Full Stack \_OJT Plan for Students

The training plan with full-stack development for students is to equip them with comprehensive skills and knowledge to build modern web applications from front to back. Students will gain proficiency in designing user interfaces, implementing server logic, managing databases, and integrating front-end and back-end components for robust and scalable applications. This includes understanding both client-side (front-end) and server-side (back-end) development technologies, frameworks, and best practices. This course aims to prepare students for careers as versatile full-stack developers capable of handling end-to-end development tasks effectively.

Tools Required

* Computer / Laptop with good internet connection
* Code Editor (Notepad++, Sublime text, VS Code etc)
* Front-End Technologies (Java, HTML, CSS, JS)
* Back-End Technologies (Java Spring)
* Database Management
* Self-paced Content

**Master Class Sessions**: The master class, led by a central trainer, will be conducted online. These sessions are project-focused, covering everything from project initiation to project deployment.

**Doubt Solving session**: These sessions are conducted daily by the local trainer who is available at the centre.

**Self-Paced Content:** Self-paced content sourced from IBM Skillsbuild platform.

Full Stack Web Development

* Full stack development is to empower developers with the skills and knowledge required to build end-to-end web applications independently. This includes proficiency in both front-end and back-end technologies, allowing the students to work on all aspects of a project, from designing user interfaces and implementing client-side logic to managing server-side processes, databases, and APIs.
* A candidate will be able to:
* Master in front end, back end latest programs and frameworks, databases.
* Build industry relevant projects that will be customer centric.
* Finding solution of the problem by enterprise design thinking pathway.
* Connect a web application to backend server data base.
* Use MVC (Model View Controller) structure to create and build modern application designs

Project Titles

1. Task Management System: Create a web application for managing tasks and to-do lists. Utilize Spring Boot for backend logic such as user authentication and authorization, while using HTML/CSS/JavaScript for frontend interfaces.
2. Blog Platform: Develop a blog website where users can read and write articles. Use Spring Boot for managing blog posts and comments and utilize jQuery for features like article filtering and searching.
3. Online Booking System: Build a system for scheduling appointments or reservations, such as for a doctor's office or restaurant. Utilize Spring Boot for backend logic and jQuery for an intuitive booking interface with calendar integration.
4. Issue Tracking System: Create a system for tracking and managing software bugs or project tasks. Utilize Spring Boot for backend APIs and jQuery for interactive issue boards and task assignment.
5. Online Polling System: Develop a platform for conducting surveys and polls. Use Spring Boot for backend data storage and processing, and jQuery for dynamic survey forms and result visualization.

|  |  |
| --- | --- |
| **Weeks** | **Milestones** |
| Weeks 1-5 | **Initial Setup and Design** |
| - Problem statement and requirements analysis |
| - Wireframing and design layout of web pages |
| - Project management plan and prototyping documentation |
| - UX design principles integration report |
| - Create well-structured web pages with semantic elements |
| - Demonstrate form validation techniques |
| - Style text and design layouts using CSS properties and the box model |
| Weeks 6-10 | **Development of Interactive Features and Database Integration** |
| - Develop interactive and dynamic web pages with client-side scripting |
| - Integrate advanced JavaScript concepts for enhanced functionality |
| - Create responsive and engaging user interfaces |
| - Develop a fully responsive and visually appealing web application |
| - Utilize Bootstrap components, grids, and utilities |
| - Integrate custom styles and responsive features |
| - Create and implement a database, perform CRUD operations, and establish server connections |
| - Initialize a GitHub repository with README and project structure |
| - Develop servlets for handling HTTP requests and integrate JDBC in Java code |
| - Integrate Java Server Pages (JSP) for dynamic web page generation |
| Weeks 11-15 | **Spring Framework and Full-Stack Development** |
| - Implement basic Spring features (beans, configurations, application context) |
| - Develop web applications using Spring MVC |
| - Create RESTful APIs with Spring Boot and Spring MVC |
| - Showcase full-stack development with a functional and well-documented web application |
| Weeks 16-20 | **Testing, Deployment, and Finalization** |
| - Integrate front-end technologies with Spring Boot backend |
| - Develop a user-friendly front-end interface |
| - Test and debug the application |
| - Deploy the application on a suitable cloud platform |
| - Troubleshoot deployment issues |
| - Finalize the project, including code, deployment, and presentation documentation |
| - Present the project outcomes and deliver a final report |
| - Discuss possible extensions and improvements |

