Product Requirements Document (PRD)

# TutorTrack – Smart Scheduling & Income Tracker for Tutors

## 1. Overview

Product Name: TutorTrack  
  
Product Goal:  
Create a clean, intuitive SaaS platform that helps tutors manage their class schedules and track real-time income. The app replaces clunky spreadsheets and disjointed tools by combining scheduling, payments, and analytics in one place.  
  
Target Users:  
- Freelance language tutors (initial focus)  
- Solo educators (online or in-person)  
- Part-time or full-time instructors with hourly/session-based income

## 2. Goals & Success Metrics

### Goals

- Provide a unified tool to manage both class schedules and income tracking  
- Help tutors easily monitor their financial and teaching activity  
- Design an easy-to-use, clean, responsive interface

### Success Metrics

- 80%+ of beta testers fully switch from spreadsheets or Google Calendar  
- MVP users report at least 2 hours/week saved on admin tasks  
- Net Promoter Score (NPS) of 8+ during beta

## 3. Core Features (MVP)

1. Dashboard:  
- Daily, weekly, and monthly earnings  
- Upcoming scheduled classes  
- Unpaid vs. paid session totals  
- Visual analytics (basic graphs/charts)  
  
2. Class Scheduling:  
- Add/edit single or recurring classes  
- Assign to student profile  
- Status indicators (Upcoming, Completed, Cancelled)  
  
3. Income Tracking:  
- Manually log payments per session  
- Paid/Unpaid toggle  
- Multiple currency support (USD, EUR in MVP)  
  
4. Student Profiles:  
- Basic student info (name, contact, notes)  
- Total sessions and payment stats  
- Filter/tag students (e.g., “Trial,” “Group,” “Owes Payment”)  
  
5. Basic Analytics:  
- Most profitable days  
- Payment completion rate  
- Missed vs. completed sessions

## 4. User Stories

- As a tutor, I want to track what I earned this week so I can plan my finances.  
- As a teacher, I want a visual overview of my week’s lessons so I stay organized.  
- As a freelancer, I want to see which students owe me money so I can follow up.

## 5. Technical Stack

Frontend:  
- Framework: Next.js (React-based, full-stack capabilities)  
- Design System: ShadCN UI – for modern, accessible UI components  
- Styling: Tailwind CSS (used by ShadCN)  
  
Backend & Database:  
- Database: Supabase (PostgreSQL + real-time sync)  
- Auth: Supabase Auth (email login)  
  
Hosting & Deployment:  
- Platform: Vercel (ideal for Next.js + serverless)  
- CI/CD: Vercel GitHub integration

## 6. Constraints & Assumptions

- MVP does not include video calling, messaging, or student-facing login  
- All features are tutor-facing only  
- Focus on desktop and mobile web; no native app in MVP  
- Minimal data validation at MVP level (no tax/invoice compliance yet)

## 7. Post-MVP Ideas

- Native mobile app (React Native)  
- Stripe integration for collecting payments  
- Invoice generation & downloadable reports  
- AI-powered scheduling suggestions  
- Group class functionality  
- Google Calendar sync