

Frontend Mistakes & Corrections Table				
Category	Mistake	Why It Fails	Correct Fix	Never Do This Again
Frontend	Using wrong field names (full_name, job_id)	Spec requires camelCase	{"fullName": "John", "jobApplied": "JOB-1"}	Don't invent field names or use snake_case
Frontend	Sending files in JSON (Content-Type: application/json)	Backend expects multipart/form-data	const f=new FormData();f.append("file",file);	Never upload a file with JSON body
Frontend	Using wrong file field name in public apply (file instead of resumeFile)	Public apply endpoint expects resumeFile	form.append("resumeFile", file);	Don't mix admin/public upload fields
Frontend	Using wrong file field for admin upload (resumeFile instead of file)	Admin upload expects file	form.append("file", file);	Don't use public field names in admin flows
Frontend	Calling admin endpoints with candidate token	RBAC denies access	Use admin/recruiter token	Don't test protected endpoints with wrong role
Frontend	Not handling required fields (missing email, jobId)	Backend validation rejects request	Always send required fields	Don't assume backend will fill defaults
Frontend	Not polling parse task properly	Resume parsing workflow needs polling	Poll /v1/tasks/parse-resume/{taskId} every 2–3 sec	Don't rely on immediate parse response
Frontend	Not using meta.total and pagination	List endpoints rely on paging	Always respect page, limit, meta	Don't fetch all candidates at once
Frontend	Sending wrong Content-Type for bulk upload	Must be multipart	form.append("file", excelFile)	Don't send CSV/XLSX in JSON

Backend Mistakes & Corrections Table				
Category	Mistake	Why It Fails	Correct Fix	Never Do This Again
Backend	Not validating required fields	Causes invalid data + 500 errors	if(!email) errors.email="required";	Don't accept incomplete payloads
Backend	Not enforcing email uniqueness	Creates duplicate candidates	if(exists) throw conflict(...)	Don't skip DB uniqueness check
Backend	Accepting wrong field names	Breaks data consistency	Enforce camelCase only	Don't map or guess field names
Backend	Not updating parse task status	Frontend sees queued forever	Update task: in_progress → success/failed	Don't leave tasks stuck
Backend	Wrong file validation (no size/type check)	Allows invalid/unreadable files	Check MIME type + size < 10MB	Don't accept unsafe uploads
Backend	Returning partial responses (missing parseTaskId)	Frontend parse workflow breaks	Always return ALL fields in spec	Don't modify response schemas
Backend	Using wrong error structure	UI cannot show messages properly	Always return: { error: { code, message, fields } }	Don't return raw DB errors
Backend	Allowing candidate role to access admin endpoints	Security issue	Enforce RBAC in middleware	Don't skip role checks
Backend	Not creating tasks/rows for parsing	Worker cannot find task	Insert task row BEFORE processing	Don't parse without proper task tracking
Backend	Bulk upload worker not running	Always stuck in "queued"	Ensure worker processes job queue	Don't leave jobs unprocessed
Backend	Incorrect delete/restore resume logic	Deletes active resume → undefined behavior	Check rule: Cannot delete last active resume	Don't delete active version without force flag