

Here are the complete front-end pages you need for the Automated Front-End VLSI Design AI Tool using LLM-RAG — aligned exactly with your diagram and matching your repo structure (frontend/src/pages/).

I will give you:

- Page list
- UI flow
- Component usage
- Clean, production-ready React (Vite) file structure
- What each page must contain

- Front-End Pages (Complete)

Your tool needs 4 main pages:

① Home Page (Home.jsx) Purpose

Landing page → Explain what the tool does + navigation to Upload & Generate.

Key UI Sections

Banner: “Automated VLSI Front-End Design using LLM RAG”

System architecture diagram (the one you uploaded)

Buttons:

Upload Specification

Generate RTL

View Outputs

Short description:

“Upload spec → Parse → RAG retrieval → RTL generation → UVM/FV verification → Final IP.”

Components Used

StatusPanel

CodeViewer (for sample preview)

FileUpload (small quick upload widget)

Upload Page (Upload.jsx) Purpose

User uploads spec documents, protocol docs, reference guides, testbench requirements, etc.

Inputs Required

PDF / TXT / DOCX spec upload

Protocol file upload (AXI, PCIe, AMBA etc.)

RTL reference library upload

Requirements form → modules/features

Components Used

- ✓ FileUpload
- ✓ RequirementsForm

Actions

Upload → stored in backend/app/services/file_service.py

Parsed via file_parser.py

Added into knowledge_base/vector_db/

Generate Page (Generate.jsx) Purpose

Main page where user presses Generate RTL.

UI Sections

Pre-processing status: NLP → chunking → parsing

RAG Retrieval Logs: Show which documents were retrieved

LLM Generation Panels:

RTL (Verilog/VHDL)

TB + UVM Env

FSM diagrams

Coverage metrics

Assertions

Post-processing Status:

Lint & Style

CDC checking

STA pre-synthesis

Components Used

StatusPanel

CodeViewer

Buttons

- ✓ Generate RTL
- ✓ Optimize RTL (PPA Optimization)
- ✓ Run Simulation
- ✓ Re-iterate (Feed back waveform logs to LLM)

4 Outputs Page (Outputs.jsx) Purpose

Displays all results from backend.

Outputs Shown

RTL IP block files

Testbench

UVM Environment

Coverage reports

Waveforms

Logs

Regression results

Timing diagrams

VIP modules

Components Used

CodeViewer

StatusPanel

Downloadable Files

rtl/*.v

uvm_env/

coverage.json

waveforms.vcd

report.txt

🎯 Recommended Folder Structure frontend

```
frontend/
  src/
    main.jsx
    App.jsx
  pages/
    Home.jsx
    Upload.jsx
    Generate.jsx
    Outputs.jsx
  components/
    FileUpload.jsx
    StatusPanel.jsx
    CodeViewer.jsx
    RequirementsForm.jsx
```

Automated front end VLSI design AI tool using LLM RAG

