Cloud Counselage

GIFT A CAREER FOUNDATION

SOFTWARE DESIGN SPECIFICATION

Web Development

FindAcademia

Created By:	Mshireen A	**Approved By:**	
Created On:	07-09-2024	**Approved On:**	

Page left blank intentionally


Cloud Counselage

1 PURPOSE

This document is created based on the requirement specification document. The purpose of this Software Design Specification (SDS) Document is to break down the FindAcademia project into components to describe in detail what the purpose of each component is and how it will be implemented.

2 PROJECT SCOPE

The scope of the **FindAcademia** project includes its distinct features, benefits, and limitations. The system's distinct features allow users to **search for the best IT and Management colleges in India** by using an intuitive web interface with filters for **city, state, category.**. The syste enables users to quickly find relevant colleges that match their criteria, streamlining the search process for prospective students. It solves the problem of manually searching for colleges by providing a simple, centralized solution.

3 SYSTEM OVERVIEW

This section provides an outline of the various components and subsystems of **FindAcademia**.

The system is composed of the following main components:

1. **Frontend (User Interface)**
    - This component is responsible for interacting with the user. It includes:
        - **Search Form:** Allows users to enter search criteria, such as city, state, category (IT or Management).
        - **Responsive Design:** Ensures the application adjusts to different device screen sizes, providing a seamless experience on mobile and desktop devices.
        - **Navbar:** Contains links to the 'Home', 'About', and 'Contact Us' pages.
2. **Backend (Data Handling & Processing)**
    - This component is responsible for handling user requests and processing data. It includes:
        - **Search Functionality:** Retrieves results from the database based on the search criteria.
        - **Database Interaction:** Handles queries to the database to fetch information about colleges, such as their name, address, courses offered, and contact details.
3. **Database (MySQL)**
    - The database stores all information about colleges, including:
        - **College Information:** College name, address, city, state, category (IT or Management), courses offered, and contact details.
        - **Query Management:** Efficiently processes user queries to return relevant results.

4. **Hosting Environment**
    - The web application is deployed on a local server for development and testing purposes, allowing users to access it through devices connected to the same network.

3 DESIGN CONSIDERATIONS

This section describes the requirements, assumptions, and dependencies that are considered to devise a complete design solution for **FindAcademia**.

3.1 Requirements

The design requirements for **FindAcademia** as identified in the Software Requirements Specification (SRS) document include:

- **User Interface Requirements:**
    - A search form that allows users to search for colleges by city, state, category (IT or Management), and course.
    - Responsive design to ensure usability across devices like mobile phones, tablets, and desktop computers.
- **Functional Requirements:**
    - A backend that processes the search query and retrieves relevant data from the database.
    - A database that stores details such as college name, address, contact info, and offered courses.
- **Performance Requirements:**
    - The system should be able to handle multiple search queries simultaneously.
    - Fast response time to display the search results to users.
- **Security Requirements:**
    - Secure handling of data input to avoid SQL injection or other security vulnerabilities.

3.2 Assumptions

The following assumptions were made in designing the **FindAcademia** system:

- Users will have internet access to use the web application.
- The application will be primarily accessed through modern web browsers.
- The college data stored in the database is reliable and up-to-date.
- Users will use valid search inputs (e.g., existing cities, states, or categories).

3.3 Dependencies

The design of **FindAcademia** is dependent on the following:

- **Technology Stack:**
  - The frontend depends on HTML, CSS (with Bootsrap), and JavaScript for responsive and interactive design.
  - The backend depends on Node.js and Express for handling requests and interacting with the database.
- **Database Management:**
  - The system relies on MySQL for storing and retrieving college information.
- **External Tools and Libraries:**
  - Dependencies on Bootstrap CSS for styling, jQuery for additional interactivity, and any Node.js modules for backend processing.
- **User Devices and Browsers:**
  - The application's performance is dependent on the user's device capability and browser compatibility.

4 Requirements :

4.1.1 1. Functional Requirements

- **Search Functionality**:
  - Users should be able to search for colleges by **city**, **state**, **category** (IT or Management).
- **Display Search Results**:
  - The system should display relevant college information such as college name, address, contact details, category, and courses offered based on user input.
- **Filter Options**:
  - The system should allow users to filter search results using multiple criteria (e.g., category, state, city).
- **Database Management**:
  - The system should store all college-related data (college name, city, state, category, address, contact details, courses offered) in a MySQL database.
  - Database should be queried dynamically based on user search input.
- **Responsive UI**:
  - The application should be responsive and should adjust to different screen sizes (desktop, tablet, mobile).
- **Navigation**:
  - Users should be able to navigate between the "Home," "About," and "Contact Us" pages.
- **Data Validation**:
  - The system should validate user inputs to ensure valid search parameters are entered.

4.1.2 2. Non-Functional Requirements

- **Performance**:
  - The system should deliver search results within 2-3 seconds under normal usage conditions.
- **Scalability**:

    - The database and backend should be designed to handle increasing data and users without significant performance degradation.
- **Security**:
    - If user authentication is implemented, passwords should be securely stored using hashing algorithms.
    - The app should use HTTPS for secure communication between the client and server.
- **Usability**:
    - The application should have a clean, intuitive interface.
    - Search results should be easy to read and comprehend.
- **Compatibility**:
    - The system should be compatible across browsers (Chrome, Firefox, Safari) and devices (laptops, tablets, smartphones).
- **Maintainability**:
    - The codebase should follow good programming practices, including modularization, comments, and adherence to coding standards, to ensure maintainability.

4.1.3 3. Hardware Requirements

- Client-Side:
    - A device capable of running a web browser (smartphone, tablet, desktop, or laptop).
- Server-Side:
    - A server capable of running Node.js with a connection to a MySQL database.

4.1.4 4. Software Requirements

- **Frontend**:
    - HTML, CSS, JavaScript (jQuery, Bootstrap for responsiveness).
- **Backend**:
    - Node.js, Express.js.
- **Database**:
    - MySQL for database management and storage.
- **Development Tools**:
    - Visual Studio Code (or another code editor).
    - MySQL Workbench (for database management).
    - Git (for version control).
- **Operating System**:
    - The system should work on Linux, macOS, and Windows.