DressUp MVP: 6-Week Integrated Learning Plan

Designed for: Shubham (INTJ) | **Time Commitment:** 30-32 hours/week | **Goal:** Build working DressUp MVP while mastering AI system architecture

---

Plan Philosophy & Structure

This plan is designed around your INTJ strengths: systems thinking, pattern recognition, and conceptual mastery. Each week connects theory to practice through DressUp, avoiding the trap of abstract learning. You'll use AI models (Claude, ChatGPT) as thinking partners throughout.

Weekly Structure:

- **Core Focus:** One major skill area
- **DressUp Application:** Concrete implementation
- **Learning Hours:** 30-32/week (20 building, 8 learning, 4 documenting)
- **Deliverables:** Always tangible outputs for portfolio

Week 1: Foundation Architecture & Prompt Engineering

Theme: "Design the System Brain"

Core Concepts

- Prompt engineering fundamentals: role, task, context, constraints (RTCC framework)
- System architecture basics: entities, relationships, data flow
- Mental models for AI-powered products
- Decision framework thinking: mapping user inputs to system outputs

DressUp Tasks

1. Core System Architecture (8 hours)

- Define DressUp's data model:
  - **Entities:** User, WardrobeItem, Outfit, Event, WeatherContext, Preference
  - **Relationships:** User owns Items, Items form Outfits, Events trigger recommendations
  - **State flow:** From user input → context gathering → recommendation logic → output
- Create visual architecture diagram showing how data flows through DressUp
- Document decision points where AI logic is needed vs. rule-based logic

2. Prompt Library Development (10 hours)

- Build 15-20 production prompts for DressUp features:
  - **Wardrobe item categorization prompt:** "You are a fashion data specialist. Given this item description [ITEM], categorize it by: type (top/bottom/footwear/accessory), style (casual/formal/sporty), season, color palette, and formality level (1-10). Format response as JSON."
  - **Outfit recommendation prompt:** Role prompt for styling assistant considering weather, event, user preferences
  - **Style preference extraction prompt:** Analyze user's existing wardrobe to infer style personality
  - **Copy generation prompts:** App notifications, outfit descriptions, style tips
- Test each prompt with Claude and ChatGPT, document which works better for what
- Create prompt debugging checklist

3. Industry Research (6 hours)

- Case study analysis (2 hours each):
  - **Stitch Fix:** How they combine data science with human stylists
  - **Rent the Runway:** Subscription model and wardrobe rotation logic
  - **Thread/Cladwell:** AI styling recommendation approaches
- Extract: What makes their recommendation engines work? What are the data requirements? Where do they succeed/fail?

4. Fashion Supply Chain Understanding (4 hours)

- Research how fashion retail works: seasons, inventory turnover, assortment planning
- Understand why outfit recommendations could be valuable to retailers (reduce returns, increase basket size, optimize inventory)
- Document how DressUp creates value for both users AND potential brand partners

5. Decision Framework (4 hours)

- Map user incentives: Why would someone use DressUp daily?
- Identify friction points: What makes people stop using styling apps?
- Game theory lens: How to design feedback loops so users improve the system by using it
- Create "user decision tree" showing decision fatigue problem → DressUp solution

Week 1 Deliverables

1. **DressUp System Architecture Document** (visual diagram + written explanation)

2. **Production Prompt Library** (15-20 tested prompts with performance notes)
3. **Industry Insight Report** (3 case studies with extracted lessons)
4. **Supply Chain Value Proposition** (how DressUp fits into fashion ecosystem)
5. **Public Post #1:** "Designing an AI Fashion System: Architecture Decisions" (LinkedIn case study)

Key Learning Resources

- Study system design patterns online (focus on recommendation systems)
- Read OpenAI/Anthropic prompt engineering guides
- Analyze Stitch Fix's public tech blog for ML insights

Week 2: Data Models, APIs, & No-Code Building

Theme: "Make It Real - Build the MVP Foundation"

Core Concepts

- Data modeling: entities, attributes, relationships, keys
- API fundamentals: what they are, how they work, authentication
- No-code tools: Glide, Airtable/Google Sheets as databases
- Understanding external data sources (weather, calendar)

