Тестовые задания

Требования к реализации:

1) Не приступайте к тестовому заданию, не выслав нам заранее ваше резюме!

2) Тестовое задание подразумевает выполнение как минимум 3-х из 5-и заданий на ваш выбор

3) Все тестовые задания выполняются в рамках одного приложения Rails, репозиторий которого необходимо выставить либо на github либо на bitbucket.

Technical requirements:

- All tasks should be performed in the format pull requests.
- Each branch should be named as VC<id >_<name of branch>
- All commits to the branch should be merged through the squash. (one pull request on create should have only one commit)
- Commit message should contain a clear explanation and begin with #VC<id of issue>.
- green rubocop
- dont use scaffolds
- on create of pull request please give the working time
- Any specific questions you can ask via mail
- Frontend on Angular, backend on Rails API on json served as one app together
- 100% test coverage - Rspec
- CSS – Less
- Javascript – coffeescript
- API – json
- Ruby unit test - RSpec (Capybara)

Задание № 1 Feedback module

Story behind:

> As a Customer with problems using our site I want to have an ability to reach out to raise awareness about my problem in order to inform development team about it.

Task:

- Write rspec+capybara test to:
  - visit main page of the site
  - click link feedback on the page
  - input
    - email
    - text
  - click submit
  - check if message that feedback been sent appears
  - check that form is empty and inactive
- API - Rails, FrontEnd - Angular

Задание № 2 Trello wrapper

Write and test wrapper which will use Trello API to fit following flow of usage

```
> board = TrelloBoard.new(boardid: "someid")
> board.create_card!(title: "sometitle", description: "somedescription", list_id: "SomeID")
```

- Trello credentials should be read as yaml from `$HOME/.trellorc` with format below
- Assumed called from Rails console
- After last command on the board with ID someid card with title and description should appear
- Nothing other than rest-client is allowed to use for the task (ruby, rspec, rest-client, yaml, json)
- wrapper should reside in `lib`
- wrapper should be divided into meaningful classes

```
---
developer_public_key: <key>
member_token: <token>
```

Задание № 3 Rake Task and Sidekiq Worker

Write rake task around class which will:

- query Rubygems API for all the versions of Rails gem
- extract versions from the response
- Iterate over versions
- Search local database table for version string
  - if found:
    enqueue to sidekiq:
    - check if file is found by `gem_copy` path - (if file does not exist - raise error)
    - check if file's sha match given by API response (if SHA does not match - raise error)
  - if missing - enqueue to sidekiq:

- Job should download that version
- store it somewhere in the filesystem under public folder
- Write record to DB with that version and file location and sha

id	version	gem_copy	sha
1	3.2.24	/User/foo/bar/baz/rails-3.2.19.gem	e3b0c44298fc1c1...

Задание № 4 Template

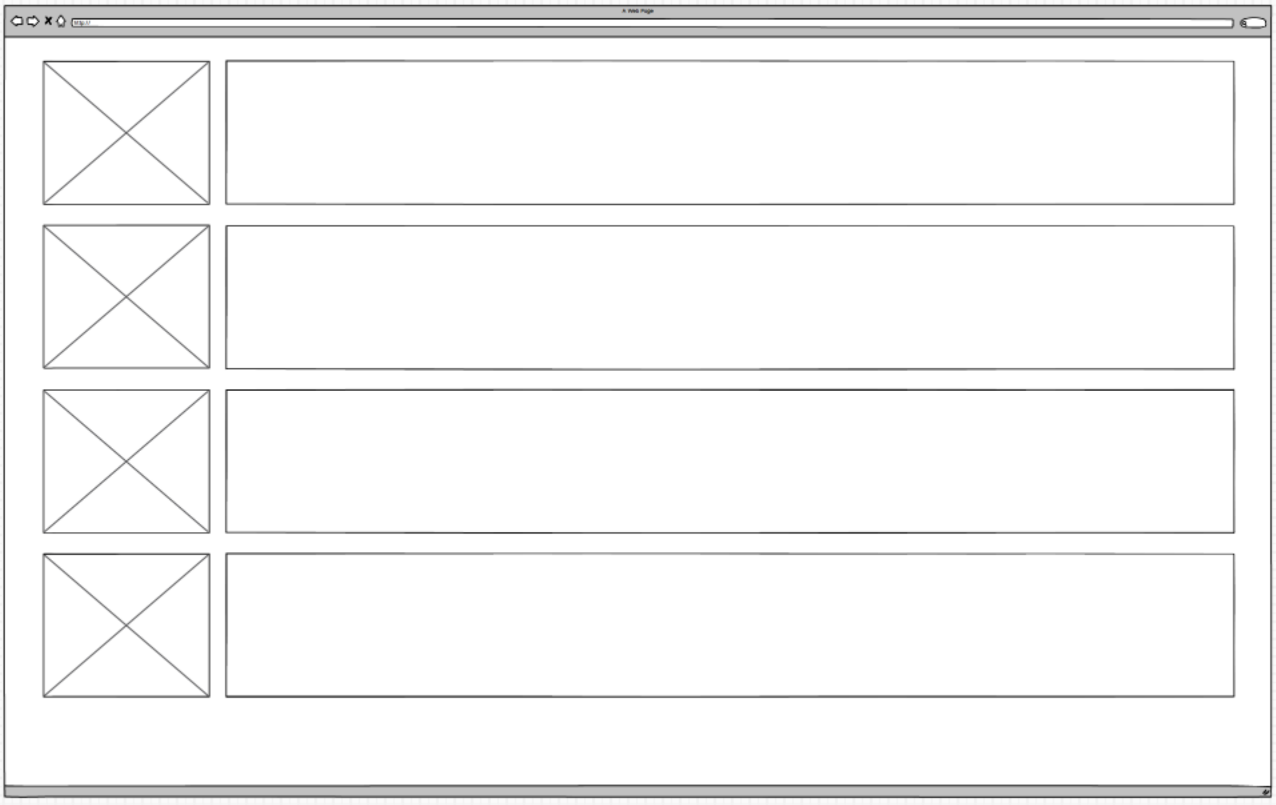
Make a responsive design to be as close as possible to mockups

