

# Django Evaluation - Final

**Deadline - Within 7 days of the date when it was assigned**

**Grace period - 1 day**

This evaluation has three problem sets:

1. [Problem Set 1 - Working with Regex](#)
2. [Problem Set 2 - Functional web app](#)
3. [Problem Set 3](#)
4. [Important Notes](#)
5. [Submission Process](#)

## Problem Set I - Regex

[Repo Link](#)

1. Write a regex to extract all the numbers with orange color background from the below text in italics (Output should be a list).

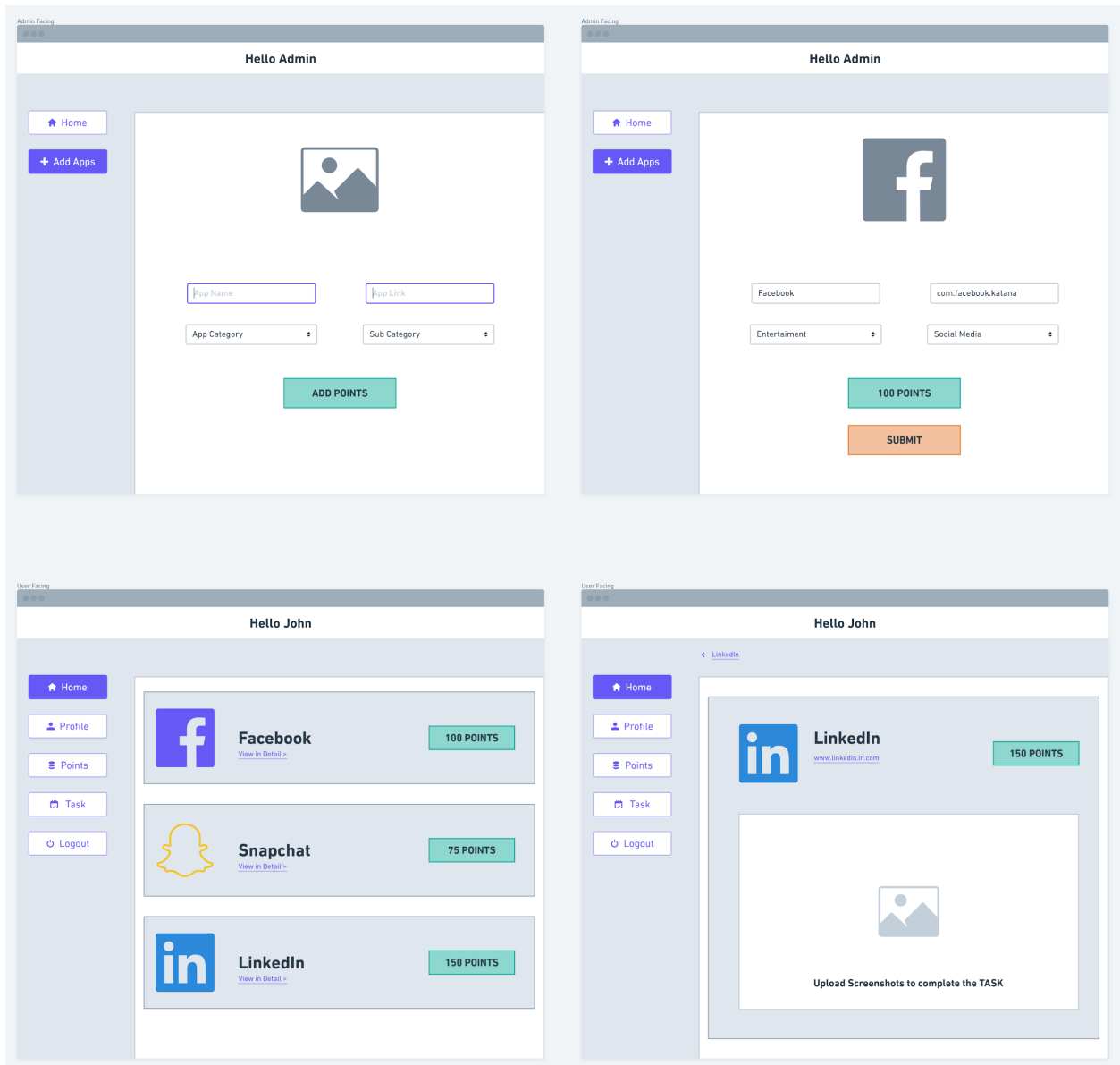
*{  
"orders": [{  
 "id": 1,  
 "id": 2,  
 "id": 3,  
 "id": 4,  
 "id": 5,  
 "id": 6,  
 "id": 7,  
 "id": 8,  
 "id": 9,  
 "id": 10,  
 "id": 11,  
 "id": 648,  
 "id": 649,  
 "id": 650,  
 "id": 651,  
 "id": 652,  
 "id": 653  
}],  
"errors": [{"code": 3, "message": "[  
PHP Warning #2] count(): Parameter must be an array or an object that  
implements Countable (153)"]}]}*

