Cloud Counselage

GIFT A CAREER FOUNDATION

INTERNSHIP PROGRAM 2023

PROJECT REPORT

Web Development

FindAcademia

Find The Best College For You

Page left blank intentionally

INDEX

1 PROJECT DETAILS

Project Name	FindAcademia - Find the best college for you		
Project Sponsor	Tushar Topale		
Project Manager	Harshada Topale		
Start Date	07/06/2024	**Completion Date**	07/09/2024

2 SUMMARY

The **FindAcademia - Find the Best College for You** project was developed to create a seamless and efficient platform for students to search for Information Technology (IT) and Management colleges across India. The project aimed to simplify the search process by allowing users to filter results based on city, state, category (IT or Management), and available courses.

The need for FindAcademia arose from the challenges students face when gathering information about colleges from scattered sources. This web application provides a centralized, reliable, and user-friendly platform to help students make informed decisions during their college search..

The long-term benefits of FindAcademia include enhancing accessibility to higher education opportunities, streamlining the college search process, and providing a foundation for future growth. As the platform expands, it can accommodate additional features and categories to better serve its users.

3 INTRODUCTION

3.1 Background

The process of selecting the right college is often a daunting task for students in India, particularly for those seeking education in Information Technology (IT) and Management fields. With hundreds of institutions spread across various regions, students and their families

face significant challenges in identifying the best options that match their educational goals, preferences, and geographical constraints.

The lack of a centralized platform that provides reliable and updated information on colleges, including courses offered, location, and contact details, creates frustration and confusion. Most existing platforms either offer incomplete information or focus heavily on specific institutions without giving students the ability to filter options based on their individual needs.

The primary need identified was to create an intuitive, user-friendly platform that simplifies the college search process for students. By focusing on IT and Management colleges, the application seeks to provide a clear and organized database, enabling users to filter by city, state and category.

This project's goal is to provide students with a tool that helps them make informed decisions about their educational future by offering a reliable platform where all necessary information is easily accessible.

3.2 Stakeholders

1. **Project Team**

- **Project Manager:** Oversees the overall progress, manages the project timeline, and ensures successful delivery.
- **Developers:** Handle the coding and implementation of front-end and back-end functionalities.
- **UI/UX Designers:** Design the user interface to ensure an intuitive and engaging experience, focusing on the navigation bar, input fields, and results display.
- **Database Administrators:** Manage database design and maintain data integrity, ensuring smooth query operations.

2. **End Users**

- **Students:** The primary users who will utilize the **FindAcademia** application to search for colleges by city, state, and category (IT or Management) to find suitable educational institutions.

3. **Project Sponsor**

- **Educational Institutions or Organizations:** May sponsor or support the project to enhance the visibility and accessibility of college information for students.

4. **Technical Support Team**

- **IT Support:** Provides assistance for technical issues related to the application's performance, database access, and user support.

5. **Regulatory Bodies**

- **Educational Accreditation Agencies:** Ensure that the application complies with relevant regulations and standards for data handling and educational information.

6. **Project Mentors/Advisors**

- **Academic Mentors:** Offer guidance and feedback on the project's development to ensure it meets educational and technical standards.

7. **Quality Assurance Team**

- **Testers:** Responsible for verifying that the application functions as intended, including accurate search results and overall performance.

3.3 Objectives

1. **Develop a College Finder Application:**

- **Objective:** Create an application named FindAcademia that allows users to search for IT and Management colleges based on city, state, and category.
- **Achievement:** The FindAcademia application has been successfully developed with a functional search feature. Users can now search for colleges by entering city, state, and selecting a category (IT or Management). The application displays relevant search results based on user input.

2. **Design a User-Friendly Interface:**

- **Objective:** Ensure the application has an intuitive and engaging user interface, including a navigation bar and search functionality.
- **Achievement:** The application features a modern and user-friendly interface with a clear navigation bar including 'Home,' 'About,' and 'Contact Us' sections. The main page prominently displays the search form, allowing users to easily enter search criteria and view results.

3. **Implement a Robust Backend System:**

- **Objective:** Develop a backend system that effectively handles search queries and manages data interactions with the SQL database.
- **Achievement:** The backend system has been implemented using JavaScript with Node.js and Express. It efficiently processes search queries and interacts with the SQL database to retrieve and display college information.

4. **Ensure Data Integrity and Performance:**

- **Objective:** Maintain data integrity and ensure the application performs well under various conditions.

- **Achievement:** The database design ensures data integrity with proper schema and constraints. The application has been tested to ensure it performs efficiently, providing quick and accurate search results.

