

Full-Stack MedTech Mini Web-App

Context

You are asked to contribute to a MedTech project focused on surgical planning. The purpose of this assignment is to evaluate your ability to develop a full-stack mini web-app, implementing a complete frontend/backend workflow with a simple biomedical image processing task.

Note: the required image processing steps are simulations only and have no clinical purpose. If the project or its requirements cannot be fully completed due to time constraints, please provide the code along with comments explaining what remains to be done.

Objective

The web app must allow a user to:

1. Upload a sample 2D medical image (A) (common formats such as JPG/PNG).
2. Select an image phase to be simulated:
 - o Arterial phase → the image should appear with increased contrast (B).
 - o Venous phase → the image should appear with a gaussian smoothing (C).
3. Display the original and processed images side by side (e.g. (A) on the left, (B) or (C) on the right).

Image processing (2.) must be executed entirely on the backend using Python, with no operations performed in the browser. The frontend will only handle user input, send requests to the backend, and display the results. The backend responsible for image processing must run on a dedicated external server (e.g., a Hugging Face Space), separate from the one hosting the frontend (e.g., a static host such as GitHub Pages).

Instructions

You will have **4 hours** to complete the task and commit the final project materials to a GitHub repository. The countdown will begin when you formally received the assignment via email and will close exactly 4 hours thereafter.

The official submission time (it could also be less than 4 hours) will be verified based on the timestamp of the last push to the **main** branch of the GitHub repository you provide.

In case the project or some requirements cannot be fully completed within the given timeframe, you could still submit the code with appropriate comments indicating the missing parts and explaining what would remain to be implemented.

Even if the backend code runs on a separate server, it must be included in a backend folder inside the same GitHub repository as the frontend, so that all materials are contained in a single delivered place. The code will be reviewed directly from the repository, while the functionality will be evaluated by accessing the public link of the deployed mini web application (e.g. the URL provided by GitHub Pages).

You are free to set up all required repositories as public.

Minimum Requirements

- Upload and display of the original image.
- Selection of phase (arterial/venous) and submission of the request to the backend.
- Image processing performed on the server and returned to the client.
- Display of the processed image next to the original.
- A working project accessible through a public URL (e.g. GitHub Pages URL).

Deliverables

- Complete codebase for the project (frontend + backend), with a concise README with essential instructions (e.g. what it does, how to use it, link to the demo).
- A working URL to test the mini web-app.

Practical Notes

- You are free to use any free services you prefer.
- Suggestion: Hugging Face Spaces can be used for the backend.
- The frontend can be hosted on GitHub Pages or any equivalent service.

Evaluation Criteria

- Basic functionality (end-to-end workflow)
- Clarity and completeness of the project (clean repo, README, code comments)
- User experience quality (frontend usability and clarity)