**Problem Statement**

*Build an Order Management System which would provide following capabilities for user.*

Sign up screen which allows to signup using gmail which can be used to pull information ( Assumptions either gmail or facebook login capability available )

1. *Left Menu section displaying following options*
   * *My Profile*
   * *My Orders*
2. *Profile should display user details (like Name, Age, DOB, Phone, Email, TimeZone, Billing/Shipping Address). Implement REST/GraphQL APIs to support profile management use cases.*
   1. *Add Profile*
   2. *Update Profile*
   3. *View Profile*
3. *Orders should provide following features*
   * Create Orders
   * *Search Orders (Lookup based on Ordered Date, Product Name, Email/PhoneNumber etc)*

Design/Implement REST/GraphQL APIs to solve order management use cases.

Your Tasks:

* Build a Order Management Tool. Obviously it is a huge problem to be built in limited time. We are not expecting a full blown solution but curious about your thought process, choice of features and technology and your ability to think and work on it individually.
* This is an intentionally vague problem. It is your job to decide what aspects of the ordering platform are worthwhile to implement for this exercise. If you have any other interesting ideas on improvising this application (delighter features) then feel free to add those features (if time permits). We encourage you to reach out to us with any additional clarifying questions.
* If you have in depth proficiency in designing and building only the backend, then you can chose to create a simple UI. Similarly if your proficiency is in developing frontend then design and build the UI with data that could be stored in memory.
* Bonus: Since the position we are looking for is a full stack engineer, this exercise would be apt to show your skills in developing an end to end application (UI, REST Services, Database). Build at least 2 REST endpoints and use a persistent data store that might be apt to this use-case.

**On the day of the interview:**

1. Come prepared with a high level architecture, designs you have thought, code (tools you have used to build) and challenges you have faced.
   * + E2E Application Design
     + Persistence Layer Design
     + Component level design
     + Unit Tests/ Automation
2. You are expected to demo the application and explain the rationale for your choice of feature set, technologies and architectural patterns.

If you have any questions on the problem statement, please feel free to reach out to

[Raza\_naqvi@intuit.com](mailto:Raza_naqvi@intuit.com), [Rahul\_tiwari@intuit.com](mailto:Rahul_tiwari@intuit.com)

When you have finished the Craft Demo, please publish to github and send the github repo link to us or email us the code and any deck you have prepared.