DressUp Tasks

1. Database Design & Setup (10 hours)

- Build DressUp data model in Google Sheets (or Airtable):
  - **Users Table:** UserID, Name, Email, StylePreferences (JSON), BodyType, Location
  - **Wardrobe Table:** ItemID, UserID, Category, SubCategory, Color, Season, FormalityLevel, ImageURL, LastWorn
  - **Outfits Table:** OutfitID, UserID, Items (array), CreatedDate, Rating, Context
  - **Events Table:** EventID, UserID, EventType, Date, Location, DressCode
  - **Feedback Table:** FeedbackID, OutfitID, Rating, Notes, Date
- Populate with sample data: Create a fictional user with 30 wardrobe items
- Document data model with clear definitions of each field

2. Weather API Integration Planning (6 hours)

- Research OpenWeather API documentation

- Understand API structure: endpoints, parameters, response formats
- Create specification document:
  - What weather data do we need? (temp, conditions, precipitation)
  - How frequently should we call it?
  - How to convert weather data into outfit recommendations (temperature thresholds, rain = need umbrella/jacket logic)
- Write pseudocode for weather-aware recommendation logic

3. MVP Feature Prioritization (4 hours)

- Apply Kano model to DressUp features:
  - **Must-haves:** Wardrobe input, basic outfit suggestion, weather consideration
  - **Performance features:** AI styling improvements, preference learning
  - **Delighters:** Social sharing, calendar integration, shopping suggestions
- Create MVP scope document: What ships in 6 weeks vs. what's Phase 2?
- Trade-off analysis: Why we're NOT building certain features yet

4. No-Code App Prototype (8 hours)

- Build functional DressUp prototype in Glide:
  - Home screen with daily outfit suggestion
  - Wardrobe management screen (add items with photos)
  - Simple recommendation logic (even if rule-based initially)
  - Basic user settings for preferences
- Connect Glide to your Google Sheets database
- Test user flow end-to-end

5. Analytics Planning (4 hours)

- Define key metrics for DressUp:
  - User engagement: Daily active users, session length
  - Feature usage: Outfits viewed, ratings given, items added
  - Recommendation quality: Acceptance rate, user feedback scores
- Design simple analytics tracking in sheets (what events to log)
- Create dashboard mockup showing metrics you want to track

Week 2 Deliverables

1. **Complete Data Model** (populated Google Sheets with sample data)
2. **API Integration Specification** (weather API usage plan)
3. **MVP Scope Document** (feature prioritization with rationale)
4. **Working Glide Prototype** (functional app with 5 core screens)
5. **Analytics Framework** (metric definitions and tracking plan)
6. **Public Post #2:** "Building DressUp: Data Architecture for AI Fashion Recommendations"

Key Learning Resources

- Glide tutorials and documentation
- OpenWeather API documentation
- Review how recommendation systems structure data (collaborative filtering basics)

Week 3: AI/ML Concepts & Recommendation Logic

Theme: "Understand the Intelligence Layer"

Core Concepts

- Supervised learning intuition: features → labels, training → prediction
- Recommendation systems: content-based vs. collaborative filtering
- Feature engineering for fashion (what makes a good "outfit feature"?)
- Model concepts: training data, validation, overfitting, bias
- Neural networks as layered decision filters (conceptual, no code)

DressUp Tasks

1. Recommendation Algorithm Design (10 hours)

- **Phase 1 - Rule-Based Logic (MVP):**
  - Define rules for outfit matching:
    - Color compatibility (complementary, analogous, neutral combinations)
    - Formality matching (don't pair sportswear with formal shoes)
    - Weather appropriateness (temperature → sleeve length, precipitation → materials)
    - Season alignment (no heavy coats in summer)
  - Implement scoring system: Each outfit gets compatibility score (0-100)
  - Write detailed pseudocode for recommendation engine

- **Phase 2 - ML Enhancement Vision (Future):**
  - Design what a learned model would look like:
    - **Features:** Item attributes (category, color, formality), user preferences, context (weather, event)
    - **Labels:** User ratings (did they wear this outfit? did they like it?)
    - **Training approach:** Start with explicit ratings, move to implicit feedback (what they actually wear)
  - Document how system would evolve from rule-based to ML-based
  - Identify what data you need to collect NOW to enable ML later

