

Theoretical and Practical Frameworks for User Interface Architecture and Version-Controlled Project Initialization

The genesis of any robust digital product is not found in the first line of code or the final aesthetic polish of a high-fidelity mockup, but rather in the deliberate, skeletal planning phase known as wireframing and the structured setup of an institutional-grade version-control environment. In the contemporary software development lifecycle (SDLC), these two pillars—interface architecture and project infrastructure—serve as the bedrock for scalable, maintainable, and user-centric applications.¹ This report examines the intricate methodologies of low-fidelity prototyping, the tactical implementation of GitHub repositories for collaborative engineering, and the professional frameworks required to justify design decisions within a competitive industrial context.

The Strategic Imperative of Low-Fidelity Wireframing

Wireframing constitutes the process of creating a structural blueprint for a website or mobile application, acting as a black-and-white skeleton that prioritizes functionality, layout, and user flow over visual ornamentation.³ Professional practitioners utilize these simplified diagrams to experiment with page layouts and information architecture before committing significant engineering resources to a project.³ The primary focus remains on the placement of buttons, menus, and content blocks to determine the optimal user experience (UX) and navigational clarity.¹

The Economics of Early-Stage Prototyping

The economic rationale for beginning with low-fidelity (lo-fi) wireframes is substantial. Research indicates that the cost of correcting a structural or usability flaw increases exponentially as a project progresses through the development stages. According to data from Forrester Research, fixing a user experience issue during the initial design phase is approximately ten times less expensive than addressing the same issue after the product has been deployed or even during the mid-development cycle.⁴ Furthermore, the Nielsen Norman Group observes that teams utilizing low-fidelity wireframes report a prototyping speed that is nearly 50% faster than those who attempt to start directly with high-fidelity tools.⁴ This efficiency allows for a higher volume of conceptual exploration within the same temporal constraints.

Prototyping Metric	Low-Fidelity (Lo-Fi)	High-Fidelity (Hi-Fi)	Strategic Implication
--------------------	----------------------	-----------------------	-----------------------

Speed of Creation	Minutes to Hours ⁵	Days to Weeks ⁵	Lo-fi enables rapid iteration. ⁴
Cost to Pivot	Minimal/Negligible ⁵	High/Significant ⁴	Lo-fi reduces financial risk. ⁵
Feedback Focus	Structure & Logic ⁵	Aesthetics & Detail ³	Lo-fi prevents "premature polish". ⁵
Engineering Effort	Zero to Low ⁵	High ⁵	Lo-fi preserves developer resources. ⁵
Post-Launch Usability	Higher Success Rate ⁶	Higher Risk of Flaws ⁶	Multi-stage validation reduces bugs. ⁶

Psychological Safety and Feedback Mechanisms

A critical, albeit often overlooked, benefit of the low-fidelity approach is the creation of "psychological safety" among stakeholders. When presented with a crude, hand-drawn sketch or a monochrome digital wireframe, stakeholders and users are fundamentally more likely to provide honest, transformative feedback.⁵ A polished, high-fidelity design often signals that the work is nearly complete, which may discourage critics from suggesting major structural changes for fear of invalidating the designer's significant effort.⁵ Conversely, the "unfinished" appearance of a lo-fi wireframe invites collaboration and experimentation, ensuring that the fundamental concept is sound before aesthetic layers are applied.⁵

The Taxonomy of Prototyping Fidelity

Fidelity in interface design is not a binary state but a multi-axial spectrum encompassing visual detail, content accuracy, and interactive complexity.⁵ Navigating this spectrum is essential for aligning the prototype's complexity with the project's current maturity level.

The Three Axes of Fidelity

Research by the Nielsen Norman Group defines fidelity along three primary dimensions: visual, content, and interactivity.⁵ In the earliest stages, all three axes remain at a low level of resolution.

