



INTERNSHIP PROGRAM 2023

SOFTWARE DESIGN SPECIFICATION

Web Development

College Finder

Created By:	Faraj Momin	Approved By:	<Domain Lead Name>
Created On:	25-07-2024	Approved On:	DD-MMM-YYYY

INDEX

1	PURPOSE.....	2
2	PROJECT SCOPE.....	2
3	SYSTEM OVERVIEW.....	2
4	DESIGN CONSIDERATIONS.....	2
4.1	Requirements.....	2
4.2	Assumptions.....	2
4.3	Dependencies.....	2
5	SYSTEM ARCHITECTURE.....	2
5.1	Architectural Strategies.....	3
5.2	Structure & Relationships.....	4
6	DETAILED DESCRIPTION OF COMPONENTS.....	4
7	INTEGRATION.....	5
8	APPENDICES.....	1
8.1	Appendix A – Detailed Description of Components.....	1

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 project into components to describe in detail what the purpose of each component is and how it will be implemented. The SDS will also serve as a tool for verification and validation of the final product.

2 PROJECT SCOPE

The scope of the "College Finder" project includes providing a centralized platform for students to search and discover colleges in India, with detailed information about courses offered and contact details of the institutions. The system aims to address the issues of fragmented and outdated information by providing a reliable, user-friendly interface.

3 SYSTEM OVERVIEW

The system consists of a web application with a database. The database stores information about colleges, including names, addresses, contact details, and courses offered. The frontend is a responsive website that allows users to search and View College details, with features like filtering by location and course type.

4 DESIGN CONSIDERATIONS

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

4.1 Requirements

- A comprehensive database of colleges and courses.
- A responsive and intuitive user interface.
- Accurate and up-to-date data.

4.2 Assumptions

- The data provided will be accurate and maintained regularly.
- Users will have access to the internet and a web browser.

4.3 Dependencies

The project depends on reliable data sources and an efficient hosting solution for the web application.

5 SYSTEM ARCHITECTURE

The software system architecture refers to the logical organization of a distributed system into software components. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team. An effective architecture serves as the conceptual glue that holds every phase of the project together

for all of its stakeholders, enabling agility, time and cost savings, and early identification of design risks.

The Software architecture:

- Defines structure of a system
- Defines behaviour of a system
- Defines component relationship
- Defines communication structure
- Balances stakeholder's needs
- Influences team structure
- Focuses on significant elements
- Captures early design decisions

Below some important characteristics which are commonly considered are explained.

Operational Architecture Characteristics:

- Availability
- Performance
- Reliability
- Low fault tolerance
- Scalability

Structural Architecture Characteristics:

- Configurability
- Extensibility
- Supportability
- Portability
- Maintainability

Cross-Cutting Architecture Characteristics:

- Accessibility
- Security
- Usability
- Privacy
- Feasibility

5.1 Architectural Strategies

- **Frontend Architecture:**

1. **User Interface (UI):**

A. Technologies Used: HTML, CSS, JavaScript

B. Design Components:

- **Responsive Design:** Ensures compatibility across different devices and screen sizes using CSS media queries.
- **Dynamic Elements:** Utilize JavaScript to create interactive elements such as search bars, college lists, and filter options.
- **Consistent Styling:** Apply a unified design language using CSS for buttons, headers, footers, and forms.

2. **Client-Side Data Management:**

- **Data Integration:** Store college data in JavaScript objects or arrays, enabling client-side filtering and searching without server requests.
- **State Management:** Use JavaScript functions and local storage to maintain state across user sessions.

- **Search Functionality:**

1. **Search and Filter Logic:**

- **JavaScript Functions:** Implement search and filtering directly in JavaScript, allowing users to find colleges based on input criteria.
- **Event Handling:** Use event listeners to capture user inputs and dynamically update the displayed results.

2. **User Feedback:**

- **No Results Handling:** Display messages for empty searches or when no matching colleges are found.

- **Data Protection:**

- **Client-Side Validation:** Validate user inputs in the browser to prevent erroneous data submissions and enhance user experience.
- **CORS Policies:** Implement cross-origin resource sharing policies to secure data when integrating external APIs, if any.

