Detailed Teaching Schedule - Chef Portal Project

Week 1: Interface Design

Monday - Breadboarding & Setup

Morning (90 min sync)

- Review breadboard draft (30 min)
- Collaboratively refine breadboard (45 min)
- Outline Week 1 deliverables (15 min)

Team owns (afternoon async):

- Install Claude Code
- Build landing page for chef network announcement
  - Learn deployment basics
  - Work backwards from the vision
  - What would you tell chefs about this new platform?

Takuma delivers:

- Development environment setup
- System prompts for interface design
- Deploy landing page to Render

Tuesday - Fat Marker Sketches

Morning (60 min sync)

- Fat marker sketching session together
- Focus on core flows:
  - Job entry
  - Payment collection (both paths)
  - Confirmation

Team owns (afternoon async):

- Build your own first interface mocks using Claude Code
- Convert sketches to Rails ERB views
- Push to repository for review
- Experiment and make design decisions independently

Takuma provides:

- System prompts for Rails interface development
- Quick feedback on pushed code
- Unblocking support as needed

Wednesday - Interface Refinement

Morning (60 min sync)

- Review interface mocks together
- Discuss user experience decisions
- Identify gaps and edge cases

Team owns (afternoon async):

- Iterate on interfaces based on feedback
- Complete all 5 core screens
- Test mobile responsiveness

Takuma provides:

- Design principle guidance
- Technical feasibility checks
- Mobile testing support

Thursday - Business Logic Introduction

Morning (60 min sync)

- Takuma explains Rails MVC pattern
- Demo how data flows through views
- Show how to hard-code sample data

Team owns (afternoon async):

- Connect all screens into clickable demo
- Add sample data to controllers
- Test flow on mobile devices

Takuma delivers:

- Controller templates with dummy data
- Routes configuration
- Navigation helpers

Friday - Specification & Setup

Takuma delivers (no action needed from team):

- Write detailed specs for Week 2 wiring:
  - Square API integration points
  - Database schema
  - Authentication flow
- Set up development kitchen on Render
- Prepare system prompts for wiring
- Create wiring checklist

Team is free to:

- Continue with regular work
- Optionally refine vision further
- Note: Thursday's clickable demo is what gets wired in Week 2

Week 2: Wiring & Automation

Monday - Integration Planning

Morning (60 min sync)

- Review wiring checklist together
- Takuma demonstrates Square webhook setup
- Pair program first integration

Team observes, Takuma implements:

- Square API authentication
- First webhook handler
- Database connections

Afternoon (async)

- Continue Square integration
- Test payment flows

Tuesday - Core Logic

Morning (60 min sync)

- Takuma wires payment confirmation logic
- Team observes and asks questions
- Test together in Square sandbox

Takuma implements (team observes):

- All payment flows
- Invoice creation logic
- Earnings calculations

Team can help with:

- Testing different scenarios
- Identifying edge cases
- UI polish

Wednesday - Deployment & Rollout

Morning (60 min sync)

- Takuma deploys to production Render
- Team creates rollout plan
- Together identify pilot chef

Team owns:

- Writing rollout communication
- Choosing pilot chef
- Planning training approach

Takuma delivers:

- Production deployment
- Bug fixes
- Documentation of deployment process

Thursday - Spreadsheet Automation

Morning (2 hours sync - Lee must attend)

- Together design new spreadsheet structure
- Takuma builds Zapier automation live
- Lee tests with real payment data
- Team learns automation principles

Takuma delivers:

- Complete Zapier setup
- New spreadsheet template
- Automation documentation

Team owns:

- Deciding spreadsheet columns
- Testing edge cases
- Approving final workflow

Afternoon (async)

- Document spreadsheet logic
- Create operation manual
- Plan Friday pilot

Friday - Pilot Launch

Team owns (Takuma on standby):

- Guide pilot chef through first payment
- Monitor data flow to spreadsheet
- Gather and document feedback
- Celebrate! 🎉

Takuma provides:

- Immediate bug fixes if needed
- Data flow monitoring
- Emergency support

Teaching Philosophy

Week 1 Focus

- **They drive**: Interface design decisions
- **Takuma guides**: Design principles, user experience
- **Outcome**: They own the product vision

Week 2 Focus

- **Takuma drives**: Technical wiring, integrations
- **They observe**: Learn by watching and asking
- **Outcome**: They understand how it works

Progressive Complexity

1. Start with static HTML (landing page)
2. Move to Rails ERB views (server-rendered)
3. Add Rails controllers and models
4. Wire real data (Square API)
5. Automate operations (spreadsheets)

System Prompts Strategy

- Week 1 prompts: Interface-only, no backend
- Week 2 prompts: Integration and wiring
- Post-project: Full-stack development

Success Metrics

Week 1

- [ ] All 6 screens designed and clickable
- [ ] Mobile-responsive interface
- [ ] Clear user flow documented
- [ ] Team can explain every design decision

Week 2

- [ ] Square payments flow to portal
- [ ] Data appears in Lee's queue
- [ ] One real payment processed
- [ ] Team can modify system independently

Risk Mitigation

If they struggle with Claude Code:

- Provide more direct examples
- Pair program key sections
- Focus on understanding over doing

If Square integration is complex:

- Start with manual confirmation
- Add webhook automation later
- Keep MVP scope tight

If spreadsheet automation fails:

- Manual CSV export as backup
- Gradual automation rollout
- Keep existing process in parallel

Post-Project Support

Week 3+:

- Weekly office hours (30 min)
- Async support for bugs
- Guide on adding features:
  - Team splits
  - Deposit tracking
  - Commission tiers
  - Historical reporting