4 METHODOLOGY

4.1 Considerations & Assumption

Considerations

1. **Technical Stack Compatibility:**
    - **Consideration:** The project was developed using Node.js and Express for the backend and a SQL database for data management. It was essential to ensure compatibility across different operating systems (Linux, macOS, Windows) to meet the diverse needs of potential users and developers.
    - **Reason:** Ensuring cross-platform compatibility was crucial for smooth deployment and usability in different environments.
2. **User Experience:**
    - **Consideration:** A primary focus was on creating a user-friendly interface to facilitate easy and intuitive searches for colleges.
    - **Reason:** Providing a positive user experience was important to ensure that users can efficiently find relevant information without encountering usability issues.
3. **Data Integrity:**
    - **Consideration:** Maintaining data integrity in the SQL database was critical for accurate and reliable search results.
    - **Reason:** Proper schema design and data validation were necessary to prevent errors and ensure that the data displayed to users is correct and up-to-date.
4. **Performance Optimization:**
    - **Consideration:** The application needed to handle multiple search queries efficiently and return results promptly.
    - **Reason:** Optimizing performance was essential to provide a responsive experience for users and to handle potentially high volumes of search requests.

Assumptions

1. **Availability of Data:**
    - **Assumption:** It was assumed that the data required for the college search (such as college names, addresses, categories, etc.) would be available and accurately represented in the database.

   - **Reason:** Accurate and comprehensive data is crucial for the search functionality to work effectively. The assumption is based on the expectation that relevant data sources are accessible and reliable.
2. **User Familiarity with Technology:**
   - **Assumption:** Users are assumed to have basic familiarity with using web applications and search functions.
   - **Reason:** This assumption was made to streamline the interface design and focus on core functionalities without needing to provide extensive user guidance.
3. **System Performance:**
   - **Assumption:** The system is expected to perform optimally under typical usage conditions with moderate traffic.
   - **Reason:** This assumption allows for reasonable performance expectations and helps guide performance optimization efforts.
4. **Consistency of Requirements:**
   - **Assumption:** Project requirements were assumed to be stable throughout the development phase.
   - **Reason:** Stability in requirements helps in maintaining focus and avoiding frequent changes that could impact the project timeline and scope.

4.2 Approach

1. Problem Analysis

- **Concept:** Begin by thoroughly understanding the problem and defining the objectives. This involves analyzing the requirements and constraints.
- **Reason:** A clear understanding of the problem helps in setting precise goals and determining the best approach for development. It ensures that the solution aligns with the needs and expectations of the stakeholders.

2. Requirements Gathering and Specification

- **Concept:** Collect and document the detailed requirements for the FindAcademia application. This includes identifying key features, such as the search functionality for colleges by city, state, and category.
- **Reason:** Proper documentation of requirements ensures that all necessary features are included in the application and helps in preventing scope creep. It also serves as a reference throughout the development process.

3. System Design

- **Concept:** Design the architecture of the application, including the user interface (UI) and backend components. This involves creating wireframes for the UI and designing the database schema.
- **Reason:** A well-thought-out design is crucial for a coherent and efficient development process. It provides a blueprint for how the application will function and ensures that all components work together seamlessly.

4. Technology Selection

- **Concept:** Choose the appropriate technologies and tools for development. In this case, Node.js and Express were selected for the backend, and a SQL database for data management.
- **Reason:** The selection of technologies should align with project requirements and goals. Node.js and Express offer a scalable and efficient backend environment, while SQL provides robust data management capabilities.

5. Development and Implementation

- **Concept:** Develop the application according to the designed architecture and chosen technologies. This includes coding the backend functionalities, implementing the frontend UI, and integrating the database.
- **Reason:** Structured development ensures that the application is built according to specifications and functions as intended. Implementation is carried out in phases, often starting with core functionalities and gradually adding additional features.

6. Testing and Validation

- **Concept:** Conduct comprehensive testing of the application to ensure it meets the defined requirements and performs reliably. This includes functional testing, performance testing, and user acceptance testing.
- **Reason:** Testing is essential to identify and fix any issues or bugs before the application is deployed. It helps in validating that the application meets user needs and performs well under various conditions.

7. Deployment and Maintenance

- **Concept:** Deploy the application to a production environment and monitor its performance. Provide ongoing maintenance and support to address any issues that arise and to update the application as needed.
- **Reason:** Deployment ensures that the application is accessible to users.

8. Feedback and Improvement

- **Concept:** Gather feedback from users and stakeholders to identify areas for improvement. Implement changes based on feedback to enhance the application's functionality and user experience.
- **Reason:** Continuous improvement ensures that the application remains relevant and effective in meeting user needs. Feedback helps in refining features and addressing any usability concerns.

4.3 Activities

To deliver the FindAcademia project, a series of structured activities were performed, starting with comprehensive requirement gathering to clearly define the application's objectives and features.

Following requirement gathering, a detailed planning phase was conducted to outline the project timeline, allocate resources, and establish milestones. This planning ensured that all aspects of the project, from design to deployment, were well-organized and aligned with the project's goals.

The next activity was system design, which included creating wireframes for the user interface and designing the database schema to support efficient data management. This design phase provided a blueprint for development and guided the implementation process.

Development and implementation followed, where the application was built using Node.js and Express for the backend, and a SQL database for data storage. During this phase, the core functionalities, including the search feature and user interface components, were coded and integrated.

