

Pixowl Web Developer Test

Exercise 1:

Create a static landing page based on the provided photoshop mockup (`VoxEditorMainPage.psd`).

Requirements

- Match pixel perfect the mockup.
- It should use the provided assets taken from the psd file.
- It should use the provided fonts.
- It should allow to easily change the text content from the site (without editing the html code itself).

Bonus Points

- Be prepared to add new sections without much effort.
- Support i18n.
- Smooth scrolling
- Animated transitions
- Responsive Web

Tools/Tech to use

- Sass
- Nunjucks
- gulp
- **Any other tech you think it's needed:** You should explain why you choosed that tech, instead of other.

Deliveries

- You should send the sources of the exercise with it's gulp build scrips in a folder.
- The build output (static site), prepared to be hosted in any static site hosting service inside an `output` directory.
- A *readme.txt* file explaining anything you think is important about your solution (including any instructions needed for building, or explanation on the project structure/tech used.)

All this should be inside a zip file named `web_dev_test_exercise_1.zip`

Exercise 2:

Create a dynamic web app based on the provided mockup (`PixowlIGram.png` and `UploadPopup.png`). The app consists of a very feature limited "*Instagram*" clone, with just the avility to upload images with a description text associattted. It's not needed to handle users or profiles, just a public global shared gallery.

How it Works

- When you enter the app it shows all the current posts made (with its image, description and upload time) arranged chronologically (from newest to oldest).
- 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.

Requirements

- Match the mockups.
- Be completely functional.
- It should be implemented as a `Vue.js` app.

Bonus Points

- Host it somewhere so we could try it live.
- Dockerized for easy dev environment setup (preferably using Docker-Compose).
- Fancy UI design, with cool animations and transitions.

Tools/Tech to use

- `Vue.js`
- **Any other tech you think it's needed:** You should explain why you choosed that tech, instead of other.

Deliveries

- You should send the sources of the exercise.
- Instructions on how to run a local hosted version of the site.
- A *readme.txt* file explaining anything you think is important about your solution.

All this should be inside a zip file named `web_dev_test_exercise_2.zip`

What we will evaluate?

- Code tidyness
- Functional aspects of the exercise
- If design follows the mockups
- Use of the requested tech/tools

Contact info

For any doubts you should contact over email to:

- Gianluca gianluca@pixowl.com
- Lucas lucas@pixowl.com
- Andres andressoe@pixowl.com
- Seb seb@pixowl.com