

Full Stack AI Software Development

AI-Assisted UI/UX Design for Software Development

Job Connector Program

Outline

- Mastering User Flows (UX)
- Visual Fundamentals (UI)
- AI-Assisted UI/UX Development



The Mindset Shift

Transition from "**Implementation-First**" to "**User-First**" thinking.

Developers build how it works; Designers build why it matters. To bridge this gap, we must adopt **Design Thinking**.

Key Takeaway: Before writing a single line of code, ask:

“What problem is the user trying to solve, and what is the shortest path to that solution?”

The Mindset Shift

| Feature | Developer Mindset (Output) | Designer Mindset (Outcome) |
|----------------|---|--|
| Focus | "How do I code this authentication system?" | "Why does the user need to log in right now?" |
| Goal | Functionality, Performance, Clean Code. | Usability, Desirability, User Success. |
| Error Handling | "Catch the exception and log it." | "Prevent the error or guide the user to fix it." |
| Success Metric | All tests passed; feature deployed. | User completed the task without frustration. |

Mastering User Flows (UX)

A User Flow is essentially an algorithm for human behavior. Visualize the user journey before defining the UI.

1. The "Happy Path" vs. Edge Cases

- **Happy Path:** The ideal scenario where everything goes right.
- **Edge Cases:** Errors, empty states, and unauthorized access (devs are naturally good at identifying these).

2. Diagramming Standards

- **Oval:** Start/End points.
- **Rectangle:** User Action (e.g., "Clicks Button").
- **Diamond:** Decision/System Logic (e.g., "Is User Logged In?").

3. Action Item: Create a Flowchart

- **Manual Method:** Use FigJam or Draw.io.
- **Developer Method:** Use Mermaid.js syntax to generate charts via code.

How to create a user flow (step-by-step)

Define the goal

Example: “User adds a product”

List user actions

Open app → Browse → Select → Confirm

Add system responses

Loading, validation, success, error

Visualize as flowchart

- Rectangles = screens
- Diamonds = decisions

Visual Fundamentals (UI)

Design is not just "art", it is a **system of rules**. Apply logical rules to design to create consistent, professional interfaces.



Space and Grid System

Space is not decoration. Space is how users understand structure without reading instructions.

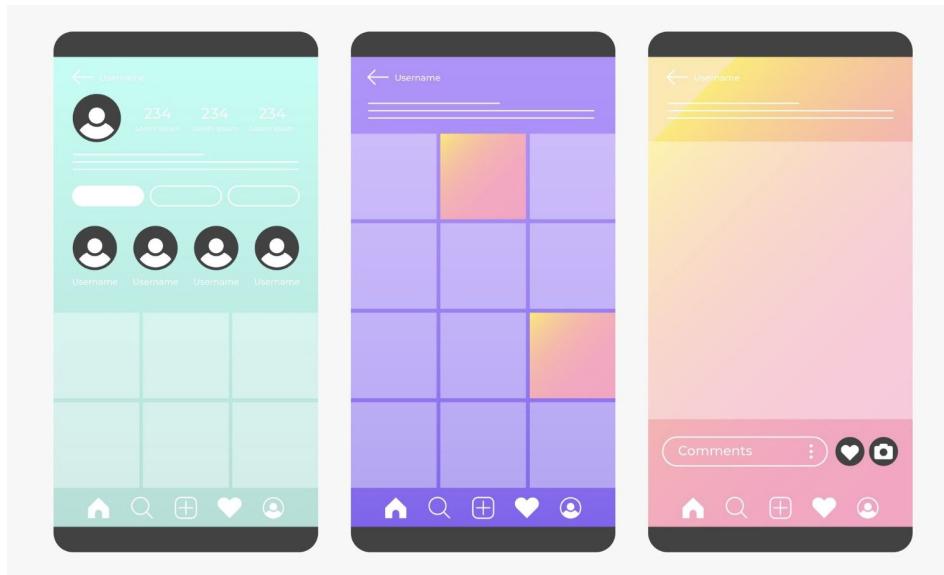
Space = intentional empty area between UI elements.

Why space matters?

- Groups related elements
- Separates unrelated elements
- Reduces cognitive load
- Improves scan-ability

Rule of thumb:

- If everything is close together, nothing feels important.



The Spacing Hierarchy

Spacing should communicate relationships.

Visual logic

- Small space → tightly related
- Medium space → related but distinct
- Large space → different sections

Example (Card UI)

- Icon ↔ Text: small spacing
- Text ↔ Button: medium spacing
- Card ↔ Card: large spacing

If spacing is random, the UI feels messy even if colors and fonts are good.