1. **Visual Axis:** This refers to the graphical sophistication of the design. Lo-fi wireframes occupy the lowest point here, utilizing boxes, lines, and grayscale shapes to represent elements.⁴
2. **Content Axis:** This involves the accuracy of the information displayed. Early wireframes

often use "lorem ipsum" or basic labels, whereas medium-fidelity versions transition to actual drafted copy to ensure the layout can accommodate real-world data.⁴

3. **Interactivity Axis:** This describes the degree to which a user can interact with the prototype. Lo-fi versions are typically static, while high-fidelity prototypes feature sophisticated animations and logic-driven transitions.⁴

Progression through the Design Lifecycle

A standardized design workflow involves progressing through these fidelity levels as the concept solidifies. According to the Interaction Design Foundation, teams that validate their designs across all fidelity levels—starting lo-fi and moving toward hi-fi—experience 37% fewer usability issues after launch compared to those who skip the foundational stages.⁶

Stage	Fidelity Level	Core Components	Primary Goal
Ideation	Low-Fidelity	Paper sketches, whiteboard drawings ⁶	Validate fundamental concept. ⁶
Definition	Mid-Fidelity	Wireframes with real copy and navigation ⁶	Test information architecture. ⁶
Validation	High-Fidelity	Pixel-perfect UI, realistic interactions ⁶	Final stakeholder approval. ⁶

Toolsets for Interface Architecture in 2025

The choice of wireframing tools is dictated by the specific needs of the project, the team's collaborative requirements, and the desired level of fidelity. While traditional analog methods like pen and paper remain highly effective for rapid brainstorming, digital platforms offer scalability and integration with development environments.⁵

Specialized Wireframing and Prototyping Platforms

Figma has established itself as a leading all-in-one platform for collaborative UI/UX design, offering real-time feedback and a robust "dev handoff" feature that provides developers with code-ready specifications.⁷ However, for practitioners seeking to maintain a low-fidelity focus, tools like Balsamiq are often preferred. Balsamiq deliberately uses a "sketch-style" visual language to prevent stakeholders from getting distracted by aesthetic details, thereby keeping the conversation focused on structure and flow.⁷

Design Tool	Best For	Standout Feature	Pricing Model
Figma	High-fidelity UI & Collaboration ⁸	Auto-layout & Dev Mode ⁷	Freemium ⁹
Balsamiq	Low-fidelity structural sketching ⁸	Hand-drawn aesthetic ⁷	Paid (Free trial) ⁸
Miro	Brainstorming & Infinite Canvases ⁸	Template-rich whiteboards ⁷	Freemium ⁹
Frame0	Rapid AI-aided app ideation ¹⁰	Prompt-to-wireframe generation ¹⁰	Free tier ¹⁰
Draw.io	Open-source system diagramming ⁹	Full Google/MS integration ⁹	Free/Open-source ⁹

The Integration of Artificial Intelligence in Layout Design

A transformative trend in 2025 is the emergence of AI-powered design tools that automate the initial stages of wireframing. Platforms like ZipWP allow users to input a business description and generate a complete website structure, including sitemaps and wireframes, in under 60 seconds.⁷ This "AI site planner" phase allows designers to focus on high-level strategy and branding before the final build is executed.⁷ Similarly, Frame0 utilizes AI to generate code snippets directly from sketch-like wireframes, significantly accelerating the transition from design to development.¹⁰

Methodological Execution of Mobile Interface Design

Designing for mobile devices introduces unique constraints that are not present in desktop environments, primarily due to limited screen real estate and the ergonomics of touch interactions.

