



# Design Automation Challenge

## Background

Design teams often spend significant time creating new user interface mockups and marketing visuals to match an existing style or brand. Whenever a new feature or user flow is proposed, designers must manually produce screens and graphics consistent with their current design language. This process is time-consuming and slows down iteration. There is a need for a smart tool that can accelerate this process by learning from existing designs and automatically generating new design assets in the same style.

## Objective

Build an AI-powered design assistant that can generate new design mockups and visual materials based on a set of provided existing designs and reference documents. The goal is to enable rapid prototyping of user flows and creation of marketing graphics that seamlessly fit into an existing product's look and feel. Participants should develop an application that lets users upload their current design assets (e.g. UI screenshots, style guides, brand images) along with any relevant documents (like requirements or guidelines) and then generate new design outputs on demand using that input.

# Key Features & Requirements

- **Design Asset Ingestion:** Allow users to import existing design files, images, or style guides from their own projects. The AI should analyze these to understand the visual style (layouts, color schemes, typography, etc.) and any branding cues.
- **Knowledge Integration:** Accept and **store** additional documents or context (e.g., feature specifications, UX guidelines, or domain knowledge articles) to guide the generation. Beyond informing the AI about purpose and constraints, use these materials to **validate and correct** outputs as needed.
- **Prompt-Based Generation:** Provide an interface for the user to describe the new feature or asset they need. For example, a user might input: “Generate a new user onboarding flow with three screens, following the style of our existing app,” or “Create a marketing banner for Feature X using the same branding as our current campaign.” The tool should interpret this request.
- **AI-Driven Output:** Using the learned style and the prompt, generate the new design mockups or graphics. The output could be UI screen layouts, user flow diagrams, marketing banners, or other visual materials as requested. The generated designs **must** closely align with the style of the provided examples (so that they look like they belong to the same product or brand). Additionally, all results must be exportable/downloadable in common formats.
- **Editing & Refinement (Bonus):** Ideally, users can tweak or refine the results. For instance, the app could allow users to regenerate with adjustments or edit the AI-generated design using an integrated editor. This is optional but can add value to the solution.

Teams are free to choose how the AI does the generation. You might use image generation models for creating new visuals, large language models to interpret requests or generate design specs, or a combination of techniques. The focus should be on **consistency** (the new designs should match the given style) and **speed** (it should be much faster than designing from scratch).

# Judging Criteria

Projects will be evaluated by the judges based on several key criteria. Each criterion will be scored to assess the overall quality of your solution:

Criteria	Description
<b>Originality &amp; Innovation</b>	How creative and innovative is the idea? Does the solution approach the problem in a novel way?
<b>Technical Implementation</b>	Quality of the technical solution. Is AI effectively used to analyze designs and generate outputs? Is the application functional and well-built?
<b>Design Quality &amp; Consistency</b>	Are the generated outputs visually coherent and truly reflective of the provided style/brand? Do they demonstrate good UX/UI practices?
<b>Impact &amp; Usefulness</b>	How useful would this tool be for designers or teams in real-world scenarios? Does it significantly speed up or improve the design process?
<b>Presentation &amp; Completeness</b>	How well did the team present their project? Is the solution complete or at least a compelling prototype? (Includes clarity of demo and documentation.)

**Judges will assign points for each category above.** Focus on delivering a working prototype that shows how quickly and reliably your AI assistant turns natural-language inputs into clear, consistent process diagrams and practical wireframe previews, with support for revisions and alternatives. Good luck, and have fun product flows with your AI collaborator!

# Project Submission

Setup and access: follow [hackathon-checklist.pdf](#).

Clone time: Monday, 08.12.2026, 13:00 MEZ. Commits after this time will **not** be considered.

Branch: the **default branch** must contain the final version.

Folder **/result/**: commit and push all outputs your app generated for the example prompts.

# Example Prompts

## DQAI (Data Quality AI)

### OCC Context

OCC manages **projects** that contain one or more **systems**. Work is performed **within a project and against its systems**. OCC itself has **no direct SAP access**. SAP admins use a **SAP plugin** to export tables/data to an **export handling app**, which reports transfer status back to OCC. A **mediator app** provisions the Azure target machine, installs and configures **JiVS IMP** on a customer-managed machine, and prepares the export handling app for incoming transfers. **JiVS IMP** (web app on the customer machine) can run analyses and expose results that OCC can surface.

### New Feature

Within a selected project, simulate the lifecycle where SAP systems are first associated with the project, an installation is initiated for the target environment, and during this phase, a simulated progress experience is represented. After the environment is “ready,” present a data transfer view in which transfer items transition from Pending to Success. When transfers have completed, show a results experience that compares metrics over time, including at least one trend chart and a run history (repeatable analyses). All data, statuses, notifications, and timings are synthetic (no real integrations). The flow should reflect the roles of the export handling app, mediator, IMP (including the notion of IMP views), and OCC’s ability to surface analysis outcomes without prescribing a fixed sequence of UI steps.

## Carveout

### OCC Context

OCC manages **projects** that contain one or more **systems**. Work is performed **within a project and against its systems**. OCC itself has **no direct SAP access**. SAP admins use a **SAP plugin** to export tables/data to an **export handling app**, which reports transfer status back to OCC. A **mediator app** provisions the Azure target machine, installs and configures **JiVS IMP** on a customer-managed machine, and prepares the export handling app for incoming transfers. **JiVS IMP** (web app on the customer machine) can run analyses and expose results that OCC can surface.

### Case - Carveout:

Within a selected project, simulate the lifecycle where SAP systems are first associated with the project, an installation is initiated for the target environment, and during this phase, a simulated progress experience is represented. After the environment is “ready,” present a data transfer view in which transfer items transition from Pending to Success. When transfers have completed, show a Carveout experience that defines a filtered subset (e.g., organizational units, time slices), previews/validates the subset (record counts, affected entities), and optionally indicates readiness to forward the carved data to a downstream target. Include at least one trend chart (e.g., carved volume over time) and a simple run history of carve operations. All data, statuses, notifications, and timings are synthetic (no real integrations). The flow should reflect the roles of the export handling app, mediator, IMP (including the notion of IMP views), and OCC’s ability to surface Carveout results without prescribing a fixed sequence of UI steps.