

# Intetics' Test Task

Position: Junior Full stack developer/ Junior Front-end developer

*"Please do your best to implement the test task parts. The more parts you implement the more points you'll have.*

*To achieve extra points: host your application on one of the cloud host (aws, heroku etc). It's enough to use a free account / plan."*

## Application requirements

- UI language – English
- UI should be user friendly and efficient. You can use sketches from this file.

The test task consists of two parts:

- Front-end part (Client). You can use any JS framework/library or vanilla JS.
- Back-end part (Server). You can use Node.js/Java/C#/PHP and MongoDB/MySQL/MSSQL for work with database.

There are two ways of implementing Intetics' test task:

- As a front-end developer – implement only the client using mock data for back-end part.
- As a full stack developer – implement both parts.

## How to send work results

The results of the work should include:

- the link to a source code of the application (please use GitHub) <sup>required</sup>
- the link to a database ERD diagram (put it to the same folder as the source code) <sup>optional</sup>
- the link to the hosted application <sup>optional</sup>

## Test task description

An image management system with user authorization/registration, to view images, search for images by tags and with the ability to add new images.

### 1. User registration and authorization

The implementation must have simple user registration and authorization. If the user is new he can follow the link to the registration page. After successful registration, the user must be authorized automatically and redirected to the image previewer page.

A user registration page includes the form with the following elements:

- username - required, only a-z or A-Z characters
- e-mail - required, email format
- password - required, at least 6 characters long, at least one lowercase letter, one uppercase letter, one digit, no special symbols eg. ~!@#\$\$%^&\*()\_+;
- retype password - the same as password field
- 'Sign me up' button

The image shows a 'SIGN UP' form. It has a title 'SIGN UP' at the top. Below the title are four input fields: 'Username', 'E-mail', 'Password', and 'Retype password'. At the bottom right of the form is a green button with the text 'SIGN ME UP'.

A user authorization page includes:

- e-mail.
- password
- 'Login' button

### SIGN IN

Email Address

Password

Login

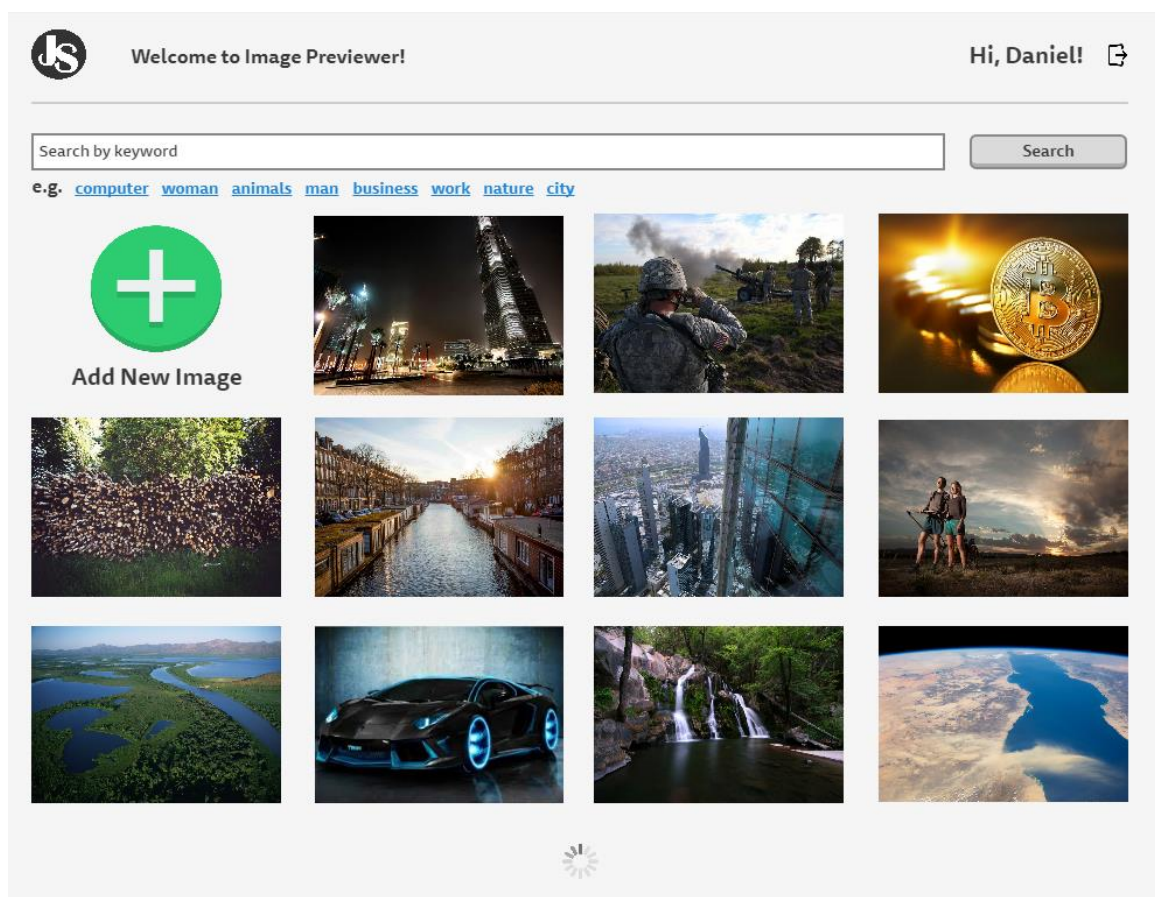
Don't have an account? [Signup](#)