Weekly Plan for Students

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Weekly Learning Outcomes** | **Weekly Milestone** | **Self Paced Course** |
| **Week 1** | * Requirements Analysis and Documentation * Understand functional and non-functional requirements * Design Layout of the Prototype which is required for the project Initiation * Install Development Tools * Project Management * Software Development * Agile Methodology * Wireframing and Prototyping * Introduction to UX design * Build a Story based design * **Masterclass 1: Understanding Project Management and SDLC** | * By the end of this week, Understanding Project Requirements * Thorough understanding of project requirements, stakeholders, and objectives. * Ability to define clear project goals, target audience, and essential features. * Wireframes and User Flow Diagrams * Creation of comprehensive wireframes and user flow diagrams. * Functional and Non-functional Requirements. * Collection of comprehensive functional and non-functional requirements. | * <https://skills.yourlearning.ibm.com/activity/URL-FCFCC25FD8A9?channelId=CNL_LCB_1616447372894> * <https://skills.yourlearning.ibm.com/activity/URL-EAA89F6480FA?channelId=CNL_LCB_1616447372894> |
| **Week 2** | * Introduction to Java * Setting up Development Environment * Understanding Java basic syntax, Data Types * Variables and Constants * Operators * Control Flow Statements * Arrays * Methods * **Master Class 2: Java Jumpstart: Foundations of Java Programming** | * Basic Java Programming Proficiency. * Write and execute basic Java programs effectively. * Demonstrate a strong understanding of arrays and methods for data manipulation. * Work with collections of data using arrays. * User Input and Output Handling. * Design and implement Java programs that involve user input, data processing, and output generation. * Showcase the ability to apply Java programming principles in practical scenarios. * Debugging and Troubleshooting. * Gain skills in debugging and troubleshooting common programming errors. | * <https://www.codecademy.com/learn/learn-java> |
| **Week 3** | * Object-Oriented Programming (OOP) Concepts * Inheritance * Polymorphism * Interfaces * Abstract Classes * Exception Handling * Error Handling and Debugging * **Master Class 3: OOPS Concept and Exception Handling in Java** | * By the end of this week, students should be able to complete the following: * OOP Principles * Design well-structured Java code using OOP principles effectively. * Demonstrate proficiency in implementing inheritance, polymorphism, and interfaces. * Abstract Classes and Code Reusability * Design and implement abstract classes proficiently. * Understand their role in providing blueprints for derived classes and promoting code reusability. * Exception Handling Expertise * Gain a deep understanding of exception handling in Java. * Become familiar with Java's exception hierarchy, try-catch blocks, and best practices for handling different types of exceptions. | * <https://www.codecademy.com/learn/learn-java> * <https://skills.yourlearning.ibm.com/activity/URL--YSX-S5FCCI> |
| **Week 4** | * Collections Framework * Multithreading Concepts * Lambda expressions and streams * Stream API for processing collections * enabling powerful data manipulation operations * **Master Class 4: Streamlining Java: Lambda Expressions and Stream API** | * Students will master using lambda expressions and streams in Java for efficient data manipulation. * They will understand how lambda expressions work for concise functional programming. * Students will learn to use the Stream API to process collections effectively. * They'll perform operations like filtering, mapping, reducing, and parallel processing with streams. * Students will improve their Java applications' readability, maintainability, and performance using streams. * This milestone shows advanced proficiency in modern Java features for robust data manipulation. | * <https://www.redbooks.ibm.com/abstracts/crse0300.html> * [https://www.codecademy.com/learn/learn-java](https://www.codecademy.com/learn/learn-java%20) |
| **Week 5** | * HTML Fundamentals * Introduction to HTML * HTML Syntax * Semantic HTML * Creating Basic Web Pages * Validating HTML * HTML Forms, tables, Multimedia * Form Validation * CSS Basics, CSS Box Model * Styling Text * **Master Class 5: HTML Essentials: Mastering the Fundamentals** | * Students will demonstrate proficiency in HTML syntax, semantic markup, and creating basic web pages. * They will apply learned concepts to develop structured and semantically meaningful web pages. * Students will integrate multimedia content like images, audio, video, and iframes using HTML tags. * They'll utilize HTML5 semantic elements to improve document structure, accessibility, and SEO. * Students will learn CSS syntax, selectors, properties, and values for styling HTML elements. * They will apply the CSS box model concept to control spacing, sizing, and positioning of elements. * Students will style text elements using CSS properties for font customization, alignment, and decoration. * They'll acquire advanced CSS skills for creating modern, responsive, and visually appealing web layouts using Flexbox, CSS Grid, and CSS frameworks | * [https://skills.yourlearning.ibm.com/activity/URL-51AF1482839B](https://skills.yourlearning.ibm.com/activity/URL-51AF1482839B%20) * [https://skills.yourlearning.ibm.com/activity/ILB-EKVDPYPKGMKV155G](https://skills.yourlearning.ibm.com/activity/ILB-EKVDPYPKGMKV155G%20) |
