IIT Delhi

Dev Club Recruitment Test

Back-End Assignment

March 1, 2019

1 Introduction

This assignment is aimed to get you familiarized with design and development. At the first glance it may look hard, but do what you can, there is a lot to learn from this assignment. Follow the guidelines, ask clarifications on the slack channel mentioned at the end (We may drop special hints there!!)

Submission Deadline : **10th March 2019 11:55 PM**

2 General Guidelines

- Front-end and Back-end are different profiles, you are required to **attempt at least one**.
- A super formal report is not required for Design documents. Even flow-charts and documenting your thinking process will do the job.
- Photos of handwritten documents will not be accepted. Any and all explanations should be in typed format.
- Coding Assignments website/app should be hosted (or APKs present as described in the submission instructions) and the code should be on Github. See the Part II References and Submission Instructions for help on these topics. Remember, the easier it is for us to run and see your work, the easier it'll be to evaluate.
- Always try and submit (even if incomplete). You will be given points on how much effort you have put in. If something doesn't work search through stackoverflow. If you still can't figure it out, then document details of what all you did before you gave up on that **BUG** and raise an issue on your repository so that we can have a look at the problems you faced.
- You should get an overall idea of the assignement beforehand. Download the softwares you would use, on IITD Wi-Fi. You can download some softwares from our website, without spending your proxy. See Part II References and Submission Instructions

Dont care about the end result, just give your best !!

Part I

Development Assignment

1 Research

Definition: To become an expert on something after having done a Google search and 10 minutes reading on the internet. – Urban Dictionary

This section is common for both the Front-end and the Back-end Assignments. Spend some time getting familiar with the questions and the underlying technologies. Submit what you understood, a basic understanding and answers to each of the questions asked.

1. Research and document how a basic networking system works [Read about reverse tunneling, UDP/TCP, ports on a machine, what setting up a server means etc]. Example IITD proxy system.

   - Write about how does a website work? What process is followed when you type in the url of a website?
   - Suppose you created an html page and wanted to show this to a person using the same wifi network. How would you accomplish this if you are neither allowed to upload your site nor give the source code through external storage? [Hint: Google setting up a local server using python]
   - Would the above method work if the person was using his mobile data and you were using IITD Wifi ? Why/Why not ?
   - Get an overall idea of what is NginX and Apache (no need to dive too deep) and what role do they play in websites.

2. Research about the various types of architectures that web app development frameworks employ. Read in detail about the Model-View-Controller Architecture. Which popular frameworks employ this Architecture?

   - In a framework, how does the controller manage the views? How are the views fetched and displayed?
   - Get familiar with the following 2 frameworks(Only basic understanding needed):
     - Ruby on Rails
     - Django

     Understand and document the differences between the architectures of the two frameworks.

- Some argue that Django is not a MVC framework but instead a MTV framework. What's the difference between the two?

3. The Accelerated Mobile Pages (AMP) Project is a website publishing technology developed by Google as a competitor to Facebook's Instant Articles. In February 2017, a year after the public launch of AMP, Adobe reported AMP pages accounted for 7% of all web traffic for top publishers in the United States.

    - Read up about AMP and prepare a brief write-up about what *you* learnt from it. Especially, look for the intricacies of the technology - What are the optimisations that AMP makes? Can it work for every webpage? Does it also optimise the forms submitted on the webpage? etc. We would love to see what all you have gathered.
    - There is also a very recent technology - Progressive Web Apps(PWA). How are PWA and AMP similar and how are they different? Compare the two on different parameters - like Speed, Optimisation, User Experience, etc.

2 Time to Code

You need to implement an IIT Bazaar portal and develop a Web-App for it. Try to break your working in steps like the following -

- First design a template layout of your app in an abstract manner, so you know what you are going to code.
- Decide upon an appropriate framework for implementing your app, like express-js, python-django, etc.
- You may start with defining the basic model structures of various object entities which would be existing in the portal (products, users, comments, etc. - specified below)
- Then design an appropriate workflow between various entities, about how they interact with one another.
- Finalize the app by debugging the functionality of all the features.

Also draft and submit a design document, stating what is the basic database model or schema of your application and how are you providing access control. Just bullet points clearly stating what and how your system works is fine.

