

# Työajanseuranta – Go (Gin + GORM) – Projektisuunnitelma

---

## 1. Tavoite ja laajuus

- Rakentaa kevyt työajanseurantajärjestelmä Go:lla.
- Käyttäjäroolit
  - Työntekijä: näkee omat kirjaukset ja luo/muokkaa/poistaa omiaan.
  - Esimies: näkee kaikkien kirjaukset, suodattaa/raportoi.
- REST-rajapinta (JSON), pysyvä tietokanta, yksikkö-/integraatiotestit, dokumentaatio ja lyhyt esitys.

## 2. Arkkitehtuuri (kerrokset)

- API-kerros (Gin)
  - HTTP-reitit ja request/response DTO:t
- Sovelluslogiikka (Services)
  - Käyttötapaukset: kirjauksen luonti, haku, muokkaus, poistot
  - RBAC-tarkistukset (roolipohjainen valtuutus)
- Tietopääsy (Repository, GORM)
  - CRUD-toiminnot, transaktiot
  - Migraatiot ja seed-data
- Yhteiset
  - Auth (JWT), salasanojen hashays (bcrypt)
  - Konfigurointi (env/flags), lokitus, virheenkäsittely

## 3. Teknologiat

- Go 1.22+, Gin ([github.com/gin-gonic/gin](https://github.com/gin-gonic/gin)), GORM ([gorm.io/gorm](https://gorm.io/gorm)), SQLite.
- JSON Web Token (JWT) -autentikointi;
- Makefile kehitystyön helpottamiseksi.

## 4. Koodin hakemistorakenne

- cmd/server
  - main.go – käynnistys, di-wiring
- internal/api
  - handlers/ – reitit ja controllerit
  - middleware/ – auth, request-id, logging
  - dto/ – request/response-mallit
- internal/domain
  - models/ – User, TimeEntry, (Project)
  - services/ – liiketoimintasäännöt
- internal/repository
  - user\_repo.go, timeentry\_repo.go – GORM-toteutukset
- internal/auth
  - jwt.go – tokenit ja validointi
  - password.go – bcrypt
- internal/config
  - config.go – .env ja oletukset
- internal/platform
  - db/ – DB-yhteys ja migraatiot
  - logger/ – yhteinen lokitus
- migrations
  - SQL/GORM automigrate; siemenet (admin-käyttäjä)
- docs
  - openapi.yaml (generoitu), arkkitehtuurikaavio, README
- scripts
  - make targets: run, test, lint, migrate

## 5. Domain-malli

- User
  - id, name, email, role ∈ {EMPLOYEE, MANAGER}, password\_hash, created\_at
- TimeEntry
  - id, user\_id (FK), date, start, end, duration\_min, description, created\_at, updated\_at
- Mahdollinen Project-entity myöhemmin (scope/raportointi).

## 6. REST-rajapinta (v1)

- Auth
  - POST /api/v1/auth/login → JWT
  - GET /api/v1/auth/me → oma profiili
- Users
  - GET /api/v1/users (MANAGER)
  - POST /api/v1/users (bootstrap/admin)
- Time entries – työntekijä
  - GET /api/v1/me/time-entries?from=&to=
  - POST /api/v1/me/time-entries
  - PATCH /api/v1/me/time-entries/{id}
  - DELETE /api/v1/me/time-entries/{id}
- Time entries – esimies
  - GET /api/v1/time-entries?user\_id=&from=&to=
  - Aggregaatit: GET /api/v1/reports/summary?from=&to=
- Virheformaatti: {"error": "...", "code": "..."}; validointivirheet listana.

## 7. Tietokanta

- SQLite
- Lokitus (request-id), audit trail (kuka muokkasi).
- Yksikkötestit (services), integraatiotestit (repos + in-memory/containertesti).

## 10. Toimitettavat (kurssivaatimukset)

- HTTP-API (JSON REST) ja toimiva demo.

- Lyhyt esitys teknologioista (Go, Gin, GORM) ja toteutuksesta.
- Dokumentaatio: README (asennus/ajo), OpenAPI, arkkitehtuuri-kuva.
- Testit ja esimerkkikutsut (curl/Postman).

## **11. Aikataulu (viikkotasolla)**

- vko 1: skeleton, DB-yhteys, domain-mallit, migraatiot
- vko 2: auth + peruskirjaukset (me/\*), testit
- vko 3: esimiesnäkymät (listaus/raportit), laadun parannus
- vko 4: viimeistely, dokumentaatio, esitys ja demo

## **12. Laajennukset (valinnaiset)**

- Projektit ja kustannuspaikat, CSV/Excel-vienti, ylityösäännöt.