First Architectural Design Review for the Meet with ME web application

Introduction:

The following is an architectural design review for a JavaScript Node.js web application that will be hosted on Azure cloud platform. The web application will be implemented using mainly JavaScript but other languages will be used as necessary. The application will allow students to schedule a consultation with their lecturers. Both lecturers and students should be able to see the bookings as well as see any changes made by the consultation organiser. The application will need to adhere to basic web security practices such as password hashing.

Architecture:

The application will be hosted on the Azure cloud platform which provides a secure and scalable environment for hosting web applications. Azure also provides a wide range of services that can be used to enhance the functionality of the application, such as Azure Functions, Azure Active Directory, and Azure Cosmos DB.

Web Security:

To ensure the security of the application, basic web security practices will be implemented, such as password hashing. The application will use the bcrypt library to hash user passwords before storing them in the database. In addition, the application will use HTTPS to encrypt all traffic between the client and server.

Authorization and Authentication:

To ensure that only authorized users can access the application, the application will implement a login system. Users will be required to provide valid credentials (e.g., email and password) to gain access to the application. The application will use Azure Active Directory to authenticate users.

Logging and Monitoring:

To ensure the accountability of the application, a logging and monitoring system will be implemented. The application will log all actions taken by both students and lecturers, including the date and time of the action, the nature of the action, and the identity of the user who initiated the action. The logs will be stored in Azure Cosmos DB.

Possible problems we might run into:

1. Technical challenges: Developing a web application that meets all the requirements can be technically challenging. We may encounter issues with database management, authentication, security, and scalability.
2. User experience: It is important to ensure that the web application is intuitive and easy to use.
3. Time management: We need to ensure that we manage our time effectively and meet all project deadlines. Weekly sprints will be used to make sure that the project stays on tract.
4. Communication: Good communication is critical for the success of the project. We need to ensure that all team members are on the same page and that we communicate regularly to avoid misunderstandings and ensure that everyone is making progress.
5. Security: We need to ensure that the web application is secure, and that user data is protected. This may involve implementing security measures such as encryption, password hashing, and access control.

Possible solutions to prevent the problems listed:

1. Lack of coordination: To prevent issues caused by a lack of coordination, we can assign roles and responsibilities to team members, establish clear communication channels, and set up regular meetings to ensure that everyone is on the same page. Additionally, using project management tools such as TextUSM can help keep tasks organized and ensure that everyone is aware of what needs to be done.
2. Technical challenges: Technical challenges are common in software development, but we can mitigate these issues by performing thorough testing, conducting code reviews, and using version control software such as Git together with regular sprints. Additionally, we can set up a development environment that is similar to the production environment, so we can test the application in an environment that closely resembles where it will be used.
3. Security concerns: To address security concerns, we can use best practices such as hashing user passwords, encrypting sensitive data, and implementing role-based access control to ensure that only authorized users can access certain parts of the application. Additionally, we can use third-party tools such as Microsoft Azure Security Centre to monitor the application for potential security threats.
4. Usability challenges: To improve usability, we can conduct user testing to identify areas where users might struggle with the application and make improvements based on their feedback. We can also follow best practices for user interface design, such as keeping the interface simple and intuitive, using clear and concise language, and providing helpful error messages.

First Major Decision:

Does the group effectively work together and can the first sprint goals be achieved. Is the code well structured and broken up so that a well organised environment is established for subsequent sprints.

Conclusion:

In conclusion, the proposed architecture for the Node.js web application using Azure provides a secure and scalable environment for developing and hosting the application. By implementing basic web security practices such as password hashing, authentication, and HTTPS, the application can ensure the confidentiality and integrity of user data. By implementing a logging and monitoring system, the application can ensure accountability and traceability of all actions taken by users.