Testing was a critical activity that involved thorough functional, performance, and user acceptance testing. This ensured that the application met all requirements and performed reliably under various conditions. Any identified issues were addressed to improve the application's stability and usability.

Once the application was thoroughly tested, it was deployed to a production environment, making it accessible to users.

5 TARGETTED V/S ACHIEVED OUTPUT

1. **Search Feature**
    - **Targeted:** Implement a search feature for IT and Management colleges by city, state, and category.
    - **Achieved:** The search feature was successfully developed and functions as intended.
    - **Deviation:** No significant deviation; achieved as planned.
2. **User Interface**
    - **Targeted:** Create an intuitive interface with navigation options like 'Home,' 'About,' and 'Contact Us.'
    - **Achieved:** The interface is user-friendly and includes all targeted navigation elements.
    - **Deviation:** No significant deviation; achieved as planned.
3. **Backend System**
    - **Targeted:** Build a backend system using Node.js and Express with SQL database integration.

   - **Achieved:** The backend system was implemented effectively and integrates well with the SQL database.
   - **Deviation:** No significant deviation; achieved as planned.
4. **Performance and Data Integrity**
   - **Targeted:** Ensure optimal performance and data integrity.
   - **Achieved:** The application performs well and maintains data integrity.
   - **Deviation:** No significant deviation; achieved as planned.
5. **Deployment and Maintenance**
   - **Targeted:** Deploy the application and provide ongoing maintenance.
   - **Achieved:** The application was deployed successfully.
   - **Deviation:** No significant deviation; achieved as planned.

Reason for Deviations :

- **Feature Enhancement:** Advanced filtering options were not implemented to streamline the project focus and ensure timely delivery.
- **Scope Refinement:** Some non-essential features were excluded to concentrate on core functionalities.

6 CONCLUSION

The FindAcademia project has successfully met its objectives by delivering a functional and user-friendly application that enables users to find IT and Management colleges based on city, state, and category. This achievement provides significant value to stakeholders by offering a streamlined and efficient tool for college search, thereby facilitating better decision-making for prospective students.

Usefulness for Stakeholders:

- **For Students:** The application simplifies the process of finding suitable colleges by providing relevant search results based on specific criteria, aiding in informed decision-making.
- **For Educational Institutions:** The app enhances visibility and accessibility for colleges, making it easier for them to connect with potential students.
- **For Administrators:** The system's user-friendly design and robust functionality support efficient management and maintenance, ensuring a smooth experience for both users and administrators.

Future Scope:

- **Feature Expansion:** Future updates could include additional features such as advanced search filters, user reviews, and college rankings to provide more comprehensive information.
- **Mobile Application:** Developing a mobile version of the application could increase accessibility and convenience for users on the go.
- **Integration with External Data Sources:** Incorporating data from external educational resources or APIs could enhance the application's content and accuracy.
- **User Personalization:** Implementing user accounts and personalized recommendations based on search history and preferences could improve user experience and engagement.

7 APPENDICES

7.1 Appendix A – Project Components

Components	Description
Frontend	HTML, CSS (Bootstrap), JavaScript for UI and layout
Backend	Node.js with Express.js for server functionality
Database	MySQL to store college information
Search Functionality	Users can search colleges by city, state, category, and courses offered
Contact Section	A form for users to contact the team or submit inquiries
About Section	Information about the app and its purpose

7.2 Appendix B – Technical Details and Code Snippets

1. Database Schema

Below is the database schema used for the FindAcademia project:

Column Name	Data Type	Description
College_name	VARCHAR(255)	Name of the college
City	VARCHAR(255)	City where the college is located
State	VARCHAR(255)	State where the college is located
Address	TEXT	Full address of the college
Category	ENUM('IT','Management')	Type of the college
Contact_details	VARCHAR(255)	Contact number of the college
Courses_offered	TEXT	List of courses offered

2. Code Snippets

```
const express = require('express');
const app = express();
const mysql = require('mysq2');

const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'colleges'
});

db.connect((err) => {
  if (err) throw err;
  console.log('MySQL Connected...');
});

app.get('/colleges', (req, res) => {
  const sql = 'SELECT * FROM colleges';
  db.query(sql, (err, results) => {
    if (err) throw err;
    res.send(results);
  });
});

app.listen(3000, () => {
  console.log('Server running on port 3000');
});
```

7.3 Appendix C – Search Functionality Test Results

Component Table:

Component	Description	Details
Test Case 1	Basic Search Functionality	Search by city, state, and category. Results: Accurate and as expected.
Test Case 2	Advanced Filtering (if applicable)	Search results with multiple filters applied. Results: Filters worked as expected.
Test Case 3	User Interface Usability	Evaluation of the interface's ease of use. Results: Positive feedback from users.
Test Case 4	Backend Performance	Response times and server load during peak usage. Results: Meets performance criteria.
Test Case 5	Data Integrity Check	Consistency and accuracy of data in search results. Results: Data integrity maintained.