



# Address Book

## Technology Stack

- Programming Language: JavaScript, TypeScript
- Platform: NodeJS, Angular
- Source Control: Git, GitHub
- DevOps Tools: Docker, Docker Hub, Travis CI
- Deployment Platform: AWS, GCP or Heroku
- Database: MongoDB

## Project

Our company wants to develop the best Address book application ever so far. Address book application consists of two parts, frontend application, and backend Rest API. Your assessment is the development of a backend Rest API and a simple UI on top of that to meet the following requirements

## Technical Requirements

We never negotiate on quality so we would like to have the best address book ever so please try to follow software engineering patterns and practices as much as possible.

### Must

- Using best practices and patterns for both parts Backend and Frontend
- Scalability and performance should be considered
- Code quality, best practices and patterns matters
- Backend API should be cloud native and be able to run any of cloud services mentioned above
- Instructions in GitHub about how to install run project.

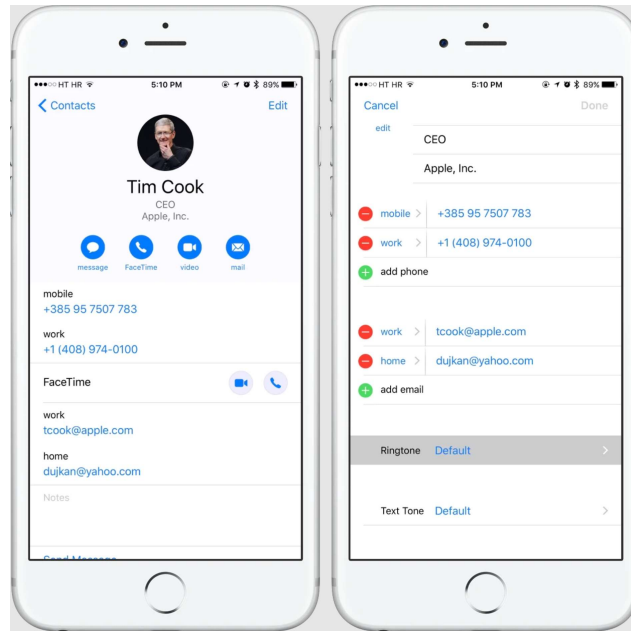
### Nice to Have

- Open API documentation
- CI/CD integration through GitHub and Travis CI or any other alternatives
- Test coverage, code quality checks and reports as part of CI/CD

After the project is completed you are expected to share the GitHub address of the project with the working and live URL of the backend API, frontend Application along Open API documentation. Please do not commit all changes at once, we would like to see commit history like a real-world project.

## Business Requirements

Our mobile frontend application should look similar to the classical iPhone contact list application as you can see below therefore, we will also need a backend API that meets our frontend requirements.



- Users should be able to add new contacts by providing their information such as name, address, phone, mobile phone and e-mail
- Name, address, and phone fields are required
- API should warn user in case of missing or invalid data
- Given e-mail address should be valid in terms of international e-mail address rules
- Users should not be able to add new contact if the contact's name is found in the address book
- Users should be able to delete contact from the address book
- Users should be able to update contact in the address book
- Users should be able to search contacts by providing any of contact's data such as name, address, phone, e-mail