**✅ Step-by-Step Workflow**

1. **Word Template**
   * You prepare a .docx file containing *merge-style placeholders* (e.g. {{deceased\_name}}, {{institution}}).
   * This file lives in the case’s upload folder.
2. **Edit Button**
   * User clicks Szerkesztés → opens a new Flask route like /szerkesztes/<case\_id>/<filename>.
3. **Editable Web Form**
   * That route:
     + Loads the .docx template.
     + Extracts the placeholders (field names).
     + Renders a form with text fields for each placeholder.
4. **User Submits Form**
   * Flask receives the submitted values.
   * Fills in the .docx template using python-docx or docxtpl.
   * Saves the result as a new file (e.g. final\_{filename}) or overwrites the original.

**🔧 Tech Needed**

* Python packages:
  + [docxtpl](https://github.com/elapouya/python-docx-template) — best for Jinja-style templates like {{ field }}
  + OR [python-docx](https://python-docx.readthedocs.io/en/latest/) — lower level

**📂 Example Workflow for a Case**

* File: végzés\_template.docx
* Placeholders: {{case\_number}}, {{deceased\_name}}, {{expert\_name}}
* User opens /szerkesztes/17/vegzes\_template.docx
* Sees:
  + Case number: [IKT-17]
  + Name: [John Doe]
  + Expert: [Dr. Kiss]
* Clicks save → végzés\_kitöltve.docx is generated in same folder.

**❓Shall we proceed?**

If yes, I’ll:

1. Write the Codex prompt to add:
   * New route: /szerkesztes/<case\_id>/<filename>
   * Form rendering and template filling logic
2. Use docxtpl and extract fields automatically

Let me know and I’ll generate it.