## Problem Set 2 - A functioning web app with API

Part A - Create a Django App

## I. Please create a website - which should have two components:

1. **Admin Facing** - Where admin user can add an **android app** as well as the number of points - earned by user for downloading the app. Please ensure that you scrape the category of the app from the playstore - every 24 hours (Bonus points for using Celery/RabbitMq or any advanced scheduling and distribution system). (Please do not use the default Django Admin)
2. **User Facing** - where the user can see the apps added by the admin and the points. The user should be able to see the following fields.
  - Signup and Login (Feel free to use any package for the same).
  - Their Name and Profile
  - Points Earned.
  - Tasks completed.
  - Option to upload a screenshot (which must include drag and drop) for that particular task. (Like if a user downloads a particular app), he can send a screenshot of the open app to confirm that he has indeed downloaded the app.



## Part B - API and Working with Google Apps Script (JS)

Once you are done with Part A, in the same project, expose all the endpoints using Rest API, and these should be permission based views. Also, create a sheet - and write Google Apps Script to fetch the data from the API. (So the user will input the serial number - 1,2,3 and it should return the respective app data - like app url, points, category, etc).

<https://docs.google.com/spreadsheets/d/14thWjndJCdA2Mwado9I2TJsQg8R4Gez0lejXO8eekX4/edit?usp=sharing>

# Problem Set 3

Please answer the below questions:

A. Write and share a small note about your choice of system to schedule periodic tasks (such as downloading a list of ISINs every 24 hours). Why did you choose it? Is it reliable enough; Or will it scale? If not, what are the problems with it? And, what else would you recommend to fix this problem at scale in production?

B. In what circumstances would you use Flask instead of Django and vice versa?

## Bonus points

- For good s/w eng practices: e.g. modularity, well documented (comments and README), Normalized db schema, requirements.txt etc. Blow my mind!

## Important Notes

- Feel free to Google or Stackoverflow (or even go as far as read a book) to understand anything, but please do NOT copy/paste any code/snippet.
- Finish your assignment before the assigned deadline. If you need to extend the deadline, please make sure you drop us an email.
- Document how you deployed the project (in README)

## How to Submit

Fill this submission form <https://forms.gle/tTUPT9x9rrqK8S7G9> We only consider the submissions through Google Form wef April 15th, 2021. (Before you fill up the form, please read the below four important pointers)

1. Please create a repository in GitLab(Not Github), and add **@NLEvaluations** for evaluation (Only maintainer's access), and deploy it to any server and share the link.
2. Please include a short video/screencast - running the evaluator through both the problem statement and your solution (While we will not assess the longer videos negatively, we'd prefer the videos to be under 5 minutes - and it should

be very concise and to the point). You can add the link to the Readme of the repository.

3. Is there any way that you can automate the delivery? Wow us and get bonus points for the task.
4. All the answers to the questions (As asked in Part D) should go in ReadMe.