Coding Task

Create a REST api to be consumed by a mobile app, which is somewhat similar to various popular apps which tell you if a number is spam, or allow you to find a person's name by searching for their phone number.

You can use whichever language/framework you're most comfortable in. However, we give strong preference to candidates in the pipeline who've done it using Django, and to a lesser extent Flask or Rails. For persistence you need to use a relational database along with an ORM for your framework. We will not evaluate NoSQL or raw SQL queries.

Terminology and assumptions:

- Each registered user of the app can have zero or more personal "contacts".
- The "global database" is basically the combination of all the registered users *and* their personal contacts (who may or may not be registered users).
- The UI will be built by someone else - you are simply making the REST API endpoints to be consumed by the front end.
- You will be writing the code as if it's for production use and should thus have the required performance and security. However, only you should use only a web server (the development server is fine) and a database, and just incorporate all concepts using these two servers. Do not use other servers.

Data to be stored for each user:

- Name, Phone Number, Email Address.

Registration and Profile:

- A user has to register with at least name and phone number, along with a password, before using. He can optionally add an email address.
- Only one user can register on the app with a particular phone number.
- A user needs to be logged in to do anything; there is no public access to anything.
- You can assume that the user's phone contacts will be automatically imported into the app's database - you don't need to implement importing the contacts.

Spam:

- A user should be able to mark a number as spam so that other users can identify spammers via the global database. Note that the number may or may not belong to any registered user or contact - it could be a random number.

Search:

- A user can search for a person by name in the global database. Search results display the name, phone number and spam likelihood for each result matching that name completely or partially. Results should first show people whose names start with the search query, and then people whose names contain but don't start with the search query.

- A user can search for a person by phone number in the global database. If there is a registered user with that phone number, show *only* that result. Otherwise, show *all* results matching that phone number completely - note that there can be multiple names for a particular phone number in the global database, since contact books of multiple registered users may have different names for the same phone number.
- Clicking a search result displays all the details for that person along with the spam likelihood. But the person's email is *only* displayed if the person is a registered user and the user who is searching is in the person's contact list.

Data Population:

- For your testing you should write a script or other facility that will populate your database with a decent amount of random, sample data.

Evaluation criteria:

- Completeness of functionality.
- Correctness under thorough testing.
- Performance and scalability of APIs.
- Security of APIs.
- Data modeling.
- Structure of code.
- Readability of code.

Offer criteria:

- If you are applying for an internship, we expect most functionality to be present with few bugs.
- If you are applying for a full time role, in addition to all functionality we expect your submission to be strong in each of the evaluation criteria, depending on the experience level for which you will be evaluated. Basically it should be something that can be used in a production application directly. *You* need to think of what all needs to be handled in a production API above and beyond what's mentioned above (we want to evaluate whether you will be able to work independently).

How to send the code:

- Reply to the *same* email in which we sent you this, with a tar or zip file containing your source code. We use a tool to keep track of submissions, so if you don't reply to the same email we won't receive it.
- Don't put it up online on Github etc - if you make it publicly accessible, we will not reply.
- Sometimes gmail doesn't allow code attachments to be sent - if that happens upload the file on Google Drive and share the access/link with us.
- Please don't include your environment or similar folder as it takes a large amount of space. Any submission that is over 1 MB will not be evaluated.
- Include instructions on how to run your code.
- If you have any questions, feel free to reply and ask.

Next steps:

- Please note that usually we try our best to reply with a decision within a week of receiving your code. However, since we are a startup, sometimes urgent work may come up and it may take us a couple of weeks to respond.
- In case we don't go ahead with an offer, please note that we won't be able to give you detailed feedback on your assignment. You can do a self-assessment based on the criteria above.

Request regarding plagiarism:

- Please ignore this if, like most people, you are not plagiarising your work.
- Please don't send code that is partly or fully written by someone else. We check every new hire's code for plagiarism one month after you join us. If any level of plagiarism is found in your assignment, or any misrepresentation is found in the professional credentials you present in your resume, then 1) your employment will be terminated immediately and no salary will be due and 2) we will add your name and email address to a webpage containing a list of plagiarizers that is publicly viewable, to help other companies avoid hiring folks who plagiarize.