| **Week 6** | * JavaScript Basics Introduction to JavaScript * Variables and Data Types * Operators and Expressions * Control Flow, Functions * Arrays and Objects * DOM Introduction * Events and Event Handling * Error Handling * **Master Class 6: CSS Unleashed: Exploring Style and Design, From Basics to Advanced Techniques** | * By the end of this week, * Explore JavaScript fundamentals like variables, data types, operators, expressions, control flow, and functions. * Integrate CSS and JavaScript to create interactive web elements, dynamic styles, animations, and user-friendly features. * Apply learned concepts in a project to develop a responsive web interface with dynamic content and interactivity. * Apply JavaScript concepts in another project to create dynamic content, handle user interactions, manipulate DOM elements, and implement error handling for a functional web application. | * [https://skills.yourlearning.ibm.com/activity/URL-4FF4AA6A2C77](https://skills.yourlearning.ibm.com/activity/URL-4FF4AA6A2C77%20) |
| **Week 7** | * JavaScript DOM Manipulation * DOM Manipulation Basics * Selecting DOM Elements * Modifying DOM Elements * Creating and Removing Elements * Traversing the DOM * JavaScript DOM Events * **Master Class 7: JavaScript Fundamentals: Building Blocks of Dynamic Web Development, DOM** | * By the end of this week, * Learn about the Document Object Model (DOM) structure, including nodes, elements, attributes, and relationships. * Explore methods for selecting DOM elements using selectors like IDs, classes, tags, attributes, and query selectors. * Practice modifying DOM elements dynamically with JavaScript by changing attributes, styles, content, classes, and event listeners. * Learn to create new DOM elements, add them to the document, manipulate their properties, and remove elements from the DOM hierarchy. * Understand DOM traversal techniques such as navigating parent, child, and sibling elements, accessing node properties, and iterating through DOM collections. | * [https://skills.yourlearning.ibm.com/activity/MDL-264](https://skills.yourlearning.ibm.com/activity/MDL-264%20) |
| **Week 8** | * Introduction to Bootstrap * Understanding and setting up Bootstrap * Bootstrap Components, Grid System * Advanced Bootstrap * Customizing Bootstrap * Bootstrap Layout Components * Responsive Design and themes with Bootstrap * Best Practices and Optimization * **Master Class 8: Bootstrap Demystified: Building Responsive Web Interfaces** | * By the end of this week, * Understand the basics of Bootstrap, including its purpose, features, and advantages in web development. * Explore Bootstrap's pre-designed components like navbar, cards, and modals, and its grid system for responsive layouts. * Customize Bootstrap styles, colors, typography, and create custom layout components using Bootstrap classes and utilities. * Apply responsive design principles with Bootstrap for mobile-friendly layouts and explore theme customization options. | * [https://skills.yourlearning.ibm.com/activity/MDL-266](https://skills.yourlearning.ibm.com/activity/MDL-266%20) |
| **Week 9** | * Familiarizing with DBMS * Working with MySQL * CRUD Operations * Connecting with Server * JDBC API * **Master Class 9: Database design principles, SQL queries, JDBC implementation** | * By the end of this week, student will be able to: * Understand the basics of Database Management Systems (DBMS) and their importance. * Install and use MySQL, including creating, modifying, and deleting databases and tables. * Practice CRUD operations (Create, Read, Update, Delete) in MySQL using SQL queries. * Learn to establish a connection between a Java application and a MySQL server. * Study and use the JDBC API for Java database interactions, including executing SQL queries and managing data. | * <https://skills.yourlearning.ibm.com/activity/URL-96CD3FFFA8E5> |
| **Week 10** | * Introduction to GitHub * Collaborating on GitHub * GitHub Branching Strategies * Basics of Servlet * JDBC connectivity using Java * Java Server Pages (JSP) * **Master Class 10: Advanced Concepts and Best Practices** | * By the end of this week, students will be able to: * Understanding Git and reposirories * Gain a comprehensive understanding of Java web development, covering Servlet basics, JDBC connectivity with Java, and Java Server Pages (JSP). * Develop a fully functional web application integrating these concepts: creating servlets to handle HTTP requests and using JDBC to interact with a database for data retrieval and manipulation. | * <https://skills.yourlearning.ibm.com/activity/URL-RGOJ5YH7EVK?channelId=CNL_LCB_1616516409884> |
| **Week 11** | * Expression Language (EL) * JSTL * Java Persistence API (JPA) * **Master Class 11: Employability Skills** | * By the end of this week, students will be able to: * Build a dynamic web application using Java technologies. * Integrate Expression Language (EL) and JavaServer Pages Standard Tag Library (JSTL) into JSP for efficient data handling. * Use EL to access and modify data in JSP, and leverage JSTL for tasks like loops and database interactions. * Utilize Java Persistence API (JPA) for managing data persistence, including CRUD operations in the database. * Showcase the ability to create dynamic and data-centric web applications with these Java technologies. | * <https://skills.yourlearning.ibm.com/activity/ILB-VQNPGXKKRPWVPKJM> |
| **Week 12** | * Basics of the Spring framework * Dependency injection and inversion of control (IoC) principles * Auto-configuration and convention-over-configuration features * Spring MVC * Spring RESTful Web Services * **Master Class 12: Employability Skills** | * By the end of this week, students will be able to: * Understand the core concepts and components of the Spring framework. * Learn about its architecture and how it simplifies Java application development. * Master the concepts of dependency injection and inversion of control (IoC). * Practice implementing IoC in Spring applications for loosely coupled and maintainable code. * Explore Spring Boot's auto-configuration. * Dive into Spring MVC (Model-View-Controller) architecture for building web applications. * Understand the role of controllers, views, and models in Spring MVC. * Learn about developing RESTful Web Services using Spring framework. | * [https://www.udemy.com/course/spring-framework-video-tutorial/](https://www.udemy.com/course/spring-framework-video-tutorial/%20%20) |
