# Generated PRD

{  
 "Project Overview": "Create a content management system (CMS) with Flask and SQLAlchemy. The CMS will enable users to create, edit, and manage digital content with ease. Flask will serve as the web framework, providing a lightweight and modular structure, while SQLAlchemy will handle database interactions. The CMS will support features such as content versioning, user roles, and a customizable frontend. The goal is to deliver a flexible and user-friendly CMS solution suitable for various content-driven websites.",  
 "Original Requirements": [  
 "Functional Requirements: \n1. User should be able to create new content \n2. User should be able to edit existing content \n3. User should be able to manage content versions \n4. User should be able to assign user roles \n5. User should be able to customize frontend \n\nNon-functional Requirements: \n1. User-friendly interface \n2. High performance and scalability \n3. Secure data storage and access"  
 ],  
 "Project Goals": [  
 "1. Develop a flexible and user-friendly CMS solution \n2. Ensure high performance and scalability \n3. Implement secure data storage and access"  
 ],  
 "User Stories": [  
 "1. As a website administrator, I want to be able to create new content easily so that I can keep the website updated. \n2. As a content editor, I want to be able to edit existing content quickly so that I can make necessary changes. \n3. As a website owner, I want to be able to manage content versions effectively so that I can track changes and revert if necessary. \n4. As a website administrator, I want to be able to assign user roles to control access to content. \n5. As a website designer, I want to be able to customize the frontend to match the website's design."  
 ],  
 "System Architecture": [  
 "The CMS will consist of a web server, application server, and database. \nThe web server will run Flask, which will handle the web requests and serve the frontend. \nThe application server will run SQLAlchemy, which will handle database interactions. \nThe database will be SQLite, providing a lightweight and portable solution. \nThe web server and application server will communicate through an API, allowing for a modular and scalable structure."  
 ],  
 "Tech Stacks": [  
 "Python", "Flask", "SQLAlchemy", "SQLite"  
 ],  
 "Requirement Pool": [  
 "P0 - Create new content \nP0 - Edit existing content \nP0 - Manage content versions \n\nP1 - Assign user roles \n\nP2 - Customize frontend"  
 ],  
 "UI/UX Design": [  
 "The UI/UX design will be clean and user-friendly. The color scheme will match the website design, and the layout will be responsive. The frontend will include features such as a WYSIWYG editor, version control, and user role management. Navigation will be intuitive, and the interface will be easy to use."  
 ],  
 "Development Methodology": [  
 "The development methodology will be Agile, with iterative and incremental development. Each iteration will last 2 weeks, and the project will be divided into sprints. Testing will be performed throughout the development process, and deployment will occur after each sprint. The team will consist of a project manager, frontend developer, backend developer, and database administrator."  
 ],  
 "Security Measures": [  
 "The CMS will use HTTPS to ensure secure communication between the web server and client. User passwords will be hashed and salted before being stored in the database. Access controls will be implemented to restrict access to sensitive data. The database will be regularly backed up to prevent data loss."  
 ],  
 "Testing Strategy": [  
 "Unit testing will be performed on each component, including the web server, application server, and database. Integration testing will be conducted to ensure all components work together correctly. User acceptance testing will be performed by stakeholders to ensure the CMS meets their requirements."  
 ],  
 "Scalability and Performance": [  
 "The CMS will be designed to handle increased load by implementing a load balancer and clustering. The system will be monitored regularly to identify potential performance bottlenecks and address them accordingly."  
 ],  
 "Deployment Plan": [  
 "Software updates will be deployed after each sprint, with thorough testing performed before deployment. Hardware deployment will occur at the beginning of the project, with regular maintenance and updates as needed."  
 ],  
 "Maintenance and Support": [  
 "Maintenance and support will be provided by the team after the project is completed. The team will be available for issue resolution and updates as needed."  
 ],  
 "Risks and Mitigations": [  
 "Risk: Development delays due to technical issues \nMitigation: Regular testing and thorough planning for each sprint \n\nRisk: Data loss due to server failure \nMitigation: Regular backups and monitoring of server health \n\nRisk: Security breaches \nMitigation: Implementation of secure communication and access controls"  
 ],  
 "Compliance and Regulations": [  
 "The project will comply with relevant regulations and standards, including the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). Any necessary certifications will be obtained before deployment."  
 ],  
 "Budget and Resources": [  
 "The budget for the project is $100,000. This includes the cost of hardware, software, and resources. The team will consist of 4 members, with the project manager making $60,000, frontend developer making $50,000, backend developer making $70,000, and database administrator making $60,000."  
 ],  
 "Timeline and Milestones": [  
 "Project Start - 8th December 2023 \n\nMilestone 1: Hardware Deployment - 8th December 2023 \nMilestone 2: Sprint 1 - 22nd December 2023 \nMilestone 3: Sprint 2 - 5th January 2024 \nMilestone 4: Sprint 3 - 19th January 2024 \nMilestone 5: Sprint 4 - 2nd February 2024 \nMilestone 6: Sprint 5 - 16th February 2024 \nMilestone 7: Sprint 6 - 1st March 2024 \nMilestone 8: Sprint 7 - 15th March 2024 \nMilestone 9: Sprint 8 - 29th March 2024 \nMilestone 10: Deployment - 5th April 2024 \nMilestone 11: Maintenance and Support - Ongoing"  
 ],  
 "Communication Plan": [  
 "Regular team meetings will be held to discuss project progress and address any issues that arise. Status updates will be provided to stakeholders after each sprint. Communication will primarily occur via email, with video calls as needed for more in-depth discussions."  
 ],  
 "Anything UNCLEAR": [  
 "If there is any unclear information in this PRD, please feel free to ask for clarification. We want to ensure that all stakeholders have a clear understanding of the project and its goals."  
 ]  
}