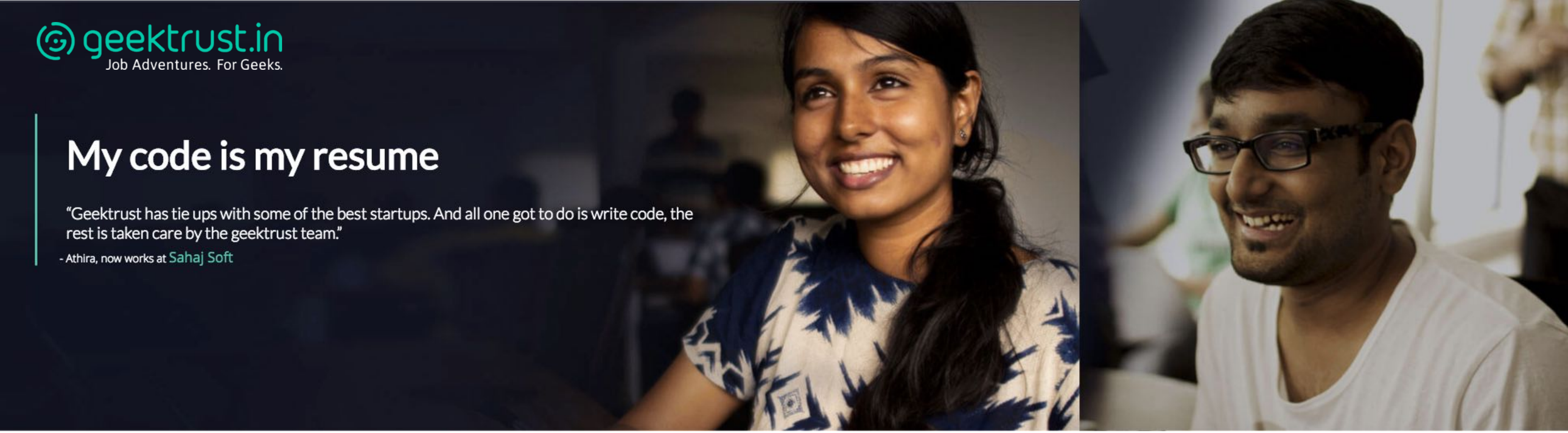


*Athira, Souranil and many more developers have solved Geektrust coding challenges to find great jobs.*

- **Get priority** and be treated as a premium candidate to directly connect with decision makers at companies.
- **Get membership** and win an exclusive Geektrust DEVELOPER t-shirt given when you write good code.

What we look for in your code - It's not just about getting output, but how you get it. We care about how well modelled your code is, how readable, extensible, well tested it is. Check out our coding help page to ensure you get a good score.

# Getting started

1. Getting the output right is important, but more important is clean code and how well designed your code is. You should **absolutely** see our Help page post on what we look for in your code, and how to get started with the coding challenge.

2. See our evaluation parameters here and the badges to earn here.

3. Add a readme with how to get your code working, and how to test your code.

# problem 1: WHATflix

King Shan is very fond of movies and wants to take his family for a movie. However, they can't agree on which movie to watch. So King Shan declares a movie watching spree and takes each person to watch the movie they want. Your coding challenge is to:

1. Build a Restful web service API which will accept a search string and userID and return unique movies in the order of preference for that user.

2. Build a Restful web service API which will list out all the users and the top 3 recommended movies for each user based on their preferences

Develop the application in a language of your choice and host the service in Cloud infrastructure like AWS/Heroku and give us the url endpoints or a Postman project for the same.

# API 1: input

Request Type : GET

URL : http://<url>/movies/user/$userId/search?text=<text>

Where the $userId is the id of the user from the user preferences JSON file (see file) and <text> is the search text. The search text could be multiple words separated by comma, in which case it will search for all of those. Or it could be a single urlencoded entry. The search text is matched against actor, director and title fields (director_name, actor_1_name, actor_2_name, actor_3_name and movie_title in the movies data) of the movies. All matches are included in the results.

<url> is the domain name in the URL where you are hosting the solution.

For example - http://<url>/movies/user/$userId/search?text=Tom%20Hanks -> This will return all the movies matching “Tom Hanks” considering the preferences of the user

# API 1: output

An array of movies name, found based on the preferences of users, sorted in this order:

1. First show the movies matching the user's preferences and search term. This should be further sorted on the alphabetic order of the titles. There is a chance this set of movies could be empty if there is no search result matching the user's preferences.

2. Next show the movies matching the search term in the alphabetic order of titles (even if it does not match the user's preferences). If #1 is empty then only these set of movies will be there.

["Movie 1","Movie 2","Movie 3"]

# API 2: input

Request Type : GET

URL : http://<url>/movies/users

# API 2: outPUT

A json array of userids and top 3 movies recommended. The movie names are sorted in the alphabetic order of titles. If there is no recommendations then it can be an empty array.

```
[
  {
    "user": "100",
    "movies": [
      "A",
      "B",
      "C"
    ]
  },
  {
    "user": "101",
    "movies": [
      "A",
      "D",
      "C"
    ]
  },
```

And so on and so forth.

# Data sets

We use the Kaggle TMDB data set for this problem. Complete movie data set can be downloaded from here.

The user preferences data file can also be downloaded from here.

# What we look for in your code

1. Overall architecture. Please submit documentation to explain your choices and the complete architecture.
2. Code scalability. Can it handle 1000s of hits a second?
3. Clean code, RESTful semantics, and object modelling
4. Bonus points for automating the deployment of your solution
5. When the instructions are not clear or there are multiple solution possibilities, make a choice that you find reasonable and practical.

# check list - submitting code

1. Please compress the file before upload. We accept .zip, .rar, .gz and .gzip

2. Name of the file should have the name of the problems you have solved TechArchitectSet1.zip in this case

3. We advise not to put your personal details in your solution as we maintain your anonymity with a company until there is genuine interest from them.

4. Please upload only source files and do not include any libraries or executables or node_modules folder.

5. You can expect your evaluation in 3-5 working days.

6. Yes, you can resubmit code based on our feedback. We accept 3 submissions in total. So do implement all feedback and make your submissions count!

# what next?

## A few good developers

Write great code. Get membership. Explore jobs.

**Write Code**

Sign up to solve interesting coding problems

**Be a Member**

Clear evaluation and get featured on GeekTrust

**Connect with Companies**

Explore opportunities as companies reach out to you

**Find the Perfect Job**

Review options, interview & find the right job for you