**🗓️ WEEK 1 — Opzet & Database**

🎯 **Doel:** De volledige basisstructuur van je Node.js API opzetten en database laten werken.

**✅ Wat je moet doen:**

* **Project opstarten**
  + Nieuwe map aanmaken en npm init -y
  + Packages installeren:  
    npm install express body-parser cors knex sqlite3
  + .gitignore aanmaken → node\_modules/ + \*.sqlite3 erin zetten
  + GitHub repo aanmaken en koppelen
* **Projectstructuur opbouwen**
  + Mappen: db, models, routes, controllers, middleware
  + Bestanden: index.js, README.md, db/knex.js, db/database.sqlite3
* **Database configureren**
  + db/knex.js maken met connectie naar SQLite
  + Tabellen aanmaken met Knex (bijv. users, posts)
  + Database testen met DB Browser for SQLite
* **Express server**
  + Server laten draaien op poort 3000
  + Root endpoint / maken dat een overzicht geeft van alle routes
* **Eerste commits doen (minstens 5)**
  + Setup, structuur, .gitignore, database, server  
    *(veel kleine commits = meer punten!)*

**🗓️ WEEK 2 — CRUD + Validatie**

🎯 **Doel:** Alle API-functionaliteit werkend maken met Knex en Express.

**✅ Wat je moet doen:**

* **CRUD-functionaliteit**
  + controllers/userController.js → GET, POST, PUT, DELETE
  + controllers/postController.js idem
  + routes/userRoutes.js & routes/postRoutes.js
  + Routes koppelen in index.js
  + Testen met Postman
* **Models (optioneel maar proper)**
  + models/userModel.js en models/postModel.js
  + Database-logica daarin zetten, controllers proper houden
* **Validatie toevoegen**
  + middleware/validation.js
  + Check verplichte velden, correcte types (bv. geen cijfers in naam)
  + Middleware toepassen in routes
* **Extra API-functies**
  + ?search=... → zoeken op naam
  + ?limit=...&offset=... → paginatie
  + Errorhandling met duidelijke JSON-berichten
* **Veel commits (minstens 8)**
  + CRUD users, CRUD posts, models, validatie, search, paginatie

**🗓️ WEEK 3 — Extra’s, testen en documentatie**

🎯 **Doel:** Je project afwerken, testen en documenteren (en extra features toevoegen voor bonuspunten).

**✅ Wat je moet doen:**

* **Extra features (voor 20/20)**
  + Sorteren (?sort=age&order=desc)
  + Eenvoudige authenticatie (API key)
  + Geavanceerde validatie (e-mail regex, dataconsistentie)
  + Eigen extra endpoint (bv. statistieken)
* **Testen**
  + Test alle endpoints met Postman
  + Controleer foutmeldingen (400, 404, 500)
  + Noteer resultaten of neem screenshots
* **Documentatie**
  + README.md goed uitschrijven:
    - Hoe te starten (npm install, node index.js)
    - Overzicht van endpoints
    - Gebruikte bronnen
    - Uitleg over je extra features
    - Vermelding: *“Project gemaakt door Laila Nounouh – BeautyConnect API”*
* **Laatste commits (minstens 5)**
  + Testen, documentatie, verbeteringen
* **Presentatie en inlevering**
  + Video opnemen waarin je toont:
    - De structuur
    - Een paar endpoints live testen
    - Uitleg over je extra feature(s)