| **Week 13** | * Spring Boot Basics * Introduction to Spring Boot * Auto-Configuration * Starter Dependencies * Spring Boot Web Development * Spring Boot Data Access * **Master Class 13: Spring Boot Essentials: Rapid Development** | * By the end of this week, students will be able to: * Understand the basics of Spring Boot and its purpose in simplifying Java application development. * Learn about the introduction to Spring Boot, including its history and key features. * Explore auto-configuration in Spring Boot and how it automates setup and configuration. * Understand starter dependencies in Spring Boot and how they provide pre-configured dependencies for common tasks. * Dive into Spring Boot web development, including building web applications and RESTful APIs using Spring Boot. | * <https://www.udemy.com/course/spring-framework-video-tutorial/> |
| **Week 14** | * Getting Familiar with RESTFful API * RESTful API, Characteristics * Difference between REST API and RESTful API * **Master Class 14: Restful APIs** | * By the end of this week, students will be able to: * Familiarize yourself with Restful API basics, understanding its purpose and structure in web development. * Learn about the characteristics of Restful API, such as statelessness, uniform interface, and resource-based URLs. * Differentiate between REST API and RESTful API, understanding their similarities and differences in design and functionality. | * [https://www.udemy.com/course/restful-web-services-with-spring-framework-a-quick-start/](https://www.udemy.com/course/restful-web-services-with-spring-framework-a-quick-start/%20) |
| **Week 15** | * Methods of RESTful API (GET, POST, PUT, DELETE) * Spring Boot JDBC connection * Implement RESTful API endpoints in Spring Boot * **Master Class 15: Employability Skills** | * Explore the methods used in RESTful API, including GET for retrieving data, POST for creating data, PUT for updating data, and DELETE for deleting data. * Implement Spring Boot JDBC connection, understanding how to connect Spring Boot applications with databases using JDBC. * By the end of this week, students will be able to: * Test endpoints using Postman or similar tools to ensure functionality. * Integrate authentication mechanisms like JWT (JSON Web Tokens) or Spring Security into Spring Boot. | * <https://fullstackopen.com/en/part0/fundamentals_of_web_apps> |
| **Week 16** | * User Authentication and Authorization * Understand the architecture of a full-stack application * Integrate front-end technologies with the Spring Boot backend * Front-end user interface. * **Master Class 16: Secure Server-Side Validation: Ensuring Data Integrity** | * By the end of this week, students will be able to: * Implement user authentication for accessing protected endpoints. * Configure role-based authorization to control user access to specific resources. * Practice integrating databases with applications for effective data storage and retrieval. * Implement business logic to process and manipulate data efficiently. * Design and create front-end interfaces for a seamless user experience. | * <https://business.tutsplus.com/courses/master-powerpoint-15-essential-tips?_ga=2.67661270.658268391.1592849702-1543735797.1592586857> |
| **Week 17** | * Testing and Debugging * Unit testing, integration testing, and end-to-end testing of the full-stack application * Troubleshoot with all test cases * Use debugging tools and techniques to fix bugs * **Master Class 17: Employability Skills** | * By the end of this week, students will be able to: * Conduct integration testing to ensure different parts work together with front-end and back-end. * Explore testing and debugging techniques to ensure application reliability and quality. * Execute end-to-end testing to validate the entire application flow. * Identify issues found during testing. * Analyze problems and their causes. * Apply debugging techniques to fix bugs efficiently. | * <https://skills.yourlearning.ibm.com/activity/URL-EEUNAIZOWRU?channelId=CNL_LCB_1616516409884> |
| **Week 18** | * Introduction to cloud platforms * Deployment to Cloud Platform * Configure server environments. * Set up databases. * Manage application logs, and monitor performance metrics * **Master Class 18: Introduction to Cloud platform** | * By the end of this week, students will be able to: * Understand the basics of cloud computing and different cloud service providers. * Explore deployment processes and tools for cloud platforms. * Deploy a sample application to a chosen cloud platform. * Set up server configurations, including security settings and resource allocation. * Set up databases on the cloud platform | * <https://skills.yourlearning.ibm.com/activity/PLAN-2EC3A305F2C3> |
| **Week 19** | * Introduction to Web Hosting * Setting Up Environment and Infrastructure * Students need to deploy * project to final field environment to the free cloud providers like ‘OOOwebhost’ etc. * Students will prepare deliverables for reporting and presentations * **Masterclass 19: Enhancing the presentation skills** | * By the end of this week, students will be able to incorporate all the required suggestions. * Understand the concept of web hosting and its role in making websites accessible online. * Deploy the project to a staging or beta environment for testing by users or stakeholders. * Ensure the deployment process is smooth and the application functions correctly in the real environment. * Deploy the project to a free cloud provider like 'OOOwebhost' for final hosting | * [https://www.indeed.com/career-advice/career-development/tips-for-giving-a-great-presentation](file:///C:\Users\Khyati\Downloads\https:\www.indeed.com\career-advice\career-development\tips-for-giving-a-great-presentation) |
| **Week 20** | * Showcase of Final project | * After the completion of week, student will be able to showcase their final project and modify according to peer team suggestions. |  |