The image shows two cards side-by-side, each titled "Create a Profile". Both cards contain three input fields: First Name (with value "John"), Last Name (with value "Smith"), and Email (with value "john.smith@email.com").

The card on the left has appropriate spacing between its input fields and includes a green checkmark icon and a long green horizontal bar at the bottom, all indicating a successful operation.

The card on the right, however, lacks proper spacing. The input fields are tightly packed together. It features a red X icon and a long red horizontal bar at the bottom, both indicating an error or failure.

| Field | Value |
|------------|----------------------|
| First Name | John |
| Last Name | Smith |
| Email | john.smith@email.com |

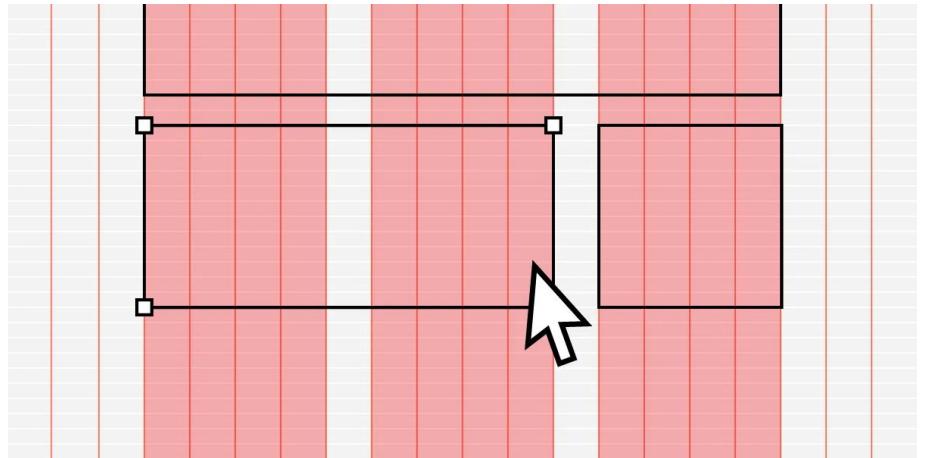
The Grid System

A grid is an invisible alignment system that keeps everything organized.

Why grids exist?

- Predictable layouts
- Easier scaling
- Cleaner responsiveness
- Faster decision-making

Users don't see grids. They feel them.



The 8px Grid System (Industry Standard)

Most modern UI systems use multiples of 8px.

Why 8px?

- Divides cleanly into 2 or 4
- Works well across screen sizes
- Matches most design systems (Material, iOS, Tailwind)

Valid spacing values

4, 8, 16, 24, 32, 40, 48

Avoid

13px, 19px, 27px

Typography and Scaling

Typography is information architecture in text form.

It tells users what to read first, what matters, and what can be skipped.

What Typography Actually Controls:

- Hierarchy → importance
- Readability → effort
- Rhythm → visual flow
- Tone → product personality

Bad typography makes good content hard. Good typography makes average content usable.

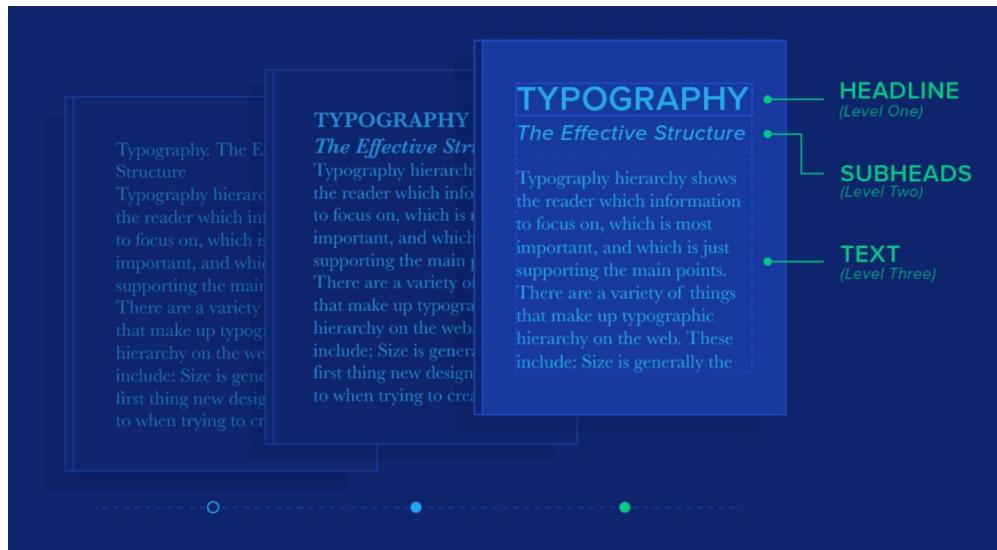


Text Hierarchy

Use clear levels:

- Heading → what this section is about
- Body → main content
- Caption → helper / hint / meta

Users scan in this order.



