**ABSTRACT**

The URL Shortener project is focused on developing a web-based application that offers users the capability to shorten long URLs and generate customized short URLs for easy sharing. This involves the thorough analysis, design, and implementation of various software modules and components to provide the desired functionality. The system architecture is designed to efficiently handle URL shortening requests, manage URL mappings, and ensure effective user management. The implementation of the project involves utilizing a carefully selected technology stack that encompasses programming languages, frameworks, and databases. This stack is chosen to support the system's functionalities effectively and provide a robust and scalable solution. Throughout the development process, rigorous testing methodologies are employed to ensure the accuracy, reliability, and security of the system. In conclusion, the URL Shortener project has successfully achieved its objectives by providing users with a convenient and efficient way to shorten URLs and generate personalized short URLs. However, it is important to acknowledge the limitations of the system. These limitations could include constraints related to scalability, performance, or certain functional aspects. To address these limitations and further enhance the system, future enhancements are proposed, such as incorporating additional features, improving user experience, and optimizing system performance. These future enhancements will contribute to the continuous development and improvement of the URL Shortener application, ensuring its effectiveness in meeting user requirements and expectations.

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure name** | **Page No** |
| **Fig 3.5** | Use case Diagram (i) | 14 |
| **Fig 3.5** | Use case Diagram (ii) | 14 |
| **Fig 3.6** | System Architecture Block Diagram | 15 |
| **Fig 3.6** | System Architecture | 16 |
| **Fig 4.1** | Data Flow Context Diagram | 17 |
| **Fig 4.1** | Data Flow Flowchart | 18 |
| **Fig 4.2** | Database Design Diagram | 19 |
| **Fig 4.3** | User Interface Design Diagram | 20 |
| **Fig 5.1** | Structure of URL | 21 |
| **Fig 5.1** | Flow chart URL shortning | 21 |
| **Fig 5.2** | Custom URL generation Flowchart | 22 |
| **Fig 5.3** | Expiration Management Flowchart | 23 |
| **Fig 5.4** | User Management Flowchart | 24 |
| **Fig 6.6.1** | Output of Flask code | 40 |
| **Fig 6.6.2** | Output of login1.html | 40 |
| **Fig 6.6.3** | Output of login2.html | 41 |
| **Fig 6.6.4** | Output of base.html | 41 |
| **Fig 6.6.5** | Output of Index.html | 42 |
| **Fig 6.6.6** | Output of the Shortened URL | 42 |

**LIST OF ABBREVIATIONS**

1. URL : Uniform Resource Locator

2. API : Application Programming Interface

3. UI : User Interface

4. ER : Entity-Relationship

5. HTML : Hypertext Markup Language

6. CSS : Cascading Style Sheets

7. DBMS : Database Management System

8. HTTP : Hypertext Transfer Protocol

9. DNS : Domain Name System/ Domain Name Service

10. CRUD : Create, Read, Update, Delete

11. SQL : Structured Query Language