

ft_transcendence - Parallel Work Plan (First 25%)

2-Week Checklist (May 25 - June 8)

Person 1 - Frontend UI + SPA

May 25-26 (Sat-Sun): Setup Vite + TypeScript SPA, project layout

May 27-28 (Mon-Tue): Create Home and Register pages (UI only)

May 29 (Wed): Implement form validation (alias, match)

May 30-31 (Thu-Fri): Add SPA routing (Back/Forward support)

June 1 (Sat): Build dummy tournament page layout

June 2-3 (Sun-Mon): Add fake fetch logic with hardcoded JSON

June 4 (Tue): Test all pages in Firefox

June 5-6 (Wed-Thu): Add loading/error states

June 7-8 (Fri-Sat): Final UI cleanup and prep for API integration

Person 2 - Pong Game Developer

May 25-26 (Sat-Sun): Setup HTML Canvas or JS game framework

May 27-28 (Mon-Tue): Paddle and ball movement, bouncing logic

May 29 (Wed): Add scoring and win condition

May 30 (Thu): Implement restart/reset option

May 31-June 1 (Fri-Sat): Code cleanup + responsive sizing

June 2 (Sun): Make game reusable (export function/class)

June 3-4 (Mon-Tue): Embed in a basic web page and test

June 5-6 (Wed-Thu): Add comments, prepare for integration

June 7-8 (Fri-Sat): Final game test and freeze version

Person 3 - Backend Developer (PHP)

May 25-26 (Sat-Sun): Setup PHP project + routing file

May 27-28 (Mon-Tue): Create /register endpoint, store alias

May 29-30 (Wed-Thu): Create /tournament endpoint with dummy logic

May 31 (Fri): Validate inputs, sanitize data

June 1 (Sat): Use SQLite or file-based storage

ft_transcendence - Parallel Work Plan (First 25%)

June 2 (Sun): Hash passwords (if used), test security

June 3-4 (Mon-Tue): Test with Postman or curl + bug fixes

June 5-6 (Wed-Thu): Add doc comments, freeze version

June 7-8 (Fri-Sat): Ready for frontend connection

Person 4 - DevOps + Docker + HTTPS

May 25-26 (Sat-Sun): Setup Dockerfile for frontend and backend

May 27 (Mon): Create docker-compose.yml

May 28-29 (Tue-Wed): Configure .env support and secrets

May 30-31 (Thu-Fri): Generate self-signed SSL cert and configure HTTPS

June 1 (Sat): Test both containers together

June 2 (Sun): Add volume or SQLite persistence (if needed)

June 3-4 (Mon-Tue): File structure cleanup, add readme/setup notes

June 5-6 (Wed-Thu): Final integration test with dummy frontend/backend

June 7-8 (Fri-Sat): Assist with full system integration

Shared Milestones

June 1 (Sat): Mid-point check-in: everyone demos basic working version

June 8 (Sat): Final code freeze: all independent parts are ready

June 9+ (Sun onward): Start integration week (Week 3)