Project based evaluation for Students

|  |  |  |
| --- | --- | --- |
| **Project-based evaluation** | **Score** | **Milestones to be tracked by the evaluator** |
| Mid Evaluation  (after 10 weeks) | 50 | * Understanding & Implementation of the Problem statement (20) * Achievement of weekly milestone (15) * Q n A on coding / Model / Weekly milestone (15) |
| End Evaluation  (after 20 weeks) | 50 | * Implementation of the Project (Compete document, source code, ppt etc )(25) * Achievement of weekly milestone(15) * Q n A on coding / Model / Real world Implementation (15) |

Problem statement of Projects

Task Management System:

Problem Statement:

The project aims to develop a web application that enables users to manage their tasks and to-do lists efficiently. Leveraging Spring Boot for backend development facilitates the implementation of user authentication, authorization, and business logic, while HTML, CSS, and JavaScript are used to create intuitive and responsive frontend interfaces for users to interact with.

Requirements:

Backend Development: Implement the backend logic for the web application using Spring Boot, including user authentication, authorization, task management functionalities, and RESTful APIs for data manipulation.

User Authentication: Develop mechanisms for user registration, login, logout, and password management, ensuring secure access to the application and user data.

Authorization: Implement role-based access control (RBAC) to restrict access to certain features or resources based on user roles (e.g., admin, regular user).

Task Management: Create functionalities for users to create, update, delete, and mark tasks as complete or incomplete, supporting CRUD (Create, Read, Update, Delete) operations on task entities.

To-Do Lists: Enable users to organize tasks into to-do lists or categories, allowing for better organization and prioritization of tasks.

Frontend Development: Design and implement responsive frontend interfaces using HTML, CSS, and JavaScript frameworks/libraries (e.g., Bootstrap, jQuery) to provide an intuitive and user-friendly experience.

User Interface (UI): Develop UI components for displaying task lists, adding/editing tasks, setting due dates, and sorting/filtering tasks based on various criteria.

Interaction Design: Implement interactive features such as drag-and-drop functionality for reordering tasks, inline editing of task details, and real-time updates without page reloads.

Data Persistence: Configure a relational database (e.g., MySQL, PostgreSQL) to store user account information, task data, and related metadata securely.

Testing and Validation: Conduct comprehensive testing of the web application to ensure functionality, usability, and security. Perform validation checks on user input to prevent data corruption or injection attacks.

Background:

Task management is essential for personal productivity and organizational efficiency, helping individuals and teams stay organized and focused on their goals. Web-based task management applications provide users with convenient access to their tasks from anywhere with an internet connection, streamlining workflow and improving productivity.

Real-world Use Case:

Personal Productivity: Users can create to-do lists, set deadlines, and prioritize tasks based on urgency or importance, helping them stay organized and focused on completing their goals.

Team Collaboration: Task management applications facilitate collaboration and communication among team members by enabling them to share tasks, assign responsibilities, and track progress collectively.

Project Planning: Individuals and teams can use task management tools to plan and organize projects, break down tasks into smaller actionable items, and track project milestones and deadlines.

Time Management: Task management applications can help users allocate time effectively by scheduling tasks, setting reminders, and tracking time spent on each task or project.