The Mobile-First Paradigm and User Flows

Mobile wireframing requires a "lean" approach, prioritizing the most critical user actions to avoid cluttering small screens.¹¹ Practitioners should begin by mapping a "wireflow"—a sequence of simplified screens that represent the user's path to a goal.¹¹ For example, a workout booking application might feature a minimal sequence: Home → Filter → Class Details → Checkout → Confirmation.¹¹ Reducing the number of steps in such flows is vital; research suggests that simplifying checkout processes can increase mobile conversion rates by up to

35%.¹⁴

Ergonomics and the "Thumb Zone"

A professional mobile wireframe must account for physical usage patterns. The "thumb zone"—the area of the screen most easily reached by a user's thumb during one-handed operation—should contain the primary call-to-action (CTA) buttons.¹¹ Vital interactive elements are typically placed in the lower two-thirds of the screen to ensure accessibility.¹¹ Furthermore, buttons must be designed with "tap-friendly" dimensions, ideally no smaller than 44px by 44px, to prevent user error and frustration.¹¹

Technical Infrastructure: GitHub Repository Initialization

Once the interface architecture is conceptualized, the project requires a secure, version-controlled environment to house code, documentation, and design assets. GitHub serves as the primary global hub for such collaboration, providing a comprehensive history of revisions and ensuring that no part of the development process is lost over time.²

Repository Creation and Naming Conventions

Initializing a GitHub repository involves several critical decisions regarding naming, visibility, and documentation. Naming conventions suggest using short, memorable, and lowercase identifiers, with words separated by hyphens (e.g., application-interface-prototype).¹⁵ This ensures cross-platform compatibility and ease of use in command-line environments.

Repository Attribute	Selection Criteria	Strategic Importance
Visibility: Public	Open-source or Portfolio projects ²	Demonstrates skill to the community. ¹⁷
Visibility: Private	Proprietary or Internal work ¹⁵	Protects intellectual property. ¹⁵
README Initialization	Mandatory for project context ¹⁸	Acts as the project's front page. ¹⁹
.gitignore Selection	Based on tech stack (e.g., Node, Python) ²	Prevents repository bloat. ²⁰
License Choice	MIT, Apache, or GPL ¹⁹	Defines legal usage of code. ¹⁹

Authentication and Connectivity

For advanced users, interacting with GitHub via the Command Line Interface (CLI) or GitHub Desktop app provides increased efficiency. Connecting a local repository to a remote GitHub server requires authentication, typically through a Personal Access Token (PAT) or SSH key, which replaces the traditional password-based system for increased security.¹⁶ The use of the "origin" alias is a standard convention for referencing the central remote repository during local development.¹⁶

Professional Workflow and Version Control Standards

Consistency in recording changes—known as committing—is fundamental to professional project management. A well-maintained Git history allows teams to track the evolution of a project and revert to previous states if errors are introduced.

The Anatomy of a High-Quality Commit

Professional practitioners follow strict guidelines for Git commit messages. The message should consist of a concise summary (limited to approximately 50 characters) and a more detailed description if necessary.²¹ A widely adopted standard is the use of the "imperative verb form," which describes what the commit *does* rather than what the developer *did*.²¹

Conventional Commit Type	Use Case	Example
feat	Adding a new feature or component ²¹	feat: add mobile login wireframe ²¹
fix	Correcting a bug or layout error ²¹	fix: align CTA button to thumb zone ²¹
docs	Changes to documentation or README ²¹	docs: update setup instructions ²¹
style	Formatting changes (no logic changes) ²¹	style: fix indentation in config ²¹
refactor	Code changes that neither fix nor add ²¹	refactor: simplify auth logic ²¹

One formula for ensuring compliance with the imperative standard is to complete the sentence: "If applied, my commit will...".²¹ For example: "If applied, my commit will *Add new login screen assets*." This results in the professional-grade commit message: *Add new login screen*

assets.²¹

Asset Management and Image Embedding

For design-focused repositories, the README.md file often serves as a gallery for wireframes and prototypes. The most resilient method for embedding images is to store them directly within the repository—ideally in an assets or images folder—and reference them using relative paths.²⁰ This approach avoids the fragility of external hosting services, which may delete images or change URLs over time.²⁰

The Markdown syntax for embedding an image is: `!(./assets/image-name.png)`.²⁰

Before committing these assets, practitioners must optimize image file sizes to prevent repository bloat. Tools like TinyPNG or ImageOptim are essential for reducing the footprint of high-resolution screenshots, ensuring that the repository remains performant and easy to clone for collaborators.²⁰

Justifying Design Decisions: Rationale and Rationale

In the professional arena, a designer's ability to explain the logic behind a layout is as important as the layout itself. Design is frequently subjective to non-designers, making it necessary to ground decisions in data and psychological principles to gain stakeholder buy-in.²³

The QOC (Questions, Options, Criteria) Framework

A structured methodology for documenting design rationale is the QOC framework, which traces the evolution of a feature from problem to solution.

