Orchids SWE Internship Take Home - Website Cloning

Background

At Orchids, we are building the best agentic AI for website design and development. One essential tool that we have implemented at Orchids to achieve this is website cloning - the ability for an AI agent to clone and capture the aesthetics of any given website. In this challenge, you will be building out a minimal version of our website cloning feature.

Implementation Guideline

You will be building a web app that will take in any given public website URL, scrape the website URL for useful design context and give it to an LLM model to replicate the website in HTML. The goal is that the resulting HTML website should be as aesthetically similar as possible to the given website at the given URL. Please follow this guideline carefully to complete the challenge.

Codebase (Link to template)

You will be working off a template codebase, which is a simple template for a Next.js + Typescript frontend and a Python FastAPI backend. Please go to ***this Google Drive link*** to download the template codebase. There will be instructions on how to set it up in the README.md file.

Pipeline

1. **Web app**: A web app that exposes an input field for the website URL, which the user can type in a public website URL link and get back a preview of the the cloned website's HTML after the cloning process is complete

2. **Website Scraping**: A website scraping process that can visit websites efficiently and extract useful design context. This can be image snapshots, DOM structure, assets, stylesheets, etc. Anything that you think might be useful for an LLM to take in and clone a website well.

3. **LLM cloning**: Develop an LLM workflow that can produce a HTML website clone based on the provided design context. Try to experiment with different models, system prompts, reasoning chains, agentic architectures to get the best result possible.

Requirements

1. The primary stack that you should be working with should be what the template codebase provides (Next.js + Typescript frontend and Python + FastAPI backend). However, you are

free to expand this stack with additional technologies (libraries, databases, storage, micro-services, external APIs, etc).

2. The website scraping process should be reliable. Websites can have firewalls, IP blocks, slow load times, etc that will make the scraping process unreliable. At the same time, if you are considering spinning up browsers for this problem, be strategic about where those browser instances live. Spinning up local browsers can be a slow and expensive process in production settings. You can consider cloud hosted solutions such as Browserbase or Hyperbrowser.

3. Providing great design context for the LLM and having an efficient LLM workflow to produce cloned websites is crucial. While 100% precision in cloning websites is not a requirement, getting as close as possible is ideal. A great starting tip would be to use a base LLM that is best at writing UI code (Claude 4 Sonnet, Gemini 2.5 Pro).

Deliverable

There are 2 things needed for a submission:

- A zip file of your complete codebase. If it requires extra set up steps, please include it in a README.md file
- A short video of you walking through your submission and showing it working.

SUBMISSION (DUE 6/6, 11:59 PM PST)

- Please submit here

That is all from us! Thank you for your interest in Orchids and please let us know if you have any questions by emailing us at kevinlu@orchids.app