2. AI Concepts Study (8 hours)

- Deep dive on recommendation systems:
  - Content-based: Recommend items similar to what user liked (compare item attributes)
  - Collaborative filtering: "Users like you also liked..." (user similarity)
  - Hybrid approaches: Combining both
  - Cold start problem: What when new user has no data?
- Study neural network intuition:
  - Layers as progressive feature extraction
  - Why they're good for pattern recognition in images/text
  - How they could be used for fashion (image recognition for wardrobe input, style classification)
- Understand bias and overfitting:
  - What if DressUp only recommends same style repeatedly?
  - How to balance exploration (new combinations) vs. exploitation (proven preferences)?

3. Build Recommendation Engine (8 hours)

- Implement Version 1.0 in your Glide app:
  - Use Google Sheets formulas or Glide logic
  - Apply rule-based scoring to generate outfit suggestions
  - Incorporate weather data (you can hard-code temperature for now)
  - Show top 3 outfit suggestions daily
- Test with your sample wardrobe data
- Document cases where it works well vs. fails

4. Industry Case Study - ML in Fashion (4 hours)

- Deep dive: How does Amazon's recommendation engine work? (research available information)
- Case study: Stitch Fix's algorithms and hybrid human-AI approach
- Extract lessons: What can DressUp learn from these systems?
- Document: At what scale does ML become necessary vs. rules?

5. Feedback Loop Design (2 hours)

- Design how DressUp learns from users:
  - Explicit feedback: Star ratings, "wore this" button, "suggest more like this"
  - Implicit feedback: What outfits are viewed, how long user looks at each
  - Feedback collection interface in app
- Plan: How this data feeds back into improving recommendations

Week 3 Deliverables

1. **Recommendation Algorithm Documentation** (rule-based V1 + ML vision for V2)
2. **Working Recommendation Engine** (integrated into Glide app)
3. **AI/ML Concepts Report** (your understanding of how DressUp could become more intelligent)
4. **ML in Fashion Case Study** (analysis of industry approaches)
5. **Feedback Loop Design** (how system improves over time)
6. **Public Post #3:** "From Rules to Learning: How DressUp Recommendations Work"

MILESTONE: Mid-Point Review

- You now have a working MVP with basic intelligence
- System can intake wardrobe, generate outfit recommendations, incorporate context
- You understand how to evolve it toward true ML
- Portfolio has 3 substantive case studies

Week 4: Automation, Deployment, & User Experience

Theme: "Make It Useful Daily"

Core Concepts

- Automation thinking: triggers, actions, workflows
- Deployment considerations: what makes an MVP "shippable"?

- User experience design: reducing friction, building habits
- Notification strategies and engagement loops

DressUp Tasks

1. Daily Automation Design (8 hours)

- Build daily outfit suggestion automation:
  - Trigger: Every morning at 7 AM (user's local time)
  - Actions:
    1. Fetch today's weather
    2. Check user's calendar for events
    3. Generate outfit recommendations
    4. Send notification with suggestion
  - Implement using Zapier or similar (connect weather API → Sheets → Glide → notification)
- Design notification copy that feels personal, not spammy
- Create fallback logic for when API fails

2. User Onboarding Flow (6 hours)

- Design 3-minute onboarding experience:
  - Welcome → Style preferences quiz (5-7 questions) → Wardrobe quick-add (camera or manual) → First outfit suggestion
  - Each step must feel easy and valuable
  - Apply behavioral psychology: progress indicators, quick wins
- Build onboarding screens in Glide
- Write all onboarding copy (conversational, encouraging)

3. Habit Formation Design (4 hours)

- Research habit loop: Cue → Routine → Reward
- Apply to DressUp:
  - **Cue:** Morning notification with weather-based outfit
  - **Routine:** Open app, see suggestion, give feedback
  - **Reward:** Decision made, time saved, feel confident
- Design streak tracking: "7 days using DressUp!"
- Plan engagement features: achievements, style insights, "most worn items"

