

Project Report

Design Choices

Simplicity & clarity: I structured the project to keep the main entry simple (single `index.html`) and grouped static assets under `assets/` (images, models, textures, scripts). This lowers cognitive load and makes it straightforward to host or export.

Separation of concerns: Visual assets, models, and scripts are separated into their own folders so designers and developers can work independently.

Portable file formats: I favored common web-friendly formats (PNG/JPEG for images, glTF/OBJ for models where applicable) to maximize compatibility across browsers and tools.

Technical Challenges & Resolutions

Asset organization at scale: As projects grow, assets become difficult to find. I applied a predictable folder structure (`assets/images`, `assets/models`, `assets/textures`, `assets/scripts`) and consistent naming to make discovery and automation easy.

Cross-origin and hosting concerns: Serving local model and texture files in browsers can hit CORS or local-file restrictions. Recommendation: host with a simple local HTTP server for development (e.g., `python -m http.server`) or use a lightweight static hosting service for deployment.

Performance when loading models/textures: Large models or high-res textures slow load times. I recommend lazy-loading, LOD (level-of-detail) models, and compressed texture variants (KTX2/ETC2) in future iterations.

Maintaining reproducible builds: Different team members may have different environment setups. I recommend adding a short `README.md` with dev commands and pinned tool versions, or provide a small `package.json` / `requirements.txt` for tooling.

Future Improvements

Automated build pipeline: Add a build step to optimize textures, compress models, and generate hashed filenames for caching.

Documentation & examples: Expand the repo with usage examples, a `README.md` and sample scene files showing how to load models and textures.

Accessibility & progressive enhancement: Ensure UI/UX works across devices and low-powered hardware; provide fallback low-resolution assets.

Add furnitures to the upper part of the building: Enhance the upper floors with detailed furniture models and textures to create a more complete and immersive interior environment.

Testing & CI: Add automated checks to validate that referenced assets exist and are not

broken links.