Flash

# Question 1

**What would you do to enhance performance of your project?**

**Optimise Database Queries**: Ensure that database queries are optimised by using appropriate indexes, minimising the number of queries, and optimising the structure of the database.

**Implement Caching:** Utilise caching mechanisms to store frequently accessed data in memory, reducing the need to fetch data from the database repeatedly.

**Asynchronous Processing:** Utilise asynchronous programming techniques to handle I/O-bound operations, such as database queries or external API calls, asynchronously, allowing the server to handle more concurrent requests.

**Load Balancing:** Implement load balancing to distribute incoming requests across multiple server instances, ensuring better utilisation of resources and improved response times.

**Horizontal Scaling:** Scale the application horizontally by adding more server instances or utilising cloud services to handle increased traffic and workload.

**Performance Monitoring and Profiling:** Implement performance monitoring and profiling tools to identify performance bottlenecks, analyse resource utilisation, and optimise critical sections of the codebase.

**Regular Code Review and Refactoring:** Conduct regular code reviews and refactoring sessions to identify and eliminate redundant code, optimise algorithms, and improve overall code quality, which indirectly enhances performance.

# Question 2

**What additional enhancements would add to the project to make it more complete ?**

**User Authentication and Authorization:** Implement better user authentication and authorization mechanisms to ensure that only authorised users can access sensitive functionalities and data.

**Logging and Error Handling:** Enhance logging and error handling mechanisms to capture and log errors, exceptions, and critical events, facilitating troubleshooting and debugging.

**Security Measures**: Implement additional security measures such as input validation, data encryption, and protection against common security vulnerabilities like SQL injection and cross-site scripting (XSS).

**Automated Testing:** Enhance test coverage by implementing automated unit tests, integration tests, and end-to-end tests to ensure the reliability and robustness of the application.

**Performance Monitoring and Analytics:** Integrate performance monitoring and analytics tools to track key performance metrics, identify performance bottlenecks, and optimise resource utilisation.

# Question 3

**Walkthrough of how you think this should be deployed in a production environment.**

**Build the Application:** Ensure that the application is properly built and tested locally. Make sure all dependencies are included and configured correctly.

**Prepare for Deployment:**

* Make sure you have access to a server or cloud platform where you intend to deploy the application.
* Prepare the production environment by installing necessary software dependencies like .NET runtime, SQL Server (if applicable), etc.
* **Configure Environment Variables:** Ensure that sensitive information such as database connection strings, API keys, and other configurations are stored securely as environment variables. These can be set directly on the server or managed using a configuration management tool.

**Deploy the Application:**

* You can deploy the application directly to the server.
* **Configure Web Server:** If you're deploying to a web server like IIS, configure the server to host the ASP.NET Core application.

**Database Setup:** Ensure that the database is properly configured and accessible from the deployed application. This may involve setting up database users, permissions, and ensuring proper network configurations.

**Monitoring and Logging:** Set up monitoring and logging for the deployed application to track its health, performance, and any errors or exceptions that occur.

**Security Considerations:** Ensure that security measures are in place to protect the application from common threats like SQL injection and other vulnerabilities. This may involve implementing authentication and authorization mechanisms, securing API endpoints, and regularly applying security patches.

**Testing:** Perform thorough testing of the deployed application to ensure that it functions correctly in the production environment. This may involve testing various scenarios, load testing to ensure scalability, and security testing to identify and address any vulnerabilities.

**Backup and Disaster Recovery:** Implement backup and disaster recovery strategies to ensure that data is protected and can be recovered in case of unexpected failures or disasters. This may involve regular backups of the database and application data, as well as setting up redundancy and failover mechanisms.