# 

# Flask Hiring Test

### **Why this test?**

We build applications for the web, so understanding how web applications work, is a pre-requisite for any new engineer. Flask is one of the simplest and well written Python based web application frameworks and can easily be learned in a few hours.

If you have never worked on web applications before, this test will also help you evaluate whether this is something you are skilled at or can easily learn doing.

### **Purchase/Sales Management Web Application**

The goal is to create a web application using Flask framework to manage a small petty shop where there are items, purchases and sales.  
  
The application should cover the following sample scenario:

This is just a sample scenario and your submission need not follow the same terminologies.

There is a company called “Namma Kadai”. Its initial cash balance is Rs. 1000.

There are five items.

* Pen - Rs.5/each
* Pencil - Rs.2/each
* Eraser - Rs.1/each
* Sharpener - Rs.2/each
* Geometry box - Rs.10/each

Namma Kadai purchases 10 Geometry boxes (10 \* 10 = Rs.100). With this its cash balance becomes (1000 - 100 = Rs.900).

Namma Kadai sells 10 boxes for 15 each and earns (15 \* 10 = Rs.150). With this, its cash balance becomes (900 + 150 = Rs.1050)  
  
At any point in time, Namma Kadai should be able to add new items, make new purchases and make new sales and see its running cash balance.

The application should cover the following functionalities:

#### Database Tables:

* Company (company\_name, cash\_balance)
* Item (item\_id, item\_name)
* Purchase (purchase\_id, timestamp, item\_id, qty, rate, amount)
* Sales (sales\_id, timestamp, item\_id,qty,rate,amount)

Note:

1. Primary keys can be text / varchar
2. “Rate” can be filled as any value while making a purchase or sales. Corresponding amount (rate \* qty) should be updated in the cash balance of the company.

#### Views:

* Add/Edit/View Item
* Add/View Purchase
* Add/View Sales

#### Report:

* Show current cash balance.

**Bonus**

* **Add the “qty” column to the “Item” table. Make it 0 initially. On every “Purchase” add the corresponding qty. On every “Sales” reduce the corresponding “qty”. Show a report of Item and its current qty.**

#### Use Cases:

* Create 3/4 Items
* Create 3/4 Purchase
* Create 3/4 Sales
* Show the updated Cash Balance
* If the Bonus task is attempted, Show the updated Item quantities.

### **How to Submit**

Once this application is ready, push your code to [GitHub](https://github.com/) and add a README.md file in your repository. Add screenshots showing the screens and reports in your README.

After this, just fill in the information in this Google Form (<https://forms.gle/Va54YX2mHnRYTsrp6>)

### **Next Steps**

After this, we will get in touch with you and you will have to walk us through your code. We will then ask for a few changes or additional features that will help us evaluate how well you know your codebase.

### **Tips to Ace the Code Evaluation**

* Build a good UI, with clean forms and reports
* Write concise SQL queries if not using an ORM
* Avoid code duplication, abstract out functions that can be reused

**A word of caution**: In the past, applicants have not done this test themselves and have completely failed to explain or make changes. We know it sounds dumb, but people do such things. So copying it from some other source will bring down your reputation. Please do not do it.

Feel free to ask for help from anyone or just look on the web. Do this project by yourself if you want to sustain a career in programming and technology!

All the best! 👍

### **Links**

* <http://flask.pocoo.org/docs/1.0/>
* <https://dev.mysql.com/doc/mysql-getting-started/en/>
* <https://www.python.org/about/gettingstarted/>
* <https://www.codecademy.com/learn/learn-python>