

Stage 2 (Practical) — CSV Contact Importer

Implement, using Laravel and Vue.js, a small feature to import contacts from a CSV file and display a processing summary.

Requirements:

- Laravel and Vue.js stack.
- Think of this project as something international — everything must be in English.
- CSV upload (via web). The CSV will contain at least the following columns: name, email, phone, birthdate. Column order may vary, and there will be cases with invalid rows. Along with this file, you will receive two CSV files: a small one (~50 lines) and a large one (~3 thousand lines). The system should work with any CSV file, not only with those provided in this test
- File processing: deduplicate by email, handle invalid rows, and keep count of results.
- Summary with: total rows read, successfully imported, ignored due to duplicates, and ignored due to validation errors.
- Simple listing of imported contacts with pagination.
- For each imported contact, display a profile picture from Gravatar based on the email address (use identicon as fallback).
- Deliver the project in a public Git repository with an execution README.

You have full freedom to organize the code as you see fit. Focus on delivering something functional, readable, and easy to run, with the best possible end-user experience. The use of AI (Artificial Intelligence) is allowed and encouraged, as long as you can explain and justify each piece of code in your application. Feel free to reach out to your engineer recruiter with any questions you may have.

Suggested duration: 4 to 6 hours of total work (delivery deadline: 48 hours).

Evaluation Criteria:

- Functional correctness (35%): Upload, summary, deduplication, and validations work correctly.
- Robustness and care (25%): Error handling, integrity, and consistency.
- Clarity and readability (20%): Code is easy to understand and maintain, with clear structure and commits.
- User experience (10%): Simple flow and clear feedback.
- Initiative and technical maturity (10%): Extra features, technical justifications.

Optional suggestions (ideas only):

Automated tests, asynchronous processing (queue) for large CSVs, progress bar, detailed validations, unique constraint on the database.

Delivery:

Send the link to the public Github repository with a README explaining how to run the project. If there are relevant decisions, include a short DECISIONS.md (max. 20 lines) explaining why you chose approach X instead of Y.