1. **Questions:** What core user problem are we trying to solve? (e.g., "How do we reduce friction in the checkout flow?").²³
2. **Options:** What were the different design approaches considered? (e.g., "A single-page checkout versus a multi-step accordion").²³
3. **Criteria:** What data or design principles led to the final choice? (e.g., "Usability testing showed a 15% higher completion rate for the multi-step version").²³

Leveraging the "Laws of UX"

Justifications should be rooted in established cognitive principles rather than personal preference. One such principle is Jakob's Law, which posits that users spend most of their time on *other* websites and therefore prefer your site to function similarly to the ones they already know.²³ By citing such laws, a designer can defend a familiar navigation structure as a way to reduce cognitive load and improve user retention.²³

Career Advancement through Repository Portfolio

Engineering

For junior developers and designers, a GitHub profile serves as a "living resume" and a public showcase of technical proficiency. Beyond the simple accumulation of code, recruiters look for evidence of professional discipline and collaborative capability.²⁵

Indicators of a High-Quality Repository

Hiring managers often utilize a checklist to evaluate candidates based on their GitHub activity. Key indicators include:

- **Cleanliness and Organization:** Are related files grouped into logical directories (e.g., `src/`, `tests/`, `docs/`)?.²⁵
- **Documentation Quality:** Does the README explain *what* the project is, *why* it was built, and *how* to use it?.¹⁷
- **Commit Consistency:** Does the contribution graph show regular, thoughtful progress rather than sporadic, massive uploads?.²⁵
- **Collaboration Fluency:** Does the candidate demonstrate an understanding of Pull Requests (PRs), code reviews, and merge conflict resolution?.²⁶

Community Integration and Networking

Professional growth is significantly accelerated by engagement with local and global design communities. In regions like Indore, India, organizations such as IxDF Indore provide a platform for designers and developers to network, share challenges, and stay updated on emerging trends like AI in the UX ecosystem.²⁸ Participating in these meetups—whether focusing on "UX Research 101" or "Creativity Meets Code"—helps practitioners build the "Connected Professional" distinction, which honors ongoing engagement and career visibility within the industry.²⁸

Conclusions and Practical Implications

The integration of wireframing and version-controlled project initialization is not merely a procedural requirement but a strategic foundation for software excellence. By adopting a low-fidelity approach to initial design, teams can realize significant cost savings and foster a culture of honest, transformative feedback.⁴ This architectural work must be mirrored by the creation of a disciplined technical environment on GitHub, characterized by clean commit histories, optimized asset management, and comprehensive documentation.²

Ultimately, the goal of these foundational stages is to align user needs with business objectives through a transparent, defensible, and iterative process. Whether a practitioner is designing a simple mobile login screen or a complex enterprise dashboard, the principles remain the same: plan structurally, document rationally, and manage version-control professionally. These skills not only ensure the success of individual projects but also serve as the primary indicators of

senior-level competency in the global technology market.

Strategic Extension: The Methodology of Comprehensive Report Construction

(Note: To reach the 10,000-word depth requested, the following sections will delve into the granular mechanics of every concept mentioned above, expanding on the historical context, psychological theories, and technical syntax required for mastery.)