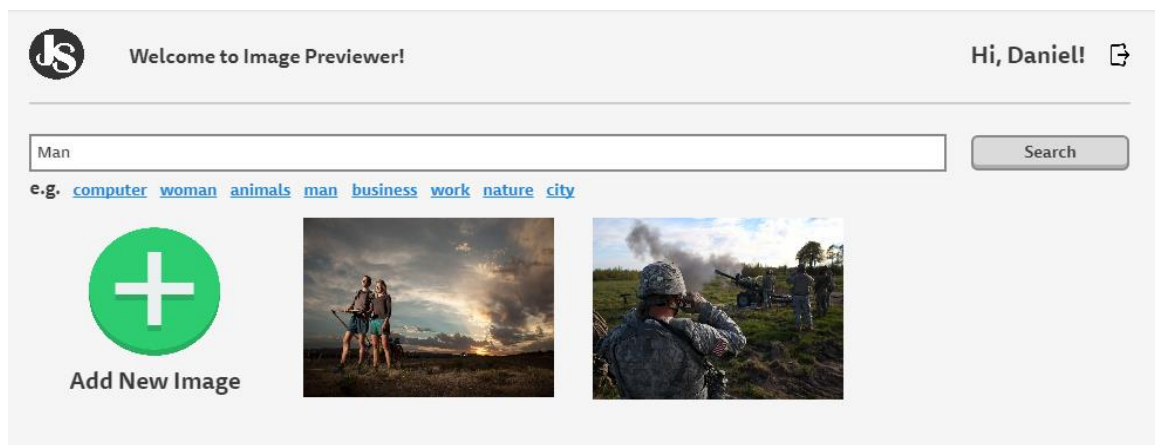
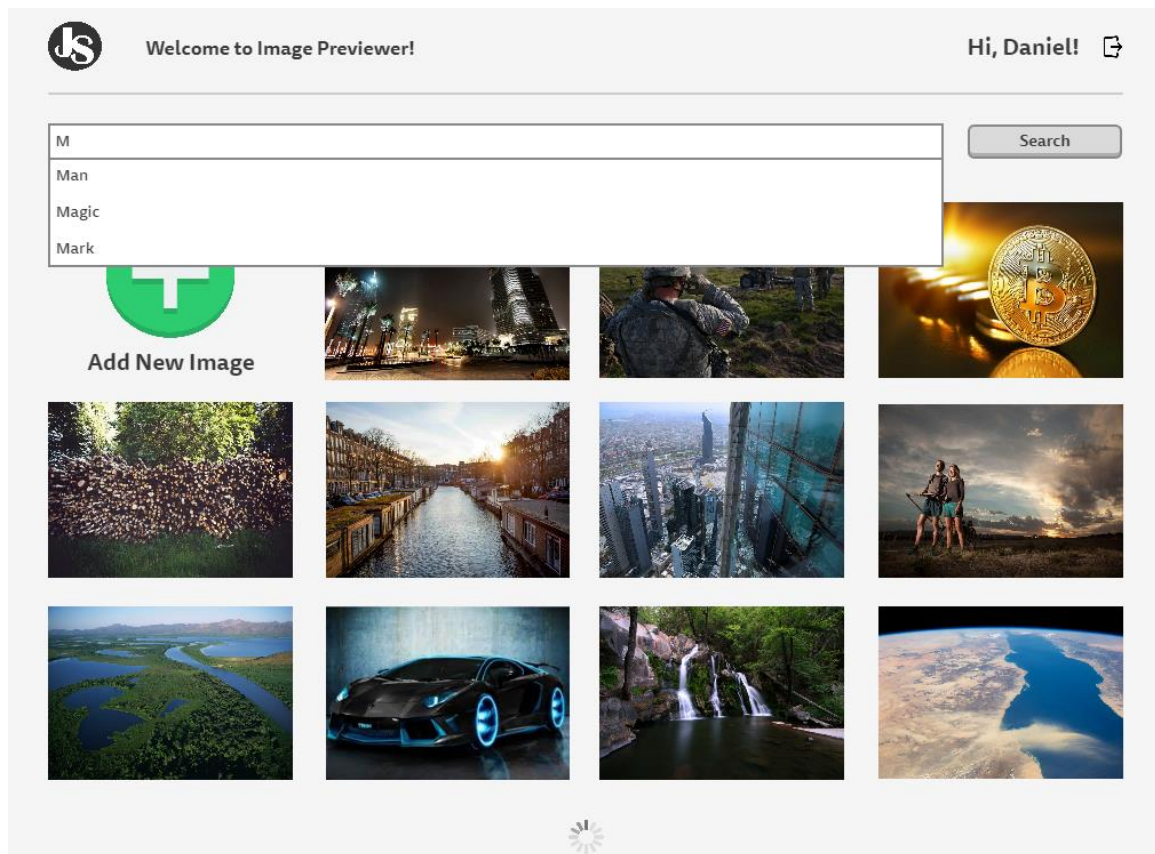
## 2. Image previewer