Font Size Scale

Use fixed steps, not random numbers.

Safe scale:

- Caption: 12px
- Body: 14–16px
- Subheading: 18–20px
- Heading: 24–32px

Rule:

- Each level is clearly bigger than the one below it
- Increase size by 1.2×–1.333×

| Name | Weight | Size | Line Height | Case | Letter Spacing |
|----------------|----------|------|-------------|----------|----------------|
| Title L | Normal | 32 | 40 | Sentence | 0 |
| Title M | Normal | 24 | 32 | Sentence | 0 |
| Title S | Normal | 20 | 28 | Sentence | 0 |
| Subtitle | Semibold | 16 | 24 | Sentence | 0 |
| Subtitle Small | Semibold | 14 | 20 | Sentence | 0 |
| Body | Normal | 16 | 24 | Sentence | 0 |
| Body Small | Normal | 14 | 20 | Sentence | 0 |
| Button | Semibold | 14 | 20 | Sentence | 0 |
| Caption | Normal | 12 | 16 | Sentence | 0 |
| LABEL | Semibold | 12 | 16 | All caps | 1 |

Line Height

Line height affects reading speed more than font size.

Safe defaults:

- Body text: 1.4–1.6
- Headings: 1.2–1.3

Developer analogy:

Line height is vertical padding for text.



Font Weight & Contrast

Weight creates hierarchy without changing size.

Good usage:

- Headings: medium / semibold
- Body: regular
- Emphasis: medium, not bold

Avoid:

- All bold UI
- Using bold for everything important



Font Families

Use 1–2 fonts max.

Common safe combos:

- Sans-serif for UI
- Monospace for code
- Optional serif for marketing pages

Too many fonts = visual chaos.



Color Theory

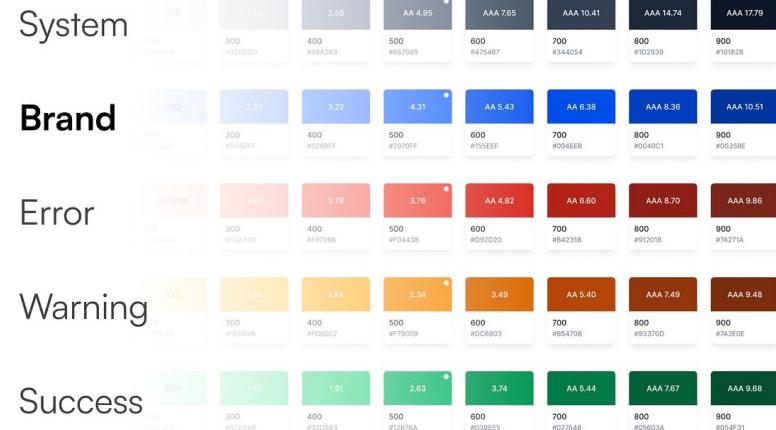
Color is a communication tool, not decoration.

It tells users what to do, what changed, and what needs attention.



Color Roles (Think in Roles, Not Colors)

| Role | Meaning |
|-----------|------------------------------|
| Primary | Main action (Save, Continue) |
| Secondary | Alternative action |
| Success | Completed / OK |
| Warning | Needs attention |
| Danger | Destructive action |
| Neutral | Text, background, border |



Semantic Colors: Standardize status colors (Green = Success, Red = Error, Yellow = Warning, Blue = Info).

The 60-30-10 Rule

- 60%: Primary/Neutral color (Backgrounds).
- 30%: Secondary/Brand color (Headers, active states).
- 10%: Accent color (Call-to-Action buttons).

