

Podsumowanie

Na podstawie dostarczonego kodu podsumowuję kluczowe decyzje architektoniczne i programmingowe, które podjąłem przy budowie tego task management systemu:

- Środowisko developerskie:
 - Backend (ASP.NET Core): Microsoft Visual Studio 2022 z włączonymi analizatorami Roslyn dla jakości kodu
 - Frontend (Angular): Visual Studio Code z ESLint i Prettier dla spójności stylistycznej
- **Front-end ↔ Back-end Separation**
Jasny podział na frontend (Angular) i backend (ASP.NET Core), komunikacja przez RESTful API, strong typing dzięki współdzielonym modelom między layerami.
- **Component-Based Architecture (Frontend)**
Modularne Components do konkretnych feature'ów:
 - **TaskAssignmentComponent** – główna logika przypisywania zadań
 - **AssignedTasksComponent** – widok już przydzielonych zadań
 - **AvailableTasksComponent** – zarządzanie unassigned tasks
 - **UserSelectorComponent** – wybór usera
Business logic i API calls wydzieliłem do Services layer.
- **Data Model Design**
 - **Task Model:** baza (TaskBase) z common props, a potem Specialized types (ImplementationTask, DeploymentTask, MaintenanceTask) z field constraints (string length, difficulty 1–5) i status tracking (ToDo/Done).
 - **User Model:** lekki model z Id, NameAndSurname, UserType; assignment rules w zależności od typu (Programmer dostaje tylko Implementation).
- **Programming Decisions**
 - **Pagination:** server-side pagination (PaginatedResponse<T>) z configurable page size i max cap, żeby nie overloadować frontendu.
 - **Task Assignment Logic:** walidacja requestu wg user type, prevention of duplicate assignments, limit max 10 tasks per request.

- **State Management:** tracking unsaved changes + guard przed accidental navigation, real-time status updates po zmianie statusu.
- **UI/UX**
 - Użycie Angular Material dla konsystentnych UI components i responsive layoutów.
 - Form controls z built-in validation, snackbar notifications dla feedback'u, user-friendly error messages.
- **Security & Validation**
 - Input validation: required fields, string lengths, difficulty range.
 - Business-rule enforcement: procentowe limity dla hard (4–5) i easy (1–2) tasks.
 - Prevent duplicate assignment: jedno taskId → jeden user.

Te decyzje zapewniają maintainability, scalability i dobrą UX, jednocześnie gwarantując data integrity oraz ściśle przestrzeganie business rules.