Historical Evolution of Design Documentation

To understand the current state of wireframing, one must look at the transition from static documentation to interactive systems. In the early 2000s, design documentation often consisted of hundreds of pages of static PDF specifications. These documents were brittle and became obsolete as soon as the first development sprint began. The shift toward "Agile UX" and "Lean UX" in the 2010s transformed wireframes into "disposable artifacts" that facilitate communication rather than serving as immutable laws.⁴

This transition was mirrored in the tool market. Sketch (launched in 2010) revolutionized the field by offering a lightweight, vector-based alternative to Adobe Photoshop, which was originally designed for photo editing rather than interface design.³¹ However, Sketch was limited by its lack of native collaboration features, leading to the rise of InVision as a prototyping layer.⁹ Figma's emergence represented the next evolutionary step: a browser-native tool where the design, the prototype, and the documentation coexist in a single, real-time environment.⁷

Deep Dive: The Mechanics of the Standard Development Lifecycle (SDLC)

Wireframing plays a specific role in each phase of the SDLC, from requirements gathering to testing.

Phase 1: Requirements Gathering and Discovery

During this phase, wireframes act as a "visual transcript" of stakeholder interviews. Instead of relying on abstract text-based requirements, which are prone to misinterpretation, a lo-fi wireframe provides a concrete reference point for discussion.¹ This phase is less about design and more about defining the "scope" of the application. For instance, if a stakeholder requests a "complex filtering system," a wireframe can quickly show the implications of that request on the screen layout, allowing for an early discussion on feasibility and user experience.¹

Phase 2: Design and Prototyping

In the design phase, the wireframe evolves from a static sketch into a clickable prototype. This is where "wireflows" become essential. A wireflow combines the structural detail of a wireframe with the sequence logic of a flowchart.¹¹ This allows designers to test "happy paths"—the most

common routes a user takes—as well as "edge cases," such as what happens when a user enters an incorrect password or loses internet connectivity.³

Phase 3: Development and Implementation

For developers, wireframes serve as the "logical blueprint." While a high-fidelity mockup shows the final colors and fonts, the wireframe shows the underlying grid system, the hierarchy of headings (H1, H2, etc.), and the functional requirements of each component.¹ Modern tools like Figma enhance this by providing "CSS inspectors," but a well-annotated lo-fi wireframe is still necessary to explain the *behavior* of elements, such as how a button should react when hovered over or how a menu should collapse on mobile devices.¹

Detailed Analysis: Mobile-Specific Interaction Patterns

The shift to mobile-first design in the last decade has forced a re-evaluation of standard desktop patterns.

Thumb-Driven Navigation Systems

On desktop, the "F-Pattern" of reading (top to bottom, left to right) dominates because users interact with a mouse.⁵ On mobile, the "Thumb Zone" is the primary constraint. Professional wireframes for mobile apps often utilize a "Bottom Navigation Bar" rather than a "Hamburger Menu" (the three-line icon in the top corner). The reason is ergonomic: the bottom bar is within the "Natural Thumb Zone," whereas the top corners are in the "Ow" or "Hard to Reach" zones for most users.¹¹

The Psychology of Mobile Friction

Every tap on a mobile screen represents a potential point of abandonment. Practitioners use wireframing to identify and remove "interaction friction." This includes:

- **Inline Validation:** Showing a checkmark next to a field as soon as it is filled correctly, rather than making the user wait until they hit "Submit" only to receive an error message.¹²
- **Adaptive Keyboards:** Specifying in the wireframe that the "Numeric" keyboard should appear when the user taps a phone number field, or the "Email" keyboard should appear for the email field.¹⁴
- **Skeleton Screens:** Using the wireframe to plan for "perceived performance." Instead of a blank screen or a spinner, the app shows a grayscale "skeleton" of the content that is loading, which makes the app feel faster to the user.¹²

Technical Deep Dive: GitHub and the DevOps Mindset for Designers

The initialization of a GitHub repository is the first step toward a "DevOps" culture, where

design and development are integrated rather than siloed.

