1. **Project Setup and Planning**
   * Set up a version control system (e.g., Git) and create a repository for the project.
   * Define the project requirements and goals based on the client's specifications.
   * Create a project timeline with milestones and deadlines.
   * Determine the technology stack and development tools to be used.
2. **Frontend Development**
   * Design the website layout and user interface based on the client's preferences and requirements.
   * Convert the design into HTML/CSS templates using frameworks like Bootstrap or Tailwind CSS.
   * Implement interactivity and dynamic behavior using JavaScript and frontend frameworks like React.js or Vue.js.
   * Ensure the website is responsive and compatible with various devices and screen sizes.
   * Integrate any third-party libraries or APIs for additional functionality (e.g., Google Maps, social media integration).
3. **Backend Development**
   * Set up the backend server using a suitable programming language and framework (e.g., Node.js with Express, Python with Django).
   * Define and create the database schema based on the website's data requirements (e.g., MySQL, MongoDB).
   * Implement server-side logic for handling user authentication, authorization, and session management.
   * Develop API endpoints for CRUD operations (Create, Read, Update, Delete) and any other required functionalities.
   * Integrate third-party services or APIs for features like payment processing or email notifications.
4. **Database Management**
   * Set up and configure the chosen database system.
   * Design and create database tables and relationships according to the defined schema.
   * Optimize database queries and ensure efficient data retrieval and storage.
   * Implement database backups and security measures to protect sensitive data.
5. **Testing and Quality Assurance**
   * Write unit tests for both frontend and backend code to ensure functionality and reliability.
   * Perform integration testing to verify the interaction between different components of the application.
   * Conduct user acceptance testing (UAT) to validate that the website meets the client's requirements and expectations.
   * Address any bugs or issues identified during testing and ensure they are resolved before deployment.
6. **Deployment and Hosting**
   * Choose a suitable hosting provider and deploy the website to a production environment.
   * Configure server settings and domain settings (if applicable) for the live website.
   * Set up SSL certificates for secure HTTPS connections.
   * Monitor server performance and uptime to ensure the website is running smoothly.
7. **Maintenance and Support**
   * Provide documentation and training for the client to manage and update the website content.
   * Offer ongoing maintenance and support services to address any future updates, bug fixes, or enhancements.
   * Monitor website analytics and performance metrics to identify areas for improvement and optimization.