Take-at-home Challenge

Dear candidate,

We're glad you're considering joining Volteras on the mission to organise and leverage data to accelerate the transition to a greener and smarter future.

The main purpose of this take-at-home challenge is to build an API for data generated by Electric Vehicles and a frontend to interact with the data.

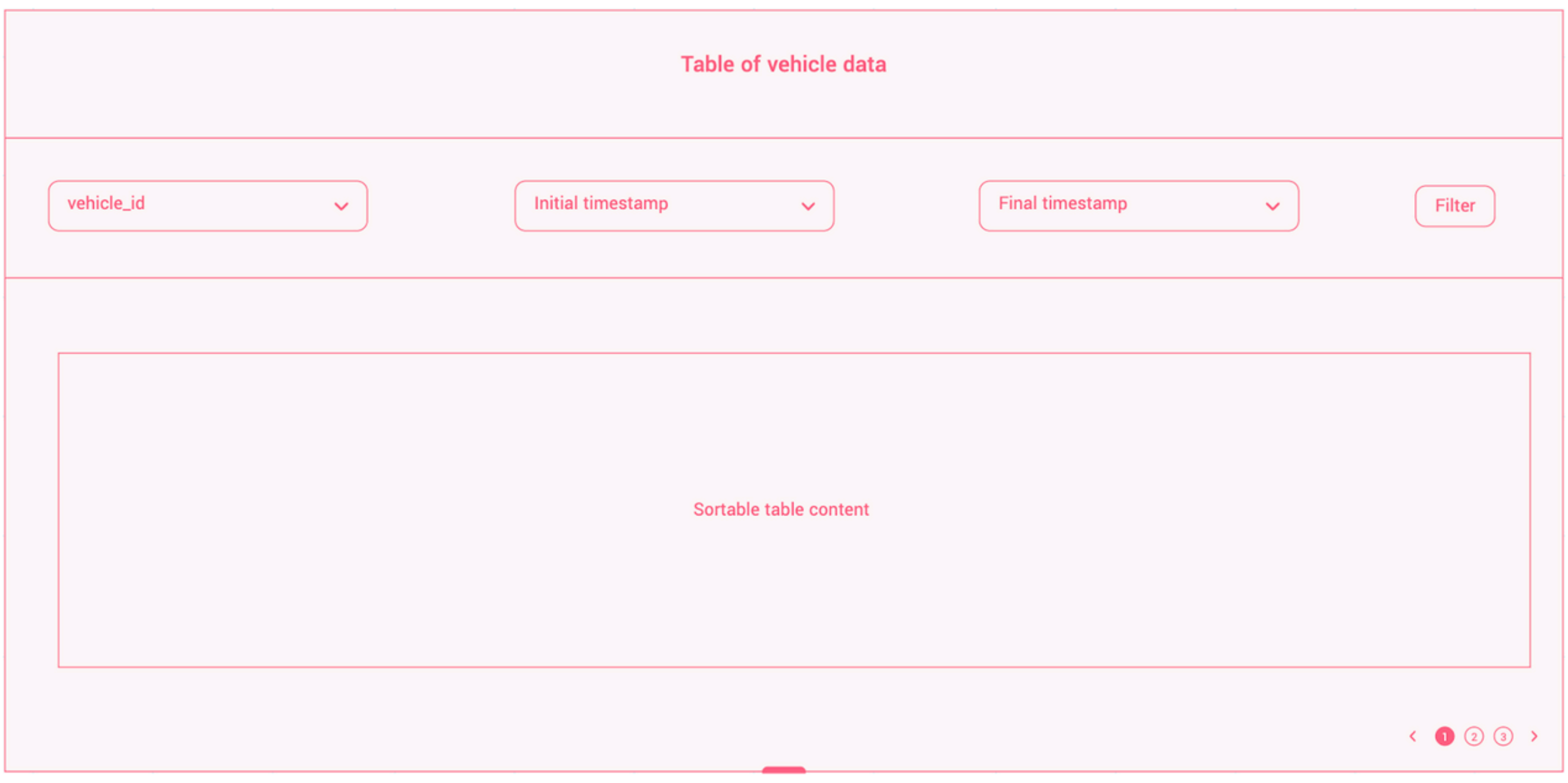
- We expect this task to take **around 4 hours**, you can organise the time invested in the task as you see fit. If you struggle to finish your solution within this time please highlight how far you were able to get within the 4 hours and explain why you made certain decisions or trade offs.
- Once you have completed the task, please send us the relevant information on how to access your solution (e.g. a URL where it's deployed). We would prefer you to use a GitHub repository for your code, but we'll also accept BitBucket and GitLab or a similar solution, this way we can see your commit history and you can easily share the link with us.
- We require solutions to be coded in FastAPI (preferred), Django, or other Python frameworks.
- Please organise, design, test and document your code as if it were going into production.
- You're free to choose additional libraries as you see fit.
- Be sure to quote sources properly.
- Do not share any content of this challenge.

Backend

- The provided zip file (download link here) contains three sample `.csv` files which contain data produced by the vehicles. The naming pattern for the files is: `vehicle_id.csv`. Create a suitable model in the backend to store the data values. Also supply means of loading the sample data into the model / database. A brief description of each column:
  - `timestamp` : ISO8601 date and time string indicating when the data was received.
  - `speed` : Speed in km/h of the vehicle. NULL if the shift state is NULL.
  - `odometer` : Total distance travelled by the vehicle represented in kilometres.
  - `soc` : Abbreviation for “State Of Charge” and informs the remaining percentage of the vehicle’s battery.
  - `elevation` : Elevation of the vehicle represented in metres (considering the sea level)
  - `shift_state` : The state of the shift represented in a single char. NULL if the shift state is neutral or the vehicle is off.
- Implement the following API endpoints which use the model / database:
  - `/api/v1/vehicle_data/` : Retrieve (`GET`) a list of generated data for a given `vehicle_id`, filtered by initial and final timestamps (optional). This endpoint should support pagination, sorting, and a way to limit the number of data returned.
  - `/api/v1/vehicle_data/<id>/` : Retrieve (`GET`) a particular vehicle data by id.

Frontend

Implement a view with functionality according to the following mock-up.



Bonus

- Implement a `POST` endpoint to fill / pre-populate the model / database via an API endpoint.
- Implement an export feature, e.g. to JSON, CSV, or Excel in the API and frontend.
- Plot any vehicle data values in the frontend rather than just displaying them in a table.
- Record a screencast to showcase your solution or point us to a URL where it's deployed.

Evaluation Criteria

- API: created model; implemented endpoints; added error checks for passed parameters; used suitable URL routing; implemented loading of sample data.
- Frontend: implemented table; added support for pagination and sorting to table.
- General: selected and explained chosen software architecture, used clean code structure, followed and explained coding conventions.
- Documentation: added clear inline & high-level documentation, added documentation on how to start & run the submitted solution.
- Tests: written unit tests.
- We evaluate your communication with us.
- *Bonus:* applied design & styling creatively.
- *Bonus:* implemented POST endpoint.
- *Bonus:* implemented export functionality to one of the mentioned target file formats.
- *Bonus:* plotted any vehicle data.
- *Bonus*: recorded a screencast.

All the best and happy coding, if you have any doubt or want any clarification around the task, feel free to reach out!

The Volteras Team