

Mighty Block - RUST Test

Exercise:

You need to develop the backend API for a single page application based on the provided mockup (MightyBlock-Gram.png and UploadPopup.png). The app consists of a feature limited *"Instagram"* clone, with just the option to upload images with a description text associated. Don't worry about managing users. You can suppose there are users provided by another system (hardcode them somewhere and use their id).

How it Works

- Inside the app you can see all the current posts made (with its image, description and upload time) arranged chronologically (from newest to oldest) (MightyBlock-Gram.png).
- It should support pagination of content.
- To create a new post you need to drag the image you want to upload onto the drag area, that will upload the image and show the post popup (UploadPopup.png).
- Once you click the *post* button, the new entry will appear in the main page.
- The app should allow posts to be *liked* and should track *likes* amount. One user should only be able to like a post once.

Tasks

- Design and implement the backend API for supporting all the functionality mentioned.
- Design and implement the DB schema.
- Document both the API and data model in Markdown format delivered in the Readme of the repo.

Bonus Points

- Dockerized for easy dev environment setup (preferably using Docker-Compose) and deploy.
- The whole API should be functional and covered by tests.

Tools/Tech to use

- Use git for version control.
- Backend should be done in RUST
- Use any DB of your choice (justify your selection in your Readme file).
- Any other tech you think it's needed:** You should explain why you chose that tech, instead of other

Deliveries

- You should send the whole project sources (also include .git folder so we can check your commits).
- Documentation (feel free to include diagrams, Swagger files or anything you think is needed).
- A *readme.txt* file explaining anything you think is important about your solution.

All this should be inside a zip file named web_backend_test_exercise.zip

What we will evaluate?

- Code tidiness/source code organization
- Functional aspects of the exercise
- Design and architecture of the solution.
- Use of the requested tech/tools

Contact info

Doubts? Contact us over email:

Gianluca gianluca@mightyblock.co