NOTE: By gaining access and viewing this document, you agree to solve the assessment by yourself and not share this document with anyone else. 🤝

Overview

The primary objective of this assessment is to assess your learning and problem-solving capability. We are looking for people who can do whatever is required to solve a given problem within a specific deadline - be it by effectively researching online for existing solutions, or learning a new language/framework, etc.

You will build an app with Python using Django. Completing all steps is not necessary but completing each additional step will earn you extra points. Being able to deliver a suitable solution in the quickest possible time will set you apart from other candidates. For more info, read the **Assessment Criteria**.

Reach out to us if you have trouble understanding the tasks or want to clear any confusions. This also helps us to understand your progress and communication skills.

Our work is mostly remote, so candidates with self-discipline are expected to perform the best.

Submit solution to: https://forms.gle/WNd9LfiAwtfmOYTe7
Contact us: careers@biniyogfintech.com

Assessment Criteria

- Time taken to submit assessment; quicker == better
- The number of tasks successfully completed; higher == better
- Being able to successfully deploy to heroku (Step 7)
- Ability to communicate clearly and effectively
- Clean & well-structured Code, with documentation where necessary
- Intuitive and effective UX design

Tasks

For each of the steps below, create a git commit with a message such as "Step 1". *You are requested to not commit multiple steps at once*; but if you have, add all the step names to the git commit message like "Step 1, 2, ..." etc.

- Step 1: Build an Auction Site like eBay. In this site anybody can signup using their email address, no complex authentication is needed, when somebody enters their email address at login screen, if

the user with the email address already exists it logs the user in and shows a dashboard for the user. If the email address does not exist, a user is created with the email address and the user logs in.

- Step 2: After login, the user will see the auction item gallery, which shows the items everybody else has put up for auction. There should be a create/plus button in the dashboard that allows the user to create an auction item for everybody to see and interact with. If the user hits that button, a form appears, which lets the user input Product Name, Product Description, Product Photo, Minimum Bid Price, and Auction End DateTime. When an auction item is created using this form, it shows up in the auction gallery of everybody else, and shows up in the "My posted items" menu for the posting user.
- Step 3: Users can place bids on items posted by others within the Auction End DateTime. If they click on any item in the auction item gallery, they will be taken to the auction page of that item, where they can see the Product Name, Product Description, Product Photo, Minimum Bid Price, and Auction End DateTime. It will also show a table of bids placed by other users for that product. An user can input their bid in the auction item page for that product. After inputting a bid, it shows up in the bid list. The user can edit their bid before the auction ends.
- Step 4: The auction will end at Auction End DateTime, If you enter the auction item page for any item, it will show the bid winner for that item.
- Step 5: Make an admin dashboard where the admin can see the auction gallery and auction item pages. Also the admin can see statistics on how many total auctions are running now, what is their total value. Include a chart where it shows a time series of auction added count, and auction completed count in two variables,add another chart that shows the total auction value by time, based on the latest bid placed. You should be able to change the frequency of these charts from a dropdown, frequencies are minute, hour and day. The admin can login at /admin endpoint, with the default admin/admin user pass.
- Step 6: Make everything look pretty based on your judgment.
- Step 7: Whatever you have built, deploy it to heroku. Also put the live website link on your README.md.
- Step 8: On your README.md file:
    - Create a section named "Challenges", write all the challenges that you faced in each of the steps, and how you solved them.
    - Create a section named "Packages", list the packages that you used and document why you used them.

Best of luck! You got this. 👍