The Philosophy of "Everything as Code"

The trend toward "Design Systems" means that design assets are increasingly managed like code. This is why version control is critical. If a designer changes the primary "Action Color" in a design system, that change should be recorded as a Git commit, allowing the team to see who made the change, why they made it, and what the previous color was.²¹

Advanced Git Techniques for Project Management

Professional repositories often utilize "Branch Protection Rules" on GitHub. For example, a project lead might configure the main branch so that no one can push code to it directly. Instead, every change must be made on a "feature branch," and a "Pull Request" must be opened and approved by at least one other team member.¹⁸ This ensures that the official version of the project (the one a recruiter or client sees) is always stable and functional.¹⁸

Git Operation	Command Line Syntax	Strategic Purpose
Initialization	<code>git init</code>	Starts tracking a new project locally. ¹⁵
Staging	<code>git add.</code>	Prepares all changed files for a commit. ²⁰
Committing	<code>git commit -m "message"</code>	Records a snapshot of the work. ²¹
Branching	<code>git checkout -b name</code>	Creates an isolated space for new work. ¹⁸
Linking Remote	<code>git remote add origin URL</code>	Connects the local repo to GitHub. ¹⁶
Pushing	<code>git push origin main</code>	Uploads local work to the server. ²

The Art of Professional Documentation: Markdown Mastery

Markdown is the language of the internet's documentation. For a project's README to be professional, it must utilize advanced Markdown features beyond simple headers and lists.

Structural Formatting for Readability

A professional README uses Markdown to create a "visual hierarchy" that mimics the principles of good UI design.

- **Code Blocks:** Using triple backticks (```) to highlight installation commands, making them easy for other developers to copy and paste.³³
- **Keyboard Shortcuts:** Using the <kbd> HTML tag to represent physical keys (e.g., Ctrl + C), which improves the accessibility and professionalism of technical guides.³³
- **Badges:** Using services like Shields.io to add dynamic "badges" to the top of the README, showing things like build status, license type, or version number.³⁴

Inclusive Design and Accessibility in Documentation

Professionalism in design documentation includes a commitment to accessibility. Every image embedded in a GitHub README must have "alt-text"—a textual description for screen readers.³³

- **Correct Alt-Text:** !.²⁰
- **Incorrect Alt-Text:** ![image] or ![screenshot123].³³ The goal is to convey the *meaning* and *context* of the image so that a visually impaired developer or recruiter can still understand the project's logic.²²

The Future of the Discipline: AI, VR, and Conversational UX

As we look toward 2026 and beyond, the role of wireframing will continue to adapt to new technologies.

AI as a Collaborative Partner

AI is no longer just a tool for generating layouts; it is becoming a "reviewer." Future design platforms will likely include AI that can "audit" a wireframe for accessibility violations, suggest improvements to the thumb-zone ergonomics, or even predict the "Eye-Tracking Heatmap" of a layout before a single user has seen it.⁷

Beyond Screens: Wireframing for New Interfaces

Wireframing is also expanding into "VUI" (Voice User Interfaces) and "XR" (Extended Reality).

- **Voice UX:** Wireframes for voice assistants like Alexa focus on "Dialogue Trees"—mapping out what the assistant says and how it handles different user responses.²⁹
- **Spatial UX:** For VR and AR, wireframes must be three-dimensional, accounting for the user's physical position in space and the "Field of View" (FOV) of the headset.³⁶

Professional Ethics and Responsible Design

A senior-level understanding of wireframing includes an awareness of "Dark Patterns"—design choices that trick users into doing something they didn't intend to, such as hidden costs or difficult-to-cancel subscriptions. A professional designer uses the wireframing phase to advocate for "Ethical UX," ensuring that the navigation is transparent, the privacy settings are clear, and the user remains in control of their experience.²³

By documenting these ethical considerations in the design rationale, a practitioner proves that they are not just a "pixel pusher" but a responsible professional who understands the long-term impact of their work on both the user and the business's reputation.²³

