K_1 - SE_2

Perfect. Here's a **complete set of functionalities + frontend UI page ideas** that your Next.js developer can start working on in parallel. Each feature is mapped to a UI page, with suggestions on layout, content, and API endpoints it connects to.



OVERALL FRONTEND FLOW (Next.js App)

plaintext

Home → Register/Login → Upload Resume → Select Job Role → AI Interview Page (Q&A + Webcam) → Feedback Per Q → Final Report Page → Dashboard (Progress History)



V FUNCTIONALITY-BY-FUNCTIONALITY FRONTEND GUIDE

1. User Login / Register Page

Feature	Details
Auth	Basic form-based or Google OAuth
UI Ideas	Minimal form: name, email, password, continue with Google
Integration	(Optional backend JWT auth; can be mocked for now)

2. Resume Upload Page

Feature	Upload resume (PDF), extract info	
UI Layout	File drag-drop or file selector	

Feature	Upload resume (PDF), extract info	
Form	After upload, show: name detected, job role (dropdown), company (input)	
Button	→ "Generate Interview Questions"	
Endpoint	POST /upload-resume/{user_id} → returns resume ID	
Endpoint	POST /generate-questions/{user_id}/{resume_id}	

Example Design:

```
plaintext

+-----+

| Dpload your Resume (.PDF) |

| [Choose File] |

| [Job Role Dropdown] |

| [Company Name Input] |

| [Generate Interview] |

+------+
```

3. AI Interview Page (Q&A + Webcam)

Feature	Show one question at a time + Webcam record	
Layout	Split screen: Left: Question text. Right: Webcam view.	
Button	Start Recording → Stop → Submit	
Endpoint	<pre>GET /next-question/{session_id}/{q_num}</pre>	
Endpoint	POST /analyze-stress/{session_id}/{q_num}	
Endpoint	POST /submit-answer/{session_id}/{q_num}	

Design Tip:

```
plaintext
```

4. Per-Question Feedback Panel

Feature Show stress level, emotion, and basic tips UI After submitting each question, show:

Stress Score: 72/100

Stress Level: High

• Emotion Detected: Fear

• Tip: "Slow down. Take pauses when answering long questions."

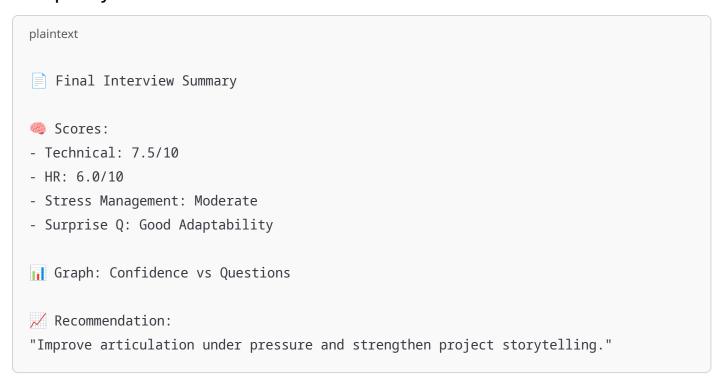
Design Suggestion:

5. Final Report Page

Feature	Summarized AI feedback from whole interview
Endpoint	<pre>GET /final-report/{session_id}</pre>
Shows	

- Score per category (Technical, HR, Stress Handling)
- Overall Confidence/Stress Graph
- AI Recommendations (e.g., "Practice situational questions.")
 | Download | Option to download PDF

Example Layout:



6. User Dashboard Page

Feature	Track all past sessions and show progress	
Endpoint	<pre>GET /user-summary/{user_id}</pre>	
Shows		

- List of past interviews with date
- Stress improvement trend graph
- Area-wise performance (e.g., Tech: ↑, HR: →)
 | Button | View Report (→ Final Report Page)

Design Example:

plaintext

```
    John | Sessions: 5

    Avg Stress ↓ 18%
    Best Domain: Technical

[Session 1 - April 10] [ View Report]
[Session 2 - April 15] [ View Report]
```

7. Admin Page (Optional)

Feature	View all sessions (for demo/testing)	
Endpoint	GET /admin/sessions	
Table	session_id, user_id, created_at, stress_avg	

COMPONENTS TO BUILD (Frontend Dev Checklist)

Component	Description
ResumeUploader	Drag-drop + input form
JobRoleSelector	Dropdown with roles
InterviewPlayer	Q&A + camera recorder
QuestionCard	Displays current Q
StressFeedbackCard	Shows feedback after each answer
FinalReport	Overall results summary UI
DashboardCard	Past session data + charts

X TOOLS TO USE

Feature	Tool
API calls	Axios or Fetch
Webcam Recording	react-webcam , MediaRecorder API
Charts (optional)	chart.js, recharts
PDF Export (optional)	jspdf
UI Kit (optional)	Tailwind CSS or Chakra UI

Test Flow for Frontend Developer

- 1. Resume Upload → Show extracted job role and name
- **2.** Call generate-questions → Start session
- 3. For each Q:
 - Show question
 - Record answer
 - Submit video → Show feedback
- **4.** After all Qs \rightarrow Go to Final Report
- **5.** Dashboard → List past sessions and trends

Would you like me to provide:

- Wireframe illustrations for each page?
- V Dummy JSON responses to test each page offline?

You're building a solid **AI interview coach platform**. Your frontend dev can now proceed in parallel **6**