4. MVP Polish (8 hours)

- Improve app UI/UX:
  - Visual design: color scheme, typography, iconography
  - Photo quality: create or source attractive wardrobe item images
  - Micro-interactions: smooth transitions, loading states
  - Error states: what happens when things go wrong?
- Write all app copy: button labels, help text, empty states
- Test entire user journey and fix friction points

5. Deployment Preparation (6 hours)

- Create launch checklist:
  - Technical: All features working, data model stable, APIs connected
  - Content: Help documentation, FAQs, terms of service
  - Marketing: App store screenshots, description, demo video
- Document what "done" looks like for this MVP
- Identify known limitations and planned improvements
- Create user testing plan: 5-10 people to beta test

Week 4 Deliverables

1. **Working Daily Automation** (morning outfit notifications)
2. **Complete Onboarding Flow** (in-app experience)
3. **Habit Design Document** (engagement strategy)
4. **Polished MVP** (production-ready app)
5. **Launch Checklist** (deployment preparation)
6. **Public Post #4:** "Designing Habit-Forming AI Products: DressUp's Engagement Strategy"

Week 5: Industry Knowledge, Business Model, & Case Studies

Theme: "Position DressUp in the Market"

Core Concepts

- Fashion retail economics and supply chain
- Business model design: user value → revenue strategy

- Competitive positioning and differentiation
- Case study methodology and writing

DressUp Tasks

1. Deep Industry Analysis (10 hours)

- Fashion retail supply chain deep dive:
  - How do seasons affect inventory? (Spring/Summer vs. Fall/Winter collections)
  - What is "fast fashion" vs. traditional retail cycle?
  - How do retailers forecast demand?
  - Why do returns hurt profitability?
- User wardrobe insights for retailers:
  - What if retailers knew what's already in customers' closets?
  - How could this reduce returns? (suggest items that match what they own)
  - Inventory optimization: push items that pair with popular customer wardrobe pieces
- Document 5 specific ways DressUp data creates value for fashion brands/retailers

2. Business Model Design (8 hours)

- Refine DressUp monetization strategy:
  - **User side:** Freemium (basic free, premium features $4.99/month)
    - Free: Daily outfit, basic wardrobe, weather integration
    - Premium: Unlimited outfits, style analytics, shopping recommendations, calendar integration
  - **Business side:** Data insights and API licensing
    - Aggregate (anonymized) trend data for retailers
    - Wardrobe gap analysis: what your customers need next
    - API for "style this item" functionality on retail sites
  - **Affiliate revenue:** Commission on referred purchases (15%)
- Create financial model:
  - Project user acquisition costs
  - Estimate conversion rates (freemium → premium)
  - Calculate LTV (lifetime value) vs. CAC (customer acquisition cost)
  - Build simple 3-year revenue projection
- Justify pricing strategy with market research

3. Competitive Analysis (6 hours)

- Deep analysis of 5 competitors:
  - What do they do well? Where do they fail?
  - How is DressUp different/better?
  - What market positioning do we take?
- Create competitive matrix: feature comparison
- Identify DressUp's unique value propositions:
  - AI-powered daily suggestions (not just catalog browsing)
  - Wardrobe-aware (suggests from what you own)
  - Context-intelligent (weather + calendar + preferences)
  - Data value for retailers (supply chain optimization angle)

4. Case Study Writing (6 hours)

- Write comprehensive DressUp case study:
  - **Context:** The decision fatigue problem
  - **Problem:** No AI-powered solution that learns YOUR wardrobe
  - **Solution:** DressUp system architecture and features
  - **Approach:** How you built it (architecture, prompts, no-code)
  - **Results:** What the MVP can do now, what's next
  - **Insights:** Key lessons on AI system design
- Structure for portfolio and public sharing
- Include visuals: architecture diagrams, app screenshots, user flow

5. Pitch Deck Creation (2 hours)

- Update your presentation deck with:
  - Clearer problem statement and solution
  - Live demo capability (your working MVP)
  - Refined business model with realistic projections
  - Your unique insight on supply chain value
  - Traction plan: how you'll get first 1,000 users

Week 5 Deliverables

1. **Fashion Industry Deep Dive Report** (supply chain + retail economics)