This page includes the following:

1. Header section must have:
  - logo.
  - application greeting
  - user greeting
  - logout button
2. Search section must have:
  - autocomplete field with button which triggers the search
  - list of popular tags - use a separate rest API endpoint for retrieving these tags from the server
3. Image preview section must have:
  - 'Add New Image' button which opens a modal dialog for adding new images. The button has to be located on the first position of image list (top left).
  - images based on search tags

Image previewer must have pagination based on infinite scroll.





### 3. Modal dialog for adding new images

This dialog includes the following:

- section for uploading images
  - additional information section for the uploading image
  - 'Publish' button
1. The section for uploading images provides us with the ability to preview the uploaded image and two options for adding an image:
    - file open dialog box
    - drag and drop
  2. The additional information section provides the ability to fill name, tags and description of the image. All these fields are required.

To add and remove tags must be implemented as a separate component with the following:

- input field
- 'Add tag' button
- list of the all added tags

The 'Publish' button provides the ability to add the uploaded image to the system for this user.

The screenshot displays a web application titled "Image Previewer" with a user greeting "Hi, Daniel!". The main content area is divided into two panels: "ADD IMAGE" and "ADD INFO".

**ADD IMAGE Panel:**

- Contains a large light gray area with a blue icon of three overlapping photos and the text: "Drag in your media", "Find media on your hard drive, then drag them in to automatically upload."
- At the bottom, there is a button labeled "Manually select media" with a plus icon.

**ADD INFO Panel:**

- Name your media:** A text input field.
- Add tags:** A container showing two existing tags, "Man" and "War", each with a close button (X). Below them is a text input field labeled "Write a tag..." and an "Add tag" button.
- Add description:** A large text area for writing a description.
- Publish:** A prominent green button at the bottom right of the panel.

The background of the application shows a grid of various landscape images. A loading spinner is visible at the bottom center of the page.