Goal Tracking: Users can set and track long-term goals, milestones, and achievements using task management tools, providing motivation and accountability for personal and professional growth.

Blog Platform:

Problem Statement:

The project aims to develop a blog website that provides users with a platform to read and write articles on various topics. Leveraging Spring Boot for backend development enables efficient management of blog posts, comments, and user authentication, while jQuery enhances the user experience with dynamic features such as article filtering and searching.

Requirements:

Backend Development: Implement the backend logic for the blog website using Spring Boot, including data models for blog posts, comments, user authentication, and RESTful APIs for CRUD (Create, Read, Update, Delete) operations.

User Authentication: Develop mechanisms for user registration, login, logout, and password management, ensuring secure access to the blog platform and user data.

Blog Post Management: Create functionalities for users to create, update, delete, and view blog posts, supporting features such as rich text editing, image uploading, and categorization of posts.

Comment Management: Implement features for users to add, view, and delete comments on blog posts, ensuring proper moderation and interaction among users.

Frontend Development: Design and implement responsive frontend interfaces using HTML, CSS, and JavaScript frameworks/libraries (e.g., Bootstrap, jQuery) to provide an intuitive and visually appealing user experience.

Article Filtering: Utilize jQuery to implement features for filtering articles based on categories, tags, or other criteria, enabling users to discover relevant content easily.

Article Searching: Implement a search functionality using jQuery to allow users to search for articles by keywords, titles, or content, providing quick access to desired information.

User Profile: Create user profile pages where users can view their own blog posts, comments, and profile information, fostering a sense of community and personalization.

Responsive Design: Ensure that the blog website is responsive and accessible on various devices and screen sizes, providing consistent user experience across desktops, tablets, and smartphones.

Testing and Validation: Conduct comprehensive testing of the website to ensure functionality, usability, and security. Perform validation checks on user input to prevent data corruption or injection attacks.

Background:

Blogging has become a popular form of online expression, allowing individuals and organizations to share knowledge, opinions, and experiences with a global audience. Developing a blog website provides users with a platform to publish and discover content on diverse topics, fostering community engagement and knowledge sharing.

Real-world Use Case:

Content Publishing: Users can create and publish articles on topics of interest, sharing their expertise and insights with others in the online community.

Information Sharing: Blog websites serve as repositories of valuable information, enabling users to access tutorials, guides, reviews, and opinions on a wide range of subjects.

Community Interaction: Users can engage with each other through comments, discussions, and feedback on blog posts, fostering a sense of community and collaboration.

Thought Leadership: Blogging allows individuals and organizations to establish themselves as thought leaders in their respective fields, building credibility and influence within their industries.

Content Discovery: Readers can discover new content and authors by browsing through categories, tags, or search results, expanding their knowledge and perspectives on various topics.

Online Booking System:

Problem Statement:

The project aims to develop a versatile scheduling system that allows users to book appointments or make reservations for various services, such as medical appointments at a doctor's office or table reservations at a restaurant. Leveraging Spring Boot for backend development facilitates efficient management of scheduling logic and data persistence, while jQuery enhances the user experience with an intuitive booking interface integrated with a calendar for easy date selection.

Requirements:

Backend Development: Implement the backend logic for the scheduling system using Spring Boot, including data models for appointments, reservations, user authentication, and RESTful APIs for CRUD (Create, Read, Update, Delete) operations.

User Authentication: Develop mechanisms for user registration, login, logout, and password management, ensuring secure access to the scheduling platform and user data.

Appointment/Reservation Management: Create functionalities for users to schedule appointments or make reservations for services, supporting features such as selecting dates, times, services, and available providers or tables.

Calendar Integration: Utilize jQuery to integrate a calendar interface into the booking interface, allowing users to view available dates, select preferred dates, and navigate through different months or weeks easily.

Real-time Availability: Implement features to display real-time availability of appointment slots or reservation times based on current bookings and capacity constraints, ensuring accurate scheduling and preventing double bookings.

Confirmation and Notifications: Provide confirmation messages and notifications to users upon successful booking of appointments or reservations, including email or SMS notifications for reminders and updates.

Admin Dashboard: Create an admin dashboard for managing appointments, reservations, user accounts, and other system settings, providing administrators with tools for monitoring and managing the scheduling platform.

Responsive Design: Ensure that the scheduling system is responsive and accessible on various devices and screen sizes, providing a consistent user experience across desktops, tablets, and smartphones.

Testing and Validation: Conduct comprehensive testing of the system to ensure functionality, usability, and security. Perform validation checks on user input to prevent data corruption or injection attacks.

Performance Optimization: Optimize the performance of the scheduling system, including database queries, API response times, and frontend rendering, to provide a smooth and responsive user experience even under high traffic conditions.