2. **Business Model Canvas** (revenue streams, cost structure, value proposition)
3. **Competitive Analysis Matrix** (positioning strategy)
4. **Comprehensive DressUp Case Study** (portfolio-ready)
5. **Updated Pitch Deck** (investor/partner ready)
6. **Public Post #5:** "The Economics of AI Fashion: Why DressUp Matters to Retailers"

Week 6: Testing, Portfolio, Public Communication

Theme: "Ship It and Tell the Story"

Core Concepts

- User testing methodology
- Portfolio presentation and personal branding
- Public communication: translating technical work for different audiences
- Iteration based on feedback

DressUp Tasks

1. User Testing (10 hours)

- Recruit 8-10 beta testers (friends, family, online communities)
- Create testing protocol:
  - Onboarding: Can they set up in 3 minutes?
  - Daily use: Do they find value in recommendations?
  - Feature discovery: Do they understand all capabilities?
  - Feedback: What works? What's confusing? What's missing?
- Conduct structured interviews (30 min each)
- Analyze patterns in feedback
- Prioritize improvements: critical fixes vs. nice-to-haves

2. Iterate Based on Feedback (8 hours)

- Implement top 3-5 improvements from user testing
- Fix critical bugs or confusion points
- Enhance any features that users loved
- Update documentation to reflect changes

- Do second round of testing with improvements

3. Portfolio Development (8 hours)

- Create comprehensive portfolio showcasing DressUp:
  - **Project overview page:** Problem, solution, your role
  - **System architecture:** Visual diagrams with explanations
  - **Feature showcase:** Screenshots and descriptions
  - **Technical approach:** How you built it (prompts, no-code, APIs)
  - **Business thinking:** Market analysis, monetization strategy
  - **Results and learnings:** What worked, what you'd change
- Design for multiple audiences:
  - AI/ML roles: Show system thinking and prompt engineering
  - Product roles: Show user research and feature prioritization
  - Strategy roles: Show market analysis and business model
- Polish visual presentation

4. Public Case Study Series (4 hours)

- Compile your 5 weekly posts into a series
- Create overarching narrative: "Building DressUp: A 6-Week Journey"
- Add final post #6: "Launch and Learnings: What I Built and What I Learned"
- Ensure each post tells a clear story with takeaways
- Format for LinkedIn, personal blog, or Medium

5. Next Phase Planning (2 hours)

- Document DressUp V2.0 roadmap:
  - Features to add next (calendar integration, social sharing)
  - Technical improvements (actual API deployment, app store submission)
  - ML implementation plan (when you have real user data)
  - Potential partnerships to pursue
- Create 90-day plan for continued development
- Identify 3 companies/roles to target with your portfolio

Week 6 Deliverables

1. **User Testing Report** (findings and improvements)
2. **Polished MVP v2.0** (post-testing improvements)
3. **Complete Portfolio Website/Page** (showcasing entire project)
4. **Public Case Study Series** (6 posts published)
5. **DressUp V2.0 Roadmap** (next phase plan)
6. **Demo Video** (3-5 min walkthrough of DressUp)

FINAL MILESTONE: Launch-Ready MVP

- Working DressUp app with core features
- Real user testing and validation
- Complete portfolio demonstrating AI system architecture skills
- Public presence establishing you as an AI product thinker
- Clear plan for next steps

Cross-Cutting Learning Themes

Prompt Engineering Practice

Every week, you'll build and refine prompts. By end:

- 50+ production prompts for various DressUp functions
- Documented comparison of Claude vs. ChatGPT for different tasks
- Framework for prompt debugging and optimization
- Portfolio of prompt engineering work

Systems Thinking Application

Each week connects to others:

- Week 1 architecture informs Week 2 data model
- Week 3 recommendation logic uses Week 2 data structure
- Week 4 automation brings Week 3 logic to daily life
- Week 5 business model monetizes Week 1-4 work
- Week 6 validates and showcases entire system

Decision Frameworks

Integrated throughout:

- Feature prioritization (Kano model, MoSCoW)
- Trade-off analysis (MVP scope, technical debt)
- Game theory lens (user incentives, engagement loops)
- Business decisions (pricing, partnerships, positioning)