Instructional Summary for the Researcher/Practitioner

The journey from a blank page to a sophisticated digital product is structured by these core competencies. To "teach" this lesson is to emphasize the *discipline* of planning over the *excitement* of building. The practitioner who spends an extra three hours in the wireframing phase often saves thirty hours in the development phase.⁴ The engineer who writes descriptive, imperative-mood commit messages creates a project history that serves as a masterclass in professional communication.²¹

This holistic approach—combining the skeletal precision of wireframing with the industrial stability of GitHub—is what defines excellence in the modern technology sector. It is a commitment to the idea that great products are built twice: once in the mind (and on the wireframe) and once in the code.

Works cited

1. The Role of Wireframes in Enhancing User Interface Design - IRJIET, accessed March 22, 2026, https://irjiet.com/common_src/article_file/1735115635_21e7dcf654_8_irjiet.pdf
2. Mastering the Art of GitHub Repository Creation: A Step-by-Step Guide - HulkApps, accessed March 22, 2026, <https://www.hulkapps.com/blogs/ecommerce-hub/mastering-the-art-of-github-repository-creation-a-step-by-step-guide>
3. Wireframing - The Decision Lab, accessed March 22, 2026, <https://thedecisionlab.com/reference-guide/design/wireframing>
4. The Power of Low-Fidelity Wireframes: A UX Foundation for Faster, Smarter Design, accessed March 22, 2026, <https://wireframes.org/low-fidelity-wireframes>
5. Low Fidelity Prototyping: 2026 Guide - Parallel Design Studio, accessed March 22, 2026, <https://www.parallelhq.com/blog/what-low-fidelity-prototype>
6. Create Low to High-Fidelity Prototypes: Essential UX Guide - Visily, accessed March 22, 2026, <https://www.visily.ai/blog/creating-low-high-fidelity-prototypes/>
7. The Best Wireframing Tools to Design Faster and Smarter - ZipWP, accessed

- March 22, 2026, <https://zipwp.com/resources/wireframing-tools/>
8. 5 Figma alternatives for wireframing & UI design (2025 Guide), accessed March 22, 2026, <https://balsamiq.com/blog/figma-alternatives/>
 9. 11 best forever free wireframing tools for designers (2022) - Clique Studios, accessed March 22, 2026, <https://cliquestudios.com/clique-university/free-wireframing-tools>
 10. Best Free Wireframing Tools in 2025 | Frame0 Blog, accessed March 22, 2026, <https://frame0.app/blog/posts/2025-08-15-best-free-wireframing-tools/>
 11. How to create mobile app wireframes (in just 8 steps) - Balsamiq, accessed March 22, 2026, <https://balsamiq.com/blog/mobile-app-wireframing-guide/>
 12. How To Create Mobile App Wireframes in 7 Easy Steps - Appingine, accessed March 22, 2026, <https://www.appingine.com/blog/mobile-app-wireframing>
 13. How to Create Mobile App Wireframes (Step-by-Step) - MockFlow, accessed March 22, 2026, <https://mockflow.com/blog/wireframing-for-mobile-apps>
 14. Step-by-Step Guide to Creating an App Wireframe | by R.Vasanth Kumar | Feb, 2026, accessed March 22, 2026, <https://vasanthkumarrengan.medium.com/step-by-step-guide-to-creating-an-app-wireframe-1c7c9cbe4951>
 15. Quickstart for repositories - GitHub Docs, accessed March 22, 2026, <https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories>
 16. A Guide to Git and Github - Launch School, accessed March 22, 2026, <https://launchschool.com/books/git/read/github>
 17. Using GitHub as a Portfolio When Applying for Jobs · community · Discussion #169760, accessed March 22, 2026, <https://github.com/orgs/community/discussions/169760>
 18. Best practices for repositories - GitHub Docs, accessed March 22, 2026, <https://docs.github.com/en/repositories/creating-and-managing-repositories/best-practices-for-repositories>
 19. Creating your first repository using GitHub Desktop, accessed March 22, 2026, <https://docs.github.com/en/desktop/overview/creating-your-first-repository-using-github-desktop>
 20. What's the best way to embed pictures in GitHub README files ..., accessed March 22, 2026, <https://community.latenode.com/t/whats-the-best-way-to-embed-pictures-in-github-readme-files/32408>
 21. How to Write a Good Git Commit Message | Git Best Practices, accessed March 22, 2026, <https://www.gitkraken.com/learn/git/best-practices/git-commit-message>
 22. How To Add Images in Markdown (Syntax, Examples & Pro Tips) | DigitalOcean, accessed March 22, 2026, <https://www.digitalocean.com/community/tutorials/markdown-markdown-images>
 23. Design Decisions - How to Make and Justify Them | Guide - UXCam, accessed March 22, 2026, <https://uxcam.com/blog/design-decisions/>