10% **Accent**

Secondary

#FAEEFF

30%

Dominant

#665FBE

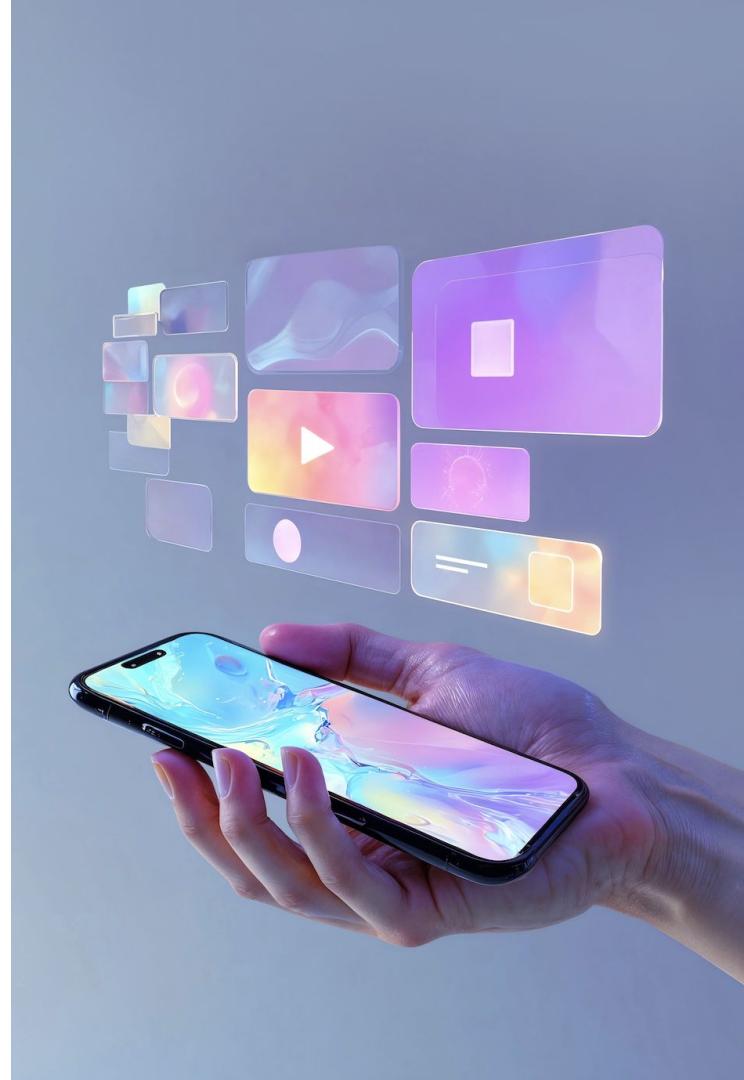
60%

AI-Assisted UI/UX Development

AI doesn't replace designers. It augments your thinking speed.

Leverage AI tools to accelerate the design process:

- Research & Discovery
- User Workflow & Information Architecture
- Visual Design (Color Palette, Assets)
- Full Design Generation (Text-to-UI)



Research & Discovery

Use AI to:

- **Understand users faster**
 - Summarize interview notes, reviews, support tickets
 - Extract pain points, goals, and motivations
- **Generate personas**
 - Convert raw data into personas, empathy maps, Jobs-to-Be-Done
- **Validate ideas early**
 - Ask AI to challenge assumptions (“What could go wrong?”)
- **Competitive & heuristic analysis**
 - Compare similar apps and identify UX gaps

Tools:

- **ChatGPT:** Persona generation, insight synthesis, UX critique
- **Claude:** Long research summary, structured analysis
- **Notion AI:** Research documentation, auto-summaries
- **Figma / FigJam:** Affinity mapping, research boards



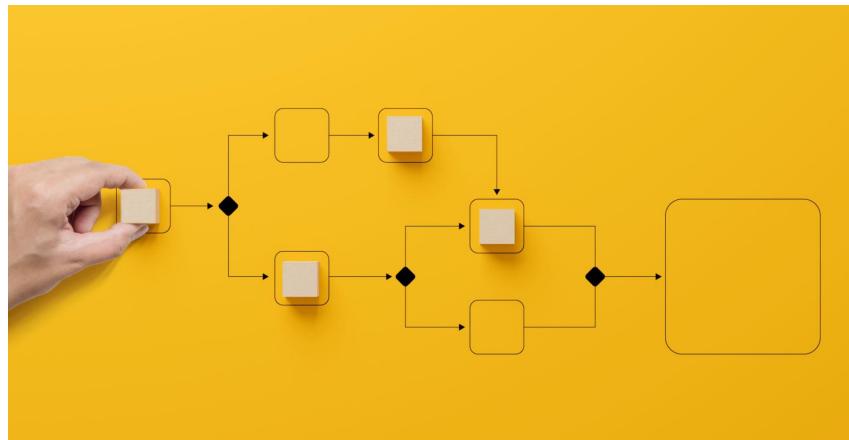
User Workflow & Information Architecture

