Member	Feature	Responsibilities	Patterns
Marwan	Thread Creation & Viewing	- Implement Create, Read, Update for threads. - Use Reflection to detect edits and log the actions. - Create MongoDB schema for threads and set up REST APIs - Handle thread editing, with reflected action types.	Any Pattern
Saleh	Commenting System	- Implement Create, Read for comments - MongoDB model for nested comments (Composite Pattern)	Composite design pattern for comments
Ehab	Commenting System	- Implement comment Update and Delete - Use reflection for edit tracking - Help Youssef with nesting logic	Command pattern for banning, comment, removal.
Omar	Voting System	- Implement voting on threads and comments (upvotes/downvotes). - Use Reflection to detect vote actions (upvote/downvote). - Send formatted notification messages (e.g., “Your post got 3 upvotes”) via RabbitMQ. - Add API endpoints	Any Pattern
Zeina	Reporting & Moderation	- Implement the reporting system for threads/comments. - Moderators can delete any thread or comment, while users can only delete their own. - Set up API endpoints	Any Pattern
All Members	**Shared Setup & API Communication**	**- All members work with the same base repository and Docker setup.** **- Follow the agreed data model structure (MongoDB, REST API standards).** **- Use RabbitMQ to define a common interface for APIs.**	