- **Performance Optimization:**

1. **Minification:**

- Minify HTML, CSS, and JavaScript files to reduce load times and improve performance.

2. Asset Optimization:

- Use image compression techniques and lazy loading for faster page load times.

- **Deployment and Hosting:**

1. Static File Hosting:

- **Hosting Provider:** Choose platforms like GitHub Pages, Netlify, or Vercel for hosting static websites.
- **Deployment:** Automate deployment using CI/CD pipelines to keep the application up-to-date with minimal effort.

- **Monitoring and Maintenance:**

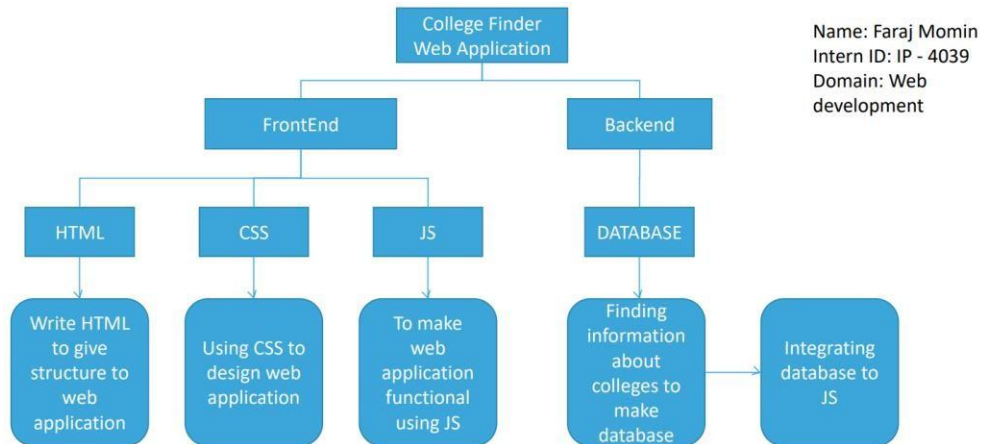
1. Error Tracking:

- Implement basic JavaScript error tracking for client-side issues using tools like Sentry or browser dev tools.

2. User Feedback:

- Enable feedback forms or survey links for continuous user input and application improvement.

5.2 Structure & Relationships



6 DETAILED DESCRIPTION OF COMPONENTS

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

The below template will be used to specify the details of all the components

Table 1: Detailed Design Specification Template

Identification	College Finder
Type	Web Application.
Purpose	To provide a searchable interface for users to find information about colleges and courses in India.
Subordinates	<ul style="list-style-type: none"> Search Interface College Details Page Course Details Page
Dependencies	<ul style="list-style-type: none"> Relies on the backend API for data retrieval. Interacts with the database for data storage.
Interfaces	<ul style="list-style-type: none"> API endpoints for college and course data. User interfaces for searching and viewing information.

Resources	<ul style="list-style-type: none">• Hosting server for the web application.• Database server for data management.
Processing	<ul style="list-style-type: none">• Handles user input, queries the database, and displays results.• Processes user actions such as searching and filtering.
Data	<ul style="list-style-type: none">• User search queries.• College and course information.

7 INTEGRATIONS

The system will integrate with a SQL database for data storage and retrieval. It may also connect to third-party APIs for additional data sources in the future.

8 APPENDICES

8.1 Appendix A – Detailed Description of Components

Identification	Search Interface
Type	Web Component
Purpose	Allows users to search for colleges and courses.
Subordinates	Search bar, filters.
Dependencies	Backend API for retrieving search results.
Interfaces	User interface, API.
Resources	Frontend resources like CSS and JavaScript files.
Processing	Captures user input and sends it to the backend API.
Data	User search terms.

Identification	College Details Page
Type	Web Component
Purpose	Displays detailed information about a selected college
Subordinates	College information sections (about, courses, contact).
Dependencies	Backend API for retrieving college details.
Interfaces	User interface, API.

Resources	Frontend resources like images and text data.
Processing	Fetches and displays data based on user selection.
Data	College details.