Background:

Scheduling systems play a crucial role in managing appointments and reservations efficiently, helping businesses and service providers streamline operations and improve customer satisfaction. By developing a versatile scheduling platform, businesses can automate booking processes, reduce scheduling conflicts, and enhance overall efficiency.

Real-world Use Case:

Healthcare Services: Patients can schedule appointments with doctors, specialists, or medical facilities online, reducing wait times and improving access to healthcare services.

Restaurant Reservations: Diners can make table reservations at restaurants, cafes, or bars, ensuring availability and avoiding long waits during peak hours.

Beauty and Wellness Services: Clients can book appointments for spa treatments, salon services, or fitness classes, optimizing resource utilization and enhancing customer experience.

Educational Institutions: Students can schedule appointments with academic advisors, tutors, or counselors, facilitating academic support and guidance.

Professional Services: Clients can schedule appointments with lawyers, consultants, or financial advisors, streamlining service delivery and client management processes.

**Issue Tracking System:**

Problem Statement:

The project aims to develop a robust system for tracking and managing software bugs and project tasks efficiently. Leveraging Spring Boot for backend APIs enables seamless integration with existing systems and tools, while jQuery enhances the user experience with interactive issue boards and task assignment features, facilitating collaboration and productivity in software development teams.

Requirements:

Backend Development: Implement backend APIs for the bug and task tracking system using Spring Boot, including data models for bugs, tasks, projects, users, and RESTful APIs for CRUD (Create, Read, Update, Delete) operations.

User Authentication: Develop mechanisms for user authentication, authorization, and password management, ensuring secure access to the system and user data.

Bug Tracking: Create functionalities for users to report software bugs, including details such as bug description, severity, status, assigned developer, and related project information.

Task Management: Implement features for creating, assigning, and tracking project tasks, including task description, priority, status, due dates, and task dependencies.

Project Management: Enable users to create and manage projects, assign team members, and track project progress, including milestones, deadlines, and overall project status.

Issue Boards: Design and implement interactive issue boards using jQuery for visualizing bugs, tasks, and project activities, providing users with a clear overview of project status and progress.

Task Assignment: Implement features for assigning tasks to team members, setting priorities, and tracking task assignments and completions, facilitating effective task delegation and workload management.

Collaboration Tools: Provide collaboration tools such as comments, attachments, and notifications for users to communicate, share updates, and collaborate on bug fixes and project tasks.

Reporting and Analytics: Develop reporting and analytics tools for generating various reports and insights related to bug trends, task completion rates, project velocity, and team performance.

Responsive Design: Ensure that the bug and task tracking system is responsive and accessible on various devices and screen sizes, providing optimal user experience across desktops, laptops, tablets, and smartphones.

Testing and Validation: Conduct comprehensive testing of the system to ensure functionality, usability, and security. Perform validation checks on user input to prevent data corruption or injection attacks.

Background:

Effective bug and task tracking are essential for software development teams to identify, prioritize, and resolve issues efficiently, ensuring the quality and reliability of software products. Developing a comprehensive tracking system streamlines bug management, task assignment, and project coordination, improving team collaboration and productivity.

Real-world Use Case:

Software Development: Development teams can use the bug and task tracking system to report, prioritize, and resolve software bugs, track project tasks, and manage project timelines effectively.

Agile Project Management: Agile teams can utilize the system to implement agile methodologies such as Scrum or Kanban, facilitating sprint planning, backlog grooming, and sprint retrospectives.

Quality Assurance: QA teams can use the system to perform software testing, report defects, and track bug fixes, ensuring the delivery of high-quality software products to end users.

Project Coordination: Project managers can use the system to coordinate project activities, allocate resources, and monitor project progress, ensuring timely delivery of project milestones and deliverables.

Client Collaboration: Clients and stakeholders can access the system to track project status, provide feedback, and prioritize feature requests or bug fixes, fostering transparency and collaboration between development teams and clients.

Online Polling System:

Problem Statement:

The project aims to develop a versatile platform for creating and conducting surveys and polls to collect feedback, opinions, and insights from users. Leveraging Spring Boot for backend data storage and processing ensures efficient management of survey data and responses, while jQuery enhances the user experience with dynamic survey forms and interactive result visualization, enabling users to create, participate in, and analyze surveys effectively.

Requirements:

Backend Development: Implement backend services for the survey and poll platform using Spring Boot, including data models for surveys, questions, responses, users, and RESTful APIs for CRUD (Create, Read, Update, Delete) operations.

User Authentication: Develop mechanisms for user registration, login, logout, and password management, ensuring secure access to the platform and user data.

Survey Creation: Create functionalities for users to create and customize surveys and polls, including adding, editing, and deleting survey questions, specifying response types, and setting survey parameters (e.g., anonymity, participation restrictions).

