WinCallem – Week 1 Onboarding Guide (For Adrian & Aaron)

Generated: September 02, 2025

# About This Document

This is a step-by-step, beginner-friendly guide covering exactly what you need to do this week and how it fits into the bigger 6-week MVP plan. If you get stuck for more than ~30 minutes:  
1. Ask ChatGPT using the included docs.  
2. Ping Tim with your notes.

# The Vision in Plain English

WinCallem is a website/app where anyone can build their own sports betting models without writing code. Users will:  
- See real sportsbook odds (with affiliate “Bet” buttons)  
- Configure a simple model (pick inputs like team strength, pitcher stats)  
- Run the model (in the background)  
- See results/metrics  
- Get alerts when there’s an edge  
- Export results (CSV/Parquet)

# MVP in 6 Weeks

By the end of 6 weeks, a user can: log in, view odds, build a simple model, run it, see results, get alerts, export data, and optionally click a sportsbook affiliate link.  
  
High-level calendar:  
• Week 1: Setup + docs + small contributions  
• Week 2: Auth (Auth.js) + Stripe subscriptions  
• Week 3: Odds ingestion → DuckDB/Parquet; basic odds table  
• Week 4: Model Builder v0 (configure → run Celery job → show metrics)  
• Week 5: Alerts (edge > X%) + exports (CSV/Parquet)  
• Week 6: Testing + polish + MVP demo

# Roles & Weekly Rhythm

• Tim (Project Lead): architecture, core features, roadmap, code review  
• Adrian & Aaron (Collaborators): setup, documentation, small UI/data tasks, testing  
• Weekly meetings: Sundays — demo progress, review Setup Logs, assign next week’s goals

# Week 1 Checklist (with Time Budget)

Goal: run the project locally, document your steps, and make one small contribution.  
Estimated time: ~10 hours each.

A. Environment Setup (2–3h)  
- Install Docker Desktop  
- Install Node.js (v18+)  
- Install GitHub Desktop (or Git + VS Code)  
- Create a GitHub account (if needed)

B. Run the Backend (2–3h)  
- Copy the API env file  
- Run docker compose up --build  
- Confirm API docs at http://localhost:8000/docs

C. Run the Frontend (1–2h)  
- Copy the web env file  
- npm install  
- npm run dev  
- Confirm http://localhost:3000 works

D. Test Dashboard (0.5–1h)  
- Ping API → { 'status': 'ok' }  
- Fetch Odds → sample JSON  
- Run Model Job → Check Result → toy metrics

E. Document Setup (1–2h)  
- Create /docs/Setup\_Log\_[Name].md  
- Record exact steps, errors, fixes

F. Make One Contribution (0.5–1.5h)  
Choose one:  
- Update Step Zero guide with clearer beginner notes  
- OR tweak a small UI detail (button spacing, color, heading text)

# Install the Tools (Windows & Mac)

Windows:  
- Docker Desktop → install, then reboot.  
- Node.js v18+ → confirm with node -v.  
- GitHub Desktop (or VS Code + Git).  
  
Mac:  
- Docker Desktop for Mac (Apple Silicon is fine).  
- Node.js v18+ → confirm with node -v.  
- GitHub Desktop (or VS Code + Git).

# Get the Project Running

Backend (from project root):  
Windows:  
 copy apps\api\.env.example apps\api\.env  
 docker compose up --build  
Mac:  
 cp apps/api/.env.example apps/api/.env  
 docker compose up --build  
  
Frontend (new terminal):  
cd apps/web  
Windows: copy .env.example .env.local  
Mac: cp .env.example .env.local  
npm install  
npm run dev  
  
Check: open http://localhost:3000

# Test the Dashboard

- Ping API → { 'status': 'ok' }  
- Fetch Odds → mocked JSON  
- Run Model Job → Check Result → toy metrics

# Document Your Setup (Setup Log)

Log everything in /docs/Setup\_Log\_[YourName].md:  
- Steps in order  
- Errors and fixes  
- Notes for next time

# Make One Contribution

Option A: improve Step Zero docs.  
Option B: tweak one UI detail.  
  
Commit and push with GitHub Desktop.

# Using ChatGPT the Right Way

- Upload the included PDFs + Markdown if you need help.  
- Good prompts:  
 “I ran docker compose up and got this error (paste). What should I try?”  
 “Explain in plain English what npm install is doing.”

# Troubleshooting (Common Issues)

Docker not running → Open Docker Desktop.  
Port in use (8000/3000) → close apps or run frontend on a different port.  
Node/npm errors → delete node\_modules and package-lock.json, reinstall.  
Mac Silicon quirks → update Docker, rebuild with docker compose build --no-cache.

# Repo Hygiene

- Don’t commit .env files.  
- Don’t commit large raw data.  
- Use small, clear commits.

# Sunday Meeting Agenda

Each of us will:  
- Demo the dashboard running.  
- Walk through our Setup Logs.  
- Show our contribution (doc or UI tweak).  
- Review next week’s plan (Auth + Stripe).

# Glossary

Frontend: website UI  
Backend: server logic  
Celery + Redis: worker + queue for jobs  
Docker: container environment  
Postgres: main database  
DuckDB: analytics database  
PWA: installable web app  
Affiliate link: sportsbook partner link

# Appendix – Commands Cheat Sheet

Windows:  
dir  
copy apps\api\.env.example apps\api\.env  
cd apps\web  
copy .env.example .env.local  
npm install  
npm run dev  
docker compose up --build  
Ctrl + C (stop)  
  
Mac:  
ls -la  
cp apps/api/.env.example apps/api/.env  
cd apps/web  
cp .env.example .env.local  
npm install  
npm run dev  
docker compose up --build  
Ctrl + C (stop)