CHAPTER -3

DESIGN AND METHODOLGY

3.1 Introduction

The design and methodology chapter outlines the approach taken to develop "ಚರ್ಚೆ," an online discussion forum. This chapter covers the problem analysis, selection of appropriate technologies, architectural design, and the step-by-step methodology followed during the project development.

3.2 Problem Analysis

The problem analysis involves identifying the key challenges faced by existing discussion forums, as highlighted in the literature survey. These challenges include:

- User Engagement: Difficulty in maintaining active participation and interaction among users.
- Scalability: Handling increased traffic and data volume as the user base grows.
- Usability: Simplifying user interface and navigation to encourage user adoption.
- Security: Implementing robust security measures to protect user data and ensure privacy.
- By addressing these challenges, "ಚರ್ಚೆ" aims to provide a comprehensive solution that enhances user experience, ensures scalability, and maintains high security standards.

3.3 System Architecture

The architecture of "ಚರ್ಚೆ" is designed to be modular, scalable, and secure. The key components of the system include:

- Frontend: Built using HTML, CSS, and JavaScript, the frontend provides an intuitive and user-friendly interface for users to interact with the forum.
- Backend: Developed using Django, the backend handles user authentication, data management, and server-side logic.
- Database: A relational database is used to store user information, discussion threads, and other relevant data.

3.4 Design Principles

The design of "ಚರ್ಚೆ" follows several key principles:

- User-Centric Design: Prioritizing user experience by making the interface intuitive and easy to navigate.
- Modularity: Developing the system in a modular fashion to facilitate easy maintenance and future enhancements.
- Scalability: Ensuring the system can handle a growing number of users and increased data volume efficiently.
- Security: Implementing robust security measures to safeguard user data and ensure privacy.

3.5 Methodology

The development methodology for "ಚರ್ಚೆ" follows an Agile approach, which involves iterative and incremental development. The key stages of the methodology are:

- Requirements Gathering: Collecting and documenting the requirements for the forum, including features, functionalities, and user expectations.
- Planning: Creating a detailed project plan, including timelines, milestones, and resource allocation.
- Design: Developing the system architecture and design specifications, including database schemas, user interfaces, and API designs.
- Development: Implementing the frontend, backend, and database components, followed by integration and testing.
- Testing: Conducting comprehensive testing to identify and fix bugs, ensure functionality, and validate performance.
- Deployment: Deploying the forum to a production environment, ensuring all components are properly configured and operational.
- Maintenance: Providing ongoing support, addressing issues, and implementing enhancements based on user feedback.

3.6 Tools and Technologies

The following tools and technologies are used in the development of "ಚರ್ಚೆ":

Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linex and macOS Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. Visual Studio Code also ships with IntelliSense for TypeScript JSON, CSS, HTML and PHP.

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −It is released under an open-source license. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. and works very quickly and works well even with large data sets.
By encapsulating UI elements into self-contained components, React.js promotes code reusability, modularity, and scalability. Additionally, React.js leverages a virtual DOM to optimize rendering performance, minimizing unnecessary DOM manipulations and ensuring fast updates to the UI. With its unidirectional data flow and robust ecosystem of libraries and tools, React.js has become a go-to-choice for building modern web applications that deliver seamless user experiences across various devices and platforms.

HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and Java Scripts, it forms a triad of cornerstone technologies for the World Wide Web. Web browser receives HTML documents from a web server or from local storage and renders the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a element. This cascading priority scheme is predictable.

DJANGO

Django is an ideal choice for developing "ಚರ್ಚೆ - An Online Discussion Forum" due to its robust feature set and ease of use. Here's why:

- Rapid Development: Django's simplicity and built-in features allow for quick development cycles, helping in getting the forum up and running efficiently.
- Scalability: The framework's design can handle the growth of user base and data, ensuring the forum remains performant as it scales.
- Security: With security features built into the framework, user data and interactions within "ಚರ್ಚೆ" will be well-protected.
- Admin Interface: The auto-generated admin interface will facilitate easy management of user accounts, discussions, and moderation tasks.
- Community Support: Django's active community provides a wealth of resources, plugins, and support, which can be invaluable during development and maintenance

BOOTSTRAP

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web

PYTHON

Python, known for its simplicity and readability, is the backbone of Django, a high-level web framework. Python's extensive standard library, dynamic typing, and strong community support enhance Django's capabilities, enabling rapid development, scalability, and robust security for web applications. This synergy makes Python ideal for efficient Django-based projects

3.7 Summary

The design and methodology chapter provides a detailed overview of the approach taken to develop "ಚರ್ಚೆ." By following a user-centric design, modular architecture, and Agile development methodology, the project aims to create a scalable, secure, and user-friendly online discussion forum. The next chapter will delve into the development and implementation details, showcasing how the design and methodology were translated into a functional system