Survey Participation: Implement features for users to participate in surveys and polls by providing responses to survey questions, ensuring user-friendly interfaces and seamless survey submission processes.

Result Visualization: Utilize jQuery to create dynamic result visualization components for displaying survey results in charts, graphs, or tables, enabling users to analyze and interpret survey data effectively.

Data Analysis: Develop tools for analyzing survey responses, generating summary statistics, and identifying trends or patterns in survey data, providing valuable insights for decision-making and planning.

User Management: Enable administrators to manage user accounts, permissions, and roles, including features for adding/removing users, resetting passwords, and monitoring user activity.

Reporting and Exporting: Provide functionalities for generating and exporting survey reports in various formats (e.g., PDF, CSV), allowing users to share survey findings with stakeholders and collaborators.

Responsive Design: Ensure that the survey and poll platform is responsive and accessible on various devices and screen sizes, providing optimal user experience across desktops, laptops, tablets, and smartphones.

Testing and Validation: Conduct comprehensive testing of the platform to ensure functionality, usability, and security. Perform validation checks on user input to prevent data corruption or injection attacks.

Background:

Surveys and polls are valuable tools for collecting feedback, conducting research, and gathering opinions from individuals or groups. Developing a user-friendly and feature-rich survey platform enables organizations, businesses, and researchers to create, distribute, and analyze surveys efficiently, facilitating data-driven decision-making and insights generation.

Real-world Use Case:

Market Research: Businesses can use the survey platform to conduct market research, gather customer feedback, and assess market trends and preferences, informing product development and marketing strategies.

Employee Feedback: Organizations can utilize the platform to collect feedback from employees on workplace culture, satisfaction, and engagement, facilitating employee feedback loops and continuous improvement initiatives.

Academic Research: Researchers and academics can use the platform to design and distribute surveys for academic studies, research projects, and data collection efforts, enabling data-driven research findings and insights.

Customer Satisfaction: Service providers can use the platform to measure customer satisfaction, identify areas for improvement, and gather insights for enhancing customer experience and loyalty.

Public Opinion Polling: Media organizations and political campaigns can leverage the platform to conduct public opinion polls, gauge public sentiment, and gather insights on social and political issues.

# Guidelines for Selecting Custom Project

### Follow the below guidelines, if you are selecting your own topic for project

Students need to choose from two technology tracks**: Artificial Intelligence (AI) and Web Development.** Your task is to create innovative solutions linked to **specific social challenges** through your problem statements.

**Social Challenges**

Your project should aim to address one of the following social challenges:

1. Eradicating Poverty and Hunger

* Solutions promoting health care, sanitation, preventive health care, and safe drinking water.
* Awareness to initiatives promoting sanitation, clean water, best agricultural practices.

2. Improving Education

* Enhancing education, vocational skills for children, women, elderly, differently-abled individuals.
* Livelihood enhancement projects.

3. Promoting Gender Equality

* Solutions for empowering women, setting up homes and hostels for women and orphans.
* Facilities for senior citizens and reducing inequalities for socially and economically backward groups.

4. Environmental Sustainability

* Safeguarding ecological balance, protection of flora and fauna, and conservation of natural resources.
* Initiatives for maintaining soil, air, and water quality, including river rejuvenation projects.

5. Protecting National Heritage and Culture

* Restoration of historical sites and works of art.
* Promotion and development of traditional arts and handicrafts.

6. Supporting Armed Forces and Their Families

* Benefits for veterans, war widows, their dependents, and Central Armed Police Forces personnel.

7. Promoting Sports

* Training and support for rural, nationally recognized, Paralympic, and Olympic sports.

8. Rural Development Projects

* Solutions aimed at improving rural infrastructure and living conditions.

9. Slum Area Development

* Projects focused on the development of areas declared as slums by government authorities.

10. Disaster Management

* Initiatives for relief, rehabilitation, and reconstru
* cted areas.

# Guidelines for Creating Problem Statement

Your problem statement should clearly outline the following:

1. The Social Challenge

* Identify which social challenge from the list above your project will address.

2. The Technology Track

* Specify whether your solution will use AI or Web Development technologies.

3. The Solution

* Describe the project you plan to create, including its goals, functionalities, and how it leverages the chosen technology to address the social challenge.

4. Impact

* Explain the potential impact of your solution on the community or target audience.

# Guidelines for Submitting Problem Statement

* Prepare it in word document
* Mentioned the below details in proposal document
  + Project title (should be concise and clear)
  + Description (should not exceed 500 words)
  + Objective (should not exceed 200 words)
  + Opportunity (should not exceed 200 words)
* Submit your problem Statement by 22nd of May

Reference document:

