

# 12-Week Backend Engineering Plan

**Project Name:** *ScoutConnect – Smart Scouting Platform for Talent Discovery & Collaboration*

---



## PHASE 1 – Foundation & Setup (Weeks 1–4)

---

### Week 1: Project Kickoff & Environment Setup

- Create GitHub repo, initialize project
  - Set up virtual environment and dependencies (FastAPI or Django)
  - Install PostgreSQL and pgAdmin
  - Design folder structure (`/models`, `/routes`, `/services`, `/utils`, etc.)
  - Set up Docker (Dockerfile, docker-compose.yml)
  - Build `.env` config and add `.gitignore`
  - Start `README.md` with project overview
- 

### Week 2: Database Design + User Auth

- Design and create your database schema (ERD)
  - `Users`, `Players`, `Evaluations`, `Stats`, `Templates`, `Comments`, `Watchlist`, `Tags`
- Implement user registration/login/logout with JWT
- Add user roles: Coach, Scout, Admin

- Secure routes with role-based access
- 

### Week 3: Core API – Player Profiles & Evaluations

- Build `/players` CRUD routes
  - Build `/evaluations`:
    - Create + retrieve evaluations
    - Attach to player + evaluator
  - Link evaluation criteria by sport/position
  - Test routes with Postman
- 

### Week 4: Multi-Sport Criteria System

- Build `/criteria-generator`:
    - Input: sport, position
    - Output: custom evaluation form (speed, IQ, clutch, etc.)
  - Create backend logic for:
    - Default forms by sport
    - Custom forms saved per coach
  - Store reusable templates in DB
  - Build `/coach/templates` route
-

## PHASE 2 – Smart Logic & Collaboration (Weeks 5–8)

---

### Week 5: Scoring Engine (Universal + Sport-Specific)

- Build evaluation scoring algorithm:
    - Weighted scores for metrics
    - Clutch rating (based on tags like 4th quarter)
  - Add `/player/{id}/score` endpoint
  - Display breakdown of:
    - Overall score, clutch %, growth %, etc.
- 

### Week 6: Recruitability & Hidden Talent Engine

- Build `/recruitability` endpoint:
    - Factors: growth, injuries, clutch, effort, feedback
  - Build `/hidden-gems` endpoint:
    - Query players with high growth & effort, low visibility
  - Add filters (age, level, sport)
- 

### Week 7: Player Comparison + Watchlists

- Create `/compare` endpoint:
  - Input: 2+ player IDs
  - Output: side-by-side metrics

- Create `/watchlist`:
    - Add/remove players per user
    - Route to view all watchlisted players
- 

## Week 8: Collaboration & Comments

- Build `/comments`:
    - Add/view comments on players or evaluations
  - Build `/share-eval`:
    - Share player evaluations with other scouts/coaches
  - Add tagging system (`#underrated`, `#clutch`, etc.)
  - Build `/notifications` (optional stretch)
- 

## PHASE 3 – Data Ingestion, Testing & Final Polish (Weeks 9–12)

---

### Week 9: Real-Time Stat Integration

- Build `/stats/upload`:
  - Upload CSV or JSON with player stats
  - Update player records & evaluation scores
- Track trends like:
  - Last 5 games

- Game-by-game performance dips/gains
  - Link to clutch rating updates
- 

## Week 10: Injury Risk & Recovery Tracker

- Add health profile to players
  - Input missed games, injuries, recovery notes
  - Predict future injury risk using:
    - Game load, stat drops, injury history
  - Build `/injury-report` endpoint
- 

## Week 11: Testing, Validation & Error Handling

- Add unit and integration tests (pytest or Django TestCase)
  - Validate inputs with Pydantic/Django Forms
  - Handle auth errors, 404s, bad input, duplicate users
  - Finalize complete API docs with Swagger/OpenAPI
- 

## Week 12: Final Polish & Demo Prep

- Final code cleanup + comments
- Write documentation (`README.md`, setup guide)
- Record a project demo or presentation
- (Optional) Deploy locally with Docker or use Render/Railway

- Present to bootcamp or mentor for feedback

---

## Deliverables by End of Project

- Fully functional backend app with auth, API, scoring, logic, collaboration
- SQL database with players, stats, evaluations, and scout feedback
- Dockerized local environment with [README.md](#)
- Postman collection for API testing
- (Optional) Live deployment link for demo