Target devices:

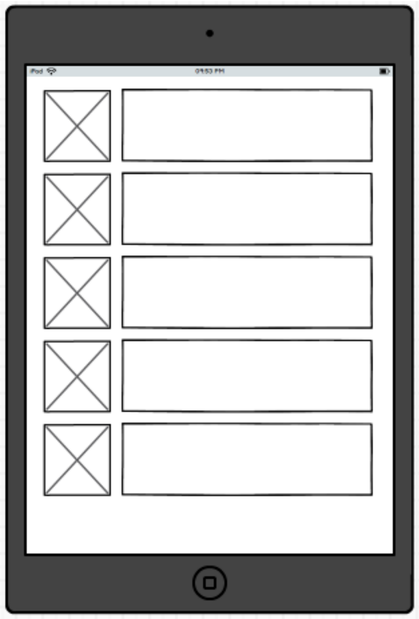
- retina macbook 2880 x 1800
- Ipad (2048 x 1536) vertical/horizontal layout
- Iphone 1136x640 vertical/horizontal layout

Layout is consist of blocks - image - corresponding lorem ipsum by your choice

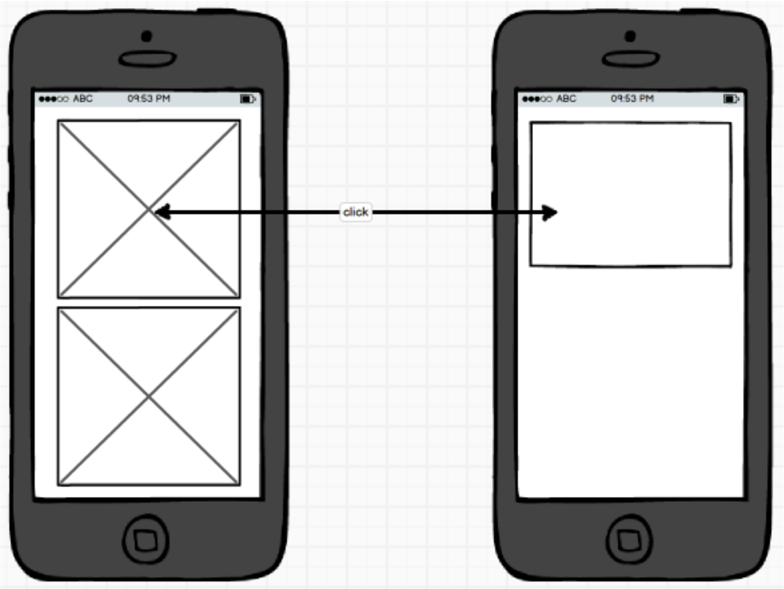
Macbook



Ipad



Iphone



On phone we need to show only images followed by click should show only text block

Задание № 5 Storage

> As a guest I want to load my picture to the site's gallery in order to populate public meme collection

- Allow on main page of the app to load a photo via regular upload process
- Process image - store 100x100 thumb
- After upload - reload page, show thumb of uploaded image 100x100 in the grid of other images already uploaded
- That photo should be stored via dragonfly on Amazon S3
- credentials to s3 should be loaded via ENV (Heroku style) (key, secret, bucket, region)

- Image should be served directly via s3 url on pages