not specifying donation frequency

- Given I am on the donation page
- When I select “Recurring donation”, enter valid CC information, but do not select the donation frequency
- And I click the “Set Recurring Donation” button
- Then I should see an error message “Please Specify Donation Frequency”
- And the payment should not be processed

User Story 3: Event Participation

U3: As a temple member, I want to register for upcoming events so that I can participate in temple activities.

Feature: Event Registration

Scenario: Successful event registration

- Given I am logged in as a temple member
- When I view the event calendar or home page
- And I select an event and click "Register"
- Then I should see a message "You have successfully registered for this event"
- And the event should appear in my profile’s upcoming events list

Scenario: Event registration failure due to no login

- Given I am not logged in
- When I attempt to register for an event

- Then I should be prompted to log in
- And I should be redirected to the login page

User Story 4: Administrator Donation Management

U4: As an administrator, I want to track all user donations so that I can manage temple contributions efficiently.

Feature: Donation Management

Scenario: View donations report

- Given I am logged in as an administrator
- When I access the "Donations" tab
- Then I should see a list of all user donations
- And I should be able to filter donations by date, amount, and donor name

Scenario: Failure to access donation records without admin privileges

- Given I am logged in as a regular temple member
- When I try to access the "Donations" tab
- Then I should receive an error message "Access restricted to administrators"

User Story 5: Content Sharing

U5: As a temple member, I want to share spiritual content so that I can contribute to the temple's online community.

Feature: Content Sharing

Scenario: Successful content sharing

- Given I am logged in as a temple member
- When I navigate to the "Share Content" page
- And I upload an image or write a post

- Then I should see a confirmation message "Your content has been shared"
- And the content should appear in the community feed

Scenario: Failed content sharing due to file size limit

- Given I am logged in as a temple member
- When I try to upload a file larger than the allowed size
- Then I should see an error message "File size exceeds the limit"
- And the content should not be uploaded

User Story 6: Admin Event Creation

U6: As a temple admin, I want to create a temple event to be displayed

Feature: Event Creation

Scenario: Successful Event Creation

- Given I am signed in as an Admin on the Home Page
- When I click "Add New Event"
- When I fill in the Title, Description, and date of the event (with the option of adding a photo) and click "Publish"
- That event should be displayed on the home page/calendar page of the site

Scenario: Unsuccessful Event Creation due to missing information

- Given I am signed in as an Admin on the Home Page
- When I click "Add New Event"
- When I fill in the Title and leave the Description of the event blank (or vice versa) and click "Publish"
- I will get an error message prompting me to fill in whichever field I left blank (Title or Description or both)

User Story 7: Admin Event Registration Viewing

U7: As a temple admin, I want to view all registrations to events in one place

Feature: Event Registration Viewing

Scenario: Successful Event Registration Viewing

- Given I am signed in as an Admin
- When I click on the 'View Registrations' tab
- When I click on the Event that I want to view
- I should have a list of links to all member profiles that have registered for that event

User Story 8: User Change Password

U8: As a user, I want to be able to change my password if I forget it.

Feature: Change Password

Scenario: Successful Password Change

- Given I have forgotten my password
- When I click on the 'Forgot Password?' link and enter the email associated with my account
- I will receive an email with a link to reset my password
- When I click the link, I will be prompted to enter a new password in two separate fields
- If the passwords match, I will be able to login with my new password
- If the passwords don't match, I'll be prompted to make them match

User Story 9: Volunteer Event Registration

U9: As a volunteer, I want to be able to register for events based on my availability and skillset.

Feature: Volunteer Event Registration

Scenario: Successful Volunteer Event Registration

- Given I am logged in as a user
- When I view a list of upcoming events
- And I can select an event to volunteer for based on my availability and skillset
- When I click the ‘volunteer’ button, then I will be registered for that event
- I should receive a confirmation of my registration for the event, and be asked if I want to receive updates on that event and future volunteer opportunities.
-

Scenario: Event Full – Cannot Register

- Given I am logged in as a user
- When I view a list of upcoming events
- And I can select an event to volunteer for based on my availability and skillset
- When I click the ‘volunteer’ button, then I will be notified that the event has reached its maximum number of volunteers
- And I will be unable to register for that event
- I should be presented with alternative events that still have available volunteer spots and be asked if I want to receive updates on that event and future opportunities.