Use AI to:

- **Generate user flows:** Convert feature ideas into step-by-step workflows
- **Detect UX friction:** Identify unnecessary steps, dead ends, edge cases
- **Alternative flow suggestions:** “Beginner vs power user flow”
- **Error & empty-state thinking:** AI lists possible failures and recovery paths

Tools:

- **ChatGPT:** Flow logic, edge-case analysis
- **Whimsical:** Clean, fast flow diagrams
- **Miro:** Collaborative journey mapping



Visual Design (Color Palette & Assets)

Use AI to:

- **Generate color palettes:** Brand-aligned, mood-based, accessible palettes
- **Check accessibility:** Contrast ratios, color-blind safety
- **Create assets:** Icons, illustrations, background visuals

Tools:

- **Midjourney / DALL-E 3:** High-fidelity image and illustration generation.
- **Nano Banana (Gemini Image Model):** Best for consistent character assets, "toy-style" 3D visualizations, and blending reference images.
- **Huemint:** AI-trained color palette generators that learn your preferences.
- **Storyset:** Ready-to-use illustrations



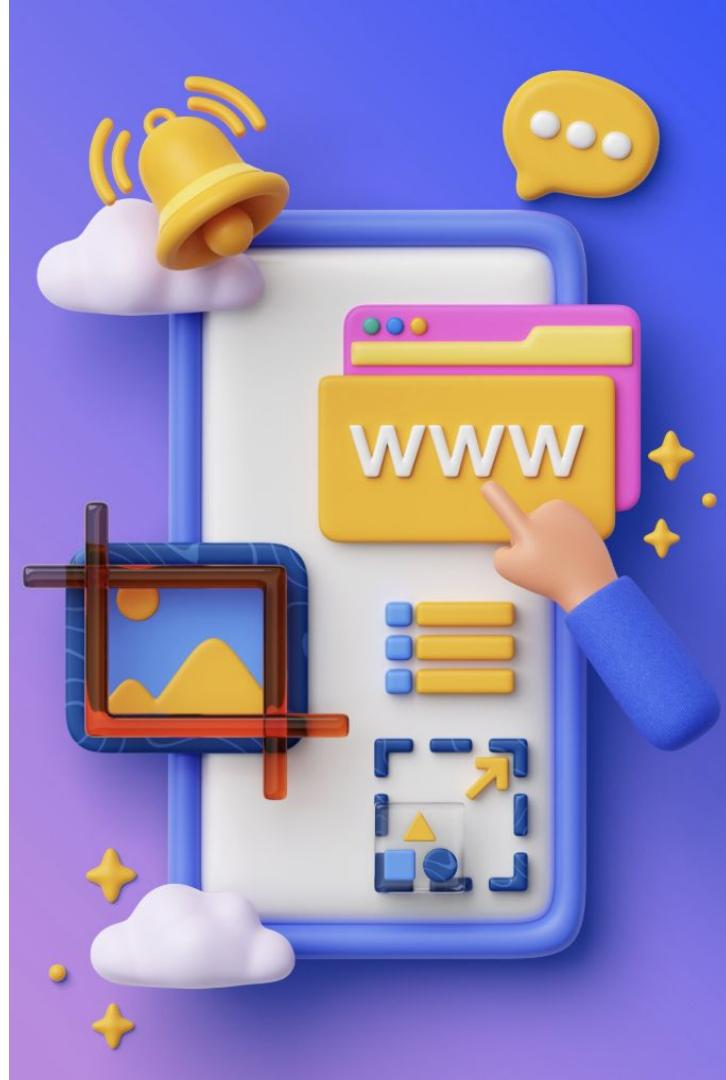
Full Design Generation

Use AI to:

- **Prototype Instantly:** Turn a text prompt or a screenshot into an editable high-fidelity design.
- **Code Generation:** Convert UI designs into production-ready code (React, Tailwind, HTML).
- **Iterate Variations:** Instantly generate A/B/C variations of a landing page or component.
- **Design Systems:** Auto-generate style guides, buttons, and typography scales.

Tools:

- **Google Stitch:** It excels at converting text/sketches into fully layered designs and clean front-end code.
- **v0.dev (by Vercel):** Generates copy-paste friendly UI components (React/Tailwind) from prompts.
- **Lovable / Bolt.new:** Generates full-stack web applications to test UI in a live environment.



Thank You!