2.1 Specifications

The specifications are as follows:

2.1.1 Minimum Requirements

1. Guest users should be able to view the products put up on sale, and some advertised features of the web apps.
2. Users registered on the portal should be able to access the main functionality of the portal as specified below.
3. Registered users should be able to both buy and sell products, i.e., initiate the procedure for buying any product publicly put up on sale, and be themselves able to put up a new product on sale.
4. It's up to you to think appropriate procedure for sale of products, and also what specifications about a product should be given by the seller like name, type, price, photograph, etc
5. You should have a profile page of the user, which shows basic info about the user and the status of all the products concerning him/her.
6. It's up to you to how to properly categorize products to provide ease of browsing to the users.
7. You can implement a bidding system like ebay or a normal purchase system like amazon.

2.1.2 Preferred Requirements

1. There should be also a provision for administrators on the portal, who should be able to filter the products and monitor the activity.
2. The admin should be able to flag inappropriate products, which would then not be visible publicly to other users and an appropriate message regarding this should be visible to the concerned seller.
3. The admin should also be able to access activity records such as who sold what to whom and by how much, and when. Only information regarding recent transactions in main display of admin, should suffice.
4. Think about how you can make the portal both transparent and secure at the same time, while implementing it.
5. Bonus: You may add exciting features to incentivise the use of the portal, for example, respect points to regular users, etc.
6. Records of "done-with" products should be properly maintained for future references, accessible by the admins.

2.1.3 BONUS

This may lead us to believe your exceptional candidature.

1. Extend the app with user roles. Have an admin, with userid as *admin* and password as *admin* to be the only one who is able to delete users within the app and add/remove products.

2. You can also have a registration system, either standalone, or linked with google, facebook, etc.

2.2 Notes

1. Assume, that there are 10 users who are registered on the portal. The user *i* has his user id as *user_i* and password as *pass_i*. So, user 1 has his id as *user_*1 and password as *pass_*1.

2. Also, put up some dummy products with dummy data, for us to be able to check the features of the web app. Include any specifications in your submission notes, if necessary.

3. The UI can be minimial i.e. basic HTML. We will not be grading on the basis of the UI but on your backend functionalities like data handling,access control etc.

Part II

References and Submission Instructions

1 References

Because we are only Human. –Unknown
If you are stuck somewhere, seek help on Slack Workspace

1. Github:
   (a) https://try.github.io
   (b) https://www.codecademy.com/learn/learn-git
   (c) https://help.github.com/
   (d) https://learngitbranching.js.org/
2. Basic HTML
   (a) Mozilla Docs
   (b) Guide to Most Tags in HTML: https://developer.mozilla.org/en/docs/Web/HTML/Element
   (c) You can try the HTML, JS and CSS you code at https://codepen.io/ (click New pen) which interactively shows you the result of what you do.
3. CSS:
   (a) http://learn.shayhowe.com/html-css/building-your-first-web-page/
   (b) https://www.w3.org/Style/Examples/011/firstcss.en.html
   (c) https://www.tutorialspoint.com/css/what_is_css.htm
   (d) http://materializecss.com/
4. Firebase:
   (a) Authentication
   (b) Database setup
5. Django
   (a) Youtube tutorials
   (b) Official Docs

6. Ruby On Rails
   (a) Youtube tutorials
   (b) Official Docs
7. Express - JS
   (a) Youtube tutorials
   (b) Official Site

2 Submission Instructions

Submission Deadline : 10th March 2019, 11:55 PM

Submission Link

Check our FB page and slack channel regularly for updates.

1. It is a good practice to use version control system like github for code. All the code should be hosted on github (Make an account if you don't have). Make sure the repository is public with your final code in the "master" branch and we can see the code. If you fear someone else will copy your code from your github repo, make a private one and later, convert it to a public repository after the deadline. But make sure, you make it public, otherwise we won't be able to see your code.

2. You need to submit the link to your github repo and link to your PDF document via the submission link.

3. We won't accept any hand written documents. They should be typed. For diagram and figures try draw.io.

4. If you want to display any of your other work, that you have done like any other website design, application etc, make a folder named EXTRA in your repository and place a file named README inside it with all the links and descriptions of them.

5. Only the last entry will be considered incase of multiple submissions per candidate on the form.