Success Metrics

By end of 6 weeks, you will have:

Technical Capabilities:

- ✅ Designed complete AI system architecture
- ✅ Built 50+ production-quality prompts
- ✅ Created working no-code MVP app
- ✅ Integrated external APIs
- ✅ Implemented recommendation algorithm
- ✅ Set up automation workflows

Knowledge Domains:

- ✅ AI/ML concepts (without coding)
- ✅ Recommendation system design
- ✅ Fashion retail industry understanding
- ✅ Business model design
- ✅ User experience principles
- ✅ Data architecture

Portfolio Assets:

- ✅ Working DressUp MVP (demonstrable)
- ✅ Complete system documentation
- ✅ 6 published case studies
- ✅ Professional portfolio page
- ✅ Pitch deck

- ✅ Demo video

Career Positioning:

- ✅ Can speak credibly about AI systems with engineers
- ✅ Portfolio proving product thinking and system design
- ✅ Public presence in AI/product space
- ✅ Real project showing end-to-end capabilities
- ✅ Clear narrative for AI strategy/product roles

Working with Your INTJ Profile

Leveraging Strengths:

- Complex systems → You're designing DressUp architecture end-to-end
- Pattern recognition → You're identifying how components connect
- Analytical thinking → Every decision is justified with logic
- Writing clarity → Case studies showcase your ability to explain
- Independence → Self-directed learning with clear milestones

Managing Challenges:

- Overthinking → Each week has defined scope; move forward even if not "perfect"
- Too many ideas → MVP feature list is bounded; save ideas for V2
- Perfectionism → 80% done and shipped beats 100% never launched
- Need for structure → Detailed weekly plans with clear deliverables
- Self-doubt → Tangible deliverables each week prove progress

Learning Style Match:

- ✅ Structured plan with clear milestones
- ✅ Conceptual understanding before implementation
- ✅ Natural language explanations over formulas
- ✅ Learn by doing (DressUp) not abstract exercises
- ✅ Frameworks and mental models throughout
- ✅ Real use case (DressUp) connects all concepts

- ✅ Multiple AI models as thinking partners

Daily Rhythm Suggestion

Recommended 30-32 hour weekly allocation:

Building Days (Mon-Wed-Fri): 7 hours each = 21 hours

- Deep work on DressUp implementation
- Prompt development and testing
- App building in Glide
- Integration work

Learning Days (Tue-Thu): 4 hours each = 8 hours

- Industry research and case studies
- AI/ML concept study
- Competitive analysis
- Reading and note-taking

Documentation Day (Sat): 4 hours

- Write weekly case study
- Update portfolio
- Reflect on learnings
- Plan next week

Accountability: Weekly review each Sunday (30 min)

- What shipped this week?
- What was learned?
- What's the plan for next week?
- Any adjustments needed?

Using AI as Your Learning Partner

Throughout this plan:

Use Claude for:

- Explaining complex concepts in intuitive terms
- Reviewing your architecture decisions
- Debugging prompts and logic
- Writing assistance (case studies, documentation)
- Acting as "technical advisor" for questions

Use ChatGPT for:

- Quick factual research
- Code snippets or formula help
- Brainstorming features or approaches
- Alternative perspectives on problems

Best Practice:

- Ask "explain this as if you're teaching an INTJ who thinks in systems"
- Request structured frameworks, not just answers
- Use AI to generate examples, then understand the pattern
- Have AI review your work and suggest improvements

Final Note: Ambitious but Achievable

This plan is intense but designed for your capabilities. You're not just learning AI concepts - you're building a real product that demonstrates system thinking. By Week 6, you'll have something impressive to show serious people in AI, product, and strategy roles.

The key: **Ship every week.** Don't let perfectionism stop momentum. Each week builds on the last, and by the end, you'll have created something meaningful while proving you can architect AI systems.

Remember: You're not trying to build the perfect app. You're building a proof of concept that demonstrates your ability to think through complex AI systems, make sound architectural decisions, and communicate your thinking clearly. That's what gets you the roles you want.

Now start with Week 1. Design the system brain.