Architecture Goals

Requirement	What	Why	How
Performance and Efficiency	how well a system or application executes its tasks and uses resources	• High performance ensures quick response times, leading to a smoother and more satisfying user interaction • Efficient systems handle high loads and complex tasks without crashing or slowing down, maintaining consistent operation • Efficient resource use reduces operational costs, as the system requires fewer resources to perform well, leading to savings in hardware, energy, and maintenance	• Write clean, efficient code to reduce processing time and resource usage • Add indexes in MongoDB to speed up search operations
Scalability	Design the system to accommodate future growth, managing an escalation in users or data volume efficiently	Scalability ensures systems manage increased users or data without performance decline, facilitating growth and long-term sustainability	• Use databases like MongoDB that can scale horizontally through sharding to handle growing data volumes
User Interface	User Interface (UI) is the design and elements through which users interact with a system or application	Essential for delivering a user-friendly and intuitive approach to engaging with systems, thereby improving usability and productivity	• Buttons and icons in a mobile app that users tap to perform actions • Text fields for entering a username and password on a login page/ entering lecture info and tutorial info and subject info • Menus and tabs on a website that help

			users move between different sections • Error/success messages or confirmation dialogs that alert users about the success or failure of their actions • Drag-and-drop functionality in a file management system for organizing files
Data persistence	Ensure information remains stored and accessible even after an application is closed or the system restarts	• Ensure information is retained and accessible across sessions • preventing data loss and ensuring consistent application performance and user experience	• use a database(e.g.Mongo DB) • Utilize cloud services for scalable data storage • Usually back up data to prevent loss and enable recovery • Convert data into a format suitable for storage (e.g.JSON)
Authentication	System should verifies user identity	It need to protects data security, offers personalized services, and supports account management	• Require users to log-in to correct account before entering sensitive information • Use Traditional Authentication • Reset password can be used to update password
Confidentiality	System should ensure sensitive information is only accessible to authorized individuals and protected from unauthorized access or disclosure.	System is used to protect sensitive information from unauthorized access, maintain privacy, and ensure that only authorized individuals can view or handle it.	• Encrypt data I transit(HTTPS) • use secure methods of storing information in database • Limit user access to only the information necessary for their role • Mask sensitive information in user interfaces and logs to prevent unauthorized viewing