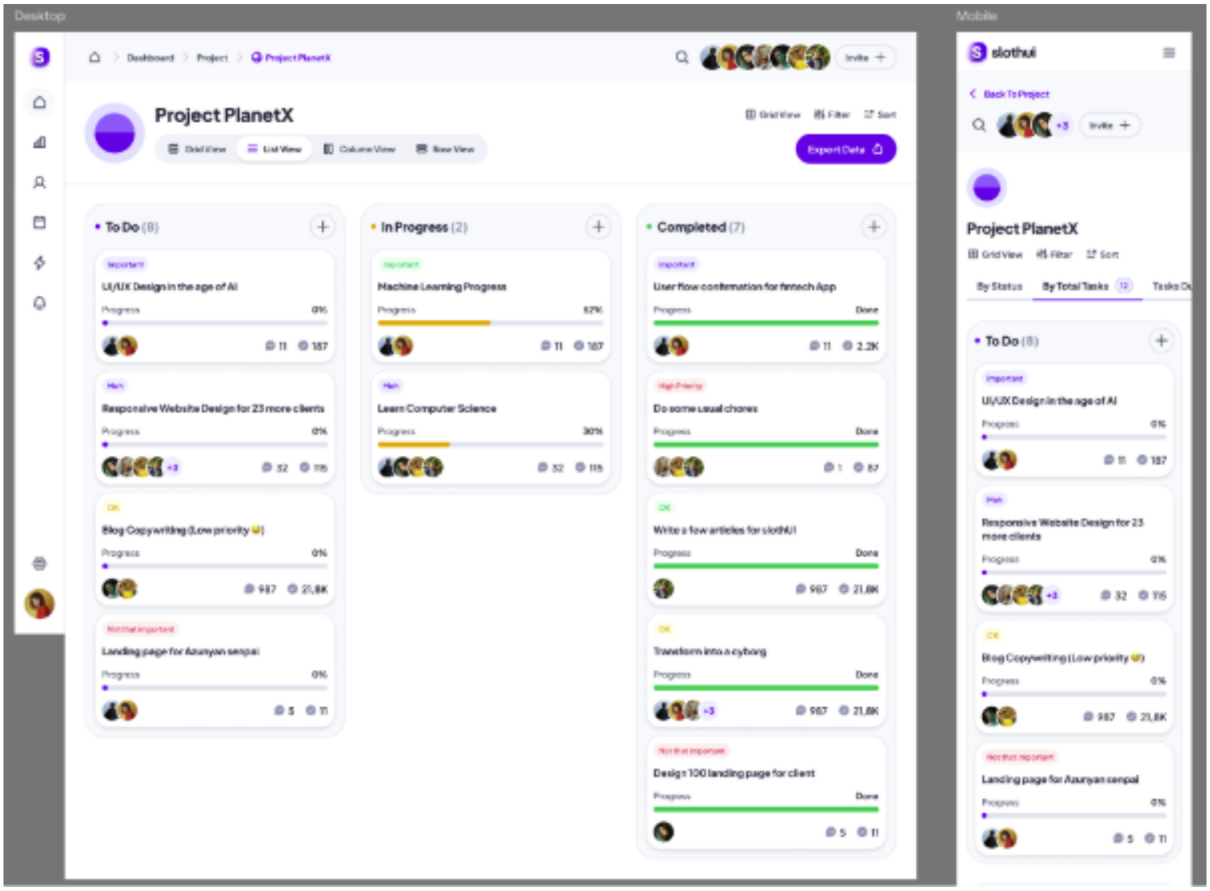
Zadatak: “Kanban API – Spring Boot back-end”

Front-end ekipe već rade React/Angular Kanban board (To Do · In Progress · Done). Treba nam pouzdan back-end u Javi i Spring Bootu koji ima sve što moderni servis mora nuditi: REST CRUD, real-time notifikacije, Swagger, itd.

Screenshot ekrana:



Cilj

Implementirati potpuno funkcionalan Kanban servis koji:

- **Persistira** zadatke u relacijskoj bazi.
- Nudi **REST CRUD** + paginaciju, filtriranje i validaciju.
- Izlaže **WebSocket/STOMP** kanal za push-obavijesti (kreiranje, update, delete).
- Automatski generira **OpenAPI 3 / Swagger UI**.
- Ima **unit i integracijske** testove s ≥ 80 % coverage-om.
- Dolazi s dockeriziranim okruženjem (app + DB).

1. Funkcionalni zahtjevi

Endpoint	Akcija	Napomene
GET /api/tasks	Listaj zadatke	Query parametri: status, page, size, sort (multi-field).
GET /api/tasks/{id}	Zadaci detalj	404 ako ne postoji.

POST /api/tasks	Kreiraj	Body: title*, description, status (TO_DO/IN_PROGRESS/DONE), priority (LOW/MED/HIGH).
PUT /api/tasks/{id}	Potpuni update	**Optimistic locking** (@Version).
PATCH /api/tasks/{id}	Djelomični update	JSON Merge Patch (application/merge-patch+json).
DELETE /api/tasks/{id}	Briši	204 No Content.

Bonus: Svaki mutacijski endpoint emitira WebSocket događaj (/topic/tasks).

Baza podataka

- **PostgreSQL ili MySQL** (Docker Compose).

2. Nefunkcionalni zahtjevi

Zahtjev	**Detalji**
OpenAPI 3	Korištenjem **springdoc-openapi**; Swagger UI dostupan na /swagger-ui.html.
Sigurnost	Simple stateless JWT auth: /api/** zaštićen; /swagger-ui.* i /v3/api-docs/** javni.
Validacija	Bean Validation (@NotBlank, @Size, @EnumValidator); globalni @ControllerAdvice za 4xx/5xx JSON-error payload.
Testovi	- *Unit*: Service i Mapper sloj (JUnit 5 + Mockito). - *Integracija*: @SpringBootTest + Testcontainers (PostgreSQL). - *WebSocket*: StompClient test koji potvrđuje broadcast nakon POST.
Docker	Dockerfile (multi-stage) + docker-compose.yml (app + db).
Observability	Spring Boot Actuator (/actuator/health, /prometheus).
Performance	Odgovor na GET /api/tasks?page=0&size=50 ≤ 150 ms na lokalnom laptopu.

3. Arhitektura & tehnički stack

Sloj	**Tehnologije**
REST	Spring Boot, Spring HATEOAS (linkovi su + bonus), springdoc-openapi
Persistencija	Spring Data JPA, Hibernate, PostgreSQL, Flyway/Liquibase migracije
Security	Spring Security (JWT filter, BCrypt)
Real-time	Spring WebSocket + STOMP + SockJS fallback
Build	Gradle/Maven, Docker BuildKit
Tests	JUnit 5, Mockito

4. Dodatni (bonus) bodovi

Tema	**Što donosi bodove**
GraphQL	/graphql endpoint paralelan s REST-om (Spring GraphQL).

Caching	Za npr. GET list/detalj.
CI pipeline	GitHub Actions: **mvn clean verify** → test check ≥ 80 % → build Docker image.
Rate limiting	Bucket4j filter (100 req/min po IP).
Helm chart	Deploy na public dostupnu VM

5. Predaja & evaluacija

- **Repo** na GitHub/GitLab – javni ili privatan (share invite).
- README s: opis izbora tehnologija i upute za pokretanje.