

SOFTWARE DESIGN SPECIFICATION

BIG DATA ANALYTICS LAB

Prepared by: **TECH INVADERS**

Group members

1. HARSH SHARMA(IIT2021131)
2. SUMIT PRAKASH (IIT2021152)
3. ARYAN PANDEY (IIT2021153)
4. SARANSH YADAV(IIT2021162)
5. SARANSH GUPTA(IIT2021163)
6. KUSHAGRA (IIT2021191)

1. Introduction:

1.1 Purpose

The purpose of this document is to provide a software design specification for the development of a Big Data Analytics Lab software.

1.2 Scope

This software aims to facilitate the management of ongoing projects and future collaboration conducted under the Big Data Analytics Lab of Indian Institute of Information Technology Allahabad.

1.3 Intended Audience

This document is intended for software developers involved in the development, implementation, maintenance and management of the Big Data Analytics Lab website.

2. System Overview:

2.1 System Architecture

The software will follow a web application architecture, where the database interacts with applications, middleware systems to ensure multiple applications can work together while the user interface provides a user-friendly interface for interacting with the system.

2.2 Key Features

- Admin:

We provide special authorisation to admin. The admin is able to add or delete projects, publications or courses.

- User/Author:

User will be able to see ongoing projects, courses and past publications made by the authors associated with BDA lab. User can search for all the projects and publications done by a particular author. Author will be able to see all of his/her publications at their login dashboards.

- For both:

Signing in is not necessary to view the website. Only admin can make any modification to data. Signing in is required only to gain special access.

3. System Components:

3.1 User Interface

- No need to learn complex commands/languages for working with UI.
- Easiness for non-technical people. A beginner can navigate through a site with ease if its simple and well informative.

- Easy setup and ready to start working. Hiding the complexity of actions from the user and displaying only the required information is the key to good interface.

3.2 Data Storage

3.5 Security and Access Control

- Implement authentication and authorisation mechanisms to control the access to the system.
- Encrypt sensitive data .

4. Maintenance and Support

- Provide regular updates and bug fixes to address issues and improve system stability.
- Offer technical support and documentation to assist users in resolving any software-related problems.
- Monitor system performance and provide proactive monitoring and alerting mechanisms.

5. User Interface Design:

- The user interface is designed using HTML and CSS to provide an intuitive and visually appealing experience.
- CSS stylesheets is used for consistent styling across all pages.
- EJS templates are used to dynamically render data on the user interface.

6. Error Handling:

- Appropriate error handling mechanisms are implemented using JavaScript and Node.js to catch and display error messages to users.
- Server-side validation is performed to validate user inputs and handle any potential errors.

7. Performance Considerations:

- Caching mechanisms are implemented on the server-side to optimise performance.

8. Security Considerations:

- User authentication and authorisation will be implemented using secure techniques such as crypt for password hashing and session management.

9. Deployment Considerations:

- The application will be deployed on a web server capable of running Node.js applications.
- The server environment will be set up to support the required technologies (HTML, CSS, Node.js, Mongo DB).
- Proper configurations will be made to ensure secure and reliable communication between the server and clients.