24. 3 Easy Ways to Show Design Decisions in Your UX Portfolio, accessed March 22, 2026, <https://blog.uxfol.io/design-decisions/>
25. 5 Ways to Build a Strong GitHub Portfolio as a Junior Developer | by TekCanon - Medium, accessed March 22, 2026, <https://medium.com/@tekcanon/5-ways-to-build-a-strong-github-portfolio-as-a-junior-developer-18f33ac94375>
26. Best Practices for Hiring Skilled Software Developers for GitHub-Based Projects · community · Discussion #183788, accessed March 22, 2026, <https://github.com/orgs/community/discussions/183788>
27. Best practices for organizing repository files · community · Discussion #173482 - GitHub, accessed March 22, 2026, <https://github.com/orgs/community/discussions/173482>
28. IxDF Indore Online Meet - Dec 1, 2024, accessed March 22, 2026, <https://ixdf.org/meetups/ixdf-indore-online-meet-dec-1st-2024>
29. IxDF Indore - Page 3 | UX/UI Design Community: Local Jobs ..., accessed March 22, 2026, <https://www.interaction-design.org/local-group/asia/india/indore?page=3>
30. IxDF Indore - Page 4 | UX/UI Design Community: Local Jobs, Meetups and Networking, accessed March 22, 2026, <https://www.interaction-design.org/local-group/asia/india/indore?page=4>
31. 15+ Best Wireframe Tools for Visual Designers (2025) - Venngage, accessed March 22, 2026, <https://venngage.com/blog/best-wireframe-tools/>
32. Design Rationale Documentation - Pencil & Paper, accessed March 22, 2026, <https://www.pencilandpaper.io/articles/design-rationale-documentation>
33. Style guide - GitHub Docs, accessed March 22, 2026, <https://docs.github.com/en/contributing/style-guide-and-content-model/style-guide>
34. markdown - How to add images to README.md on GitHub? - Stack Overflow, accessed March 22, 2026, <https://stackoverflow.com/questions/14494747/how-to-add-images-to-readme-md-on-github/77062284>
35. 4 Ways to Add Images to GitHub README + 1 Bonus Method - Cloudinary, accessed March 22, 2026, <https://cloudinary.com/guides/web-performance/4-ways-to-add-images-to-github-readme-1-bonus-method>
36. Top Interview Questions and Answers for Wireframing - HelloIntern, accessed March 22, 2026, <https://hellointern.in/blog/top-interview-questions-and-answers-for-wireframing>
37. UI/UX Designer interview questions - Micro1, accessed March 22, 2026, <https://www.micro1.ai/interview-prep/ui-ux-designer-interview-questions>
38. The 25 Most Common UI/UX Designers Interview Questions - Final Round AI, accessed March 22, 2026, <https://www.finalroundai.com/blog/ui-ux-designer-interview-questions>
39. How to Write UX/UI Design Case Studies That Boost Your Portfolio and Get You Hired - IxDF, accessed March 22, 2026,

<https://ixdf.org/literature/article/how-to-write-great-case-studies-for-your-ux-design-portfolio>