

Fullstack Engineer Assessment Task

Introduction

If you are reading this, it means that you have been selected for the next round. Congratulations! We require you to perform a simple task to assess your technical qualifications which are in line with your the responsibilities you're supposed to take. Please read this document carefully and submit your answer in a timely manner. Thank you in advance for your time!

Setup

You will be given access to a remote git repository via email. Please make sure you can fork the repository and contact us in case of any problems. You must submit your answer via a pull request for your fork.

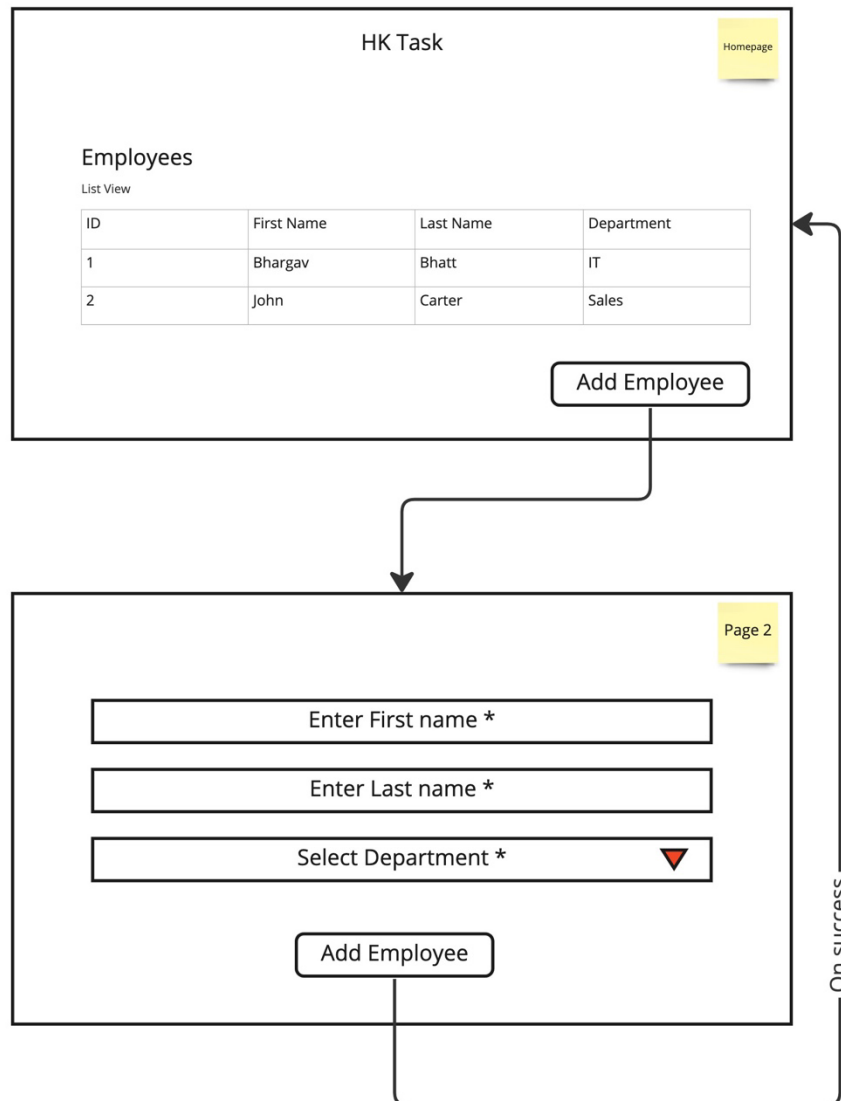
You can opt for the programming language either python or Java. You can choose any framework you feel comfortable with.

Main Task

We want you to create a minimal full-stack setting for Heimkapital. In this respect, your app should:

- Contain a PostgreSQL database with two tables *employee* and *department*
 - The *employee* table should have (employee ID, first name, last name, department id [foreign key])
 - The *department* table should have (department name, department id)
 - You'll need to add three departments initially: [Sales, Ops, IT]
- Contain a REST API where the connection between the front end and database is facilitated.
 - You'll need to implement READ and INSERT operations for the tables through this API, but you don't need to cover delete & update
 - You'll need to use a sort of ORM (like sqlalchemy or Django ORM) to have an object-oriented view of your DB tables and perform DB operations in your code. DON'T WRITE RAW SQL IN YOUR CODE!

- Contain a simple front end where the listing of employees is visible and new employees can be created
- Communication between the backend and front end should be through the REST APIs you created before
- The UI should look clean but we don't require fancy frontend functionality. Be creative. The wireframe is as below:



Final Notes

Make sure you have scalability in mind. Be as modular as you can. Write clean code, comment, use git branches if necessary, write clean commit messages etc. DON'T FORGET TO UPDATE THE README WITH INSTRUCTIONS TO RUN YOUR CODE.

One final note: containerization gets plus points! In case of any questions, feel free to write an email at bhargav.bhatt@heimkapital.de

[Happy Coding!](#)