**Design and Implementation Procedure**

The Event Management System was designed using a modular, layered architecture to separate key concerns. The Presentation Layer handles the user interface with Django-rendered HTML templates, styled using CSS, and enhanced with JavaScript for interactivity. The Business Logic Layer uses Django views and models to manage core functionalities, such as authentication, event creation, and email reminders. The Data Layer employs SQLite to store user and event data. For asynchronous tasks like email notifications, Redis and Celery are integrated for efficiency and scalability.

The system was implemented in phases using the Agile Scrum methodology to ensure flexibility and iterative improvement. Each sprint focused on a specific set of features, including user authentication, event management, and notification reminders. Testing and feedback were incorporated at the end of each sprint to refine the requirements and improve the design.

The choice of tools and libraries was guided by project requirements and scalability needs. Django was chosen for its simplicity and secure development capabilities, while Redis was selected for its speed in handling task queues. Celery was used for task scheduling, and SMTP ensured reliable email delivery. Each tool was thoroughly researched, and its suitability for the project was documented for future reference.

Design decisions were driven by ease of use, performance, and scalability. A layered architecture ensures modularity and maintainability. Redis and Celery were integrated to ensure smooth handling of asynchronous tasks. These choices were made after evaluating alternative solutions and prioritizing project scope and scalability needs.