## Auth Service for Multi tenant Saas - backend engineer

### Tables

Please create these tables in Postgresql or MySQL using any migration tools,You can find the table schema in the last page of this document.

1. User
2. Organization
3. Member
4. Role

The above shown 4 tables are related like shown below

**One User** is part of **Many Organizations**

**Member** Table is the Many to Many mapping table

A User has a Role in an Organization

### Table schema

Organisation

name = String, nullable=False

status = Integer, default=0, nullable=False

personal = Boolean, default=False, nullable=True

settings = JSON, default={}, nullable=True

created\_at = BigInteger, nullable=True

updated\_at = BigInteger, nullable=True

User

email = String, unique=True, nullable=False

password = String, unique=False, nullable=False

profile = JSON, default={}, nullable=False

status = Integer, default=0, nullable=False

settings = JSON, default={}, nullable=True

created\_at = BigInteger, nullable=True

updated\_at = BigInteger, nullable=True

Member

org\_id =

Integer, ForeignKey"organisation.id", ondelete="CASCADE", nullable=False

user\_id = Integer, ForeignKey

"user.id", ondelete="CASCADE", nullable=False

role\_id = Integer, ForeignKey

"role.id", ondelete="CASCADE", nullable=False

status = Integer, nullable=False, default=0

settings = JSON, default={}, nullable=True

created\_at = BigInteger, nullable=True

updated\_at = BigInteger, nullable=True

Role

name = Column(String, nullable=False)

description = Column(String, nullable=True)

org\_id = Integer, ForeignKey"organisation.id", ondelete="CASCADE", nullable=False

### APIs

Write these below shown apis in any Python Server (Framework is not important). Test the different scenarios and send us a github repo link along with the postman export collection

1) Sign in

Verify User encrypted password in User table and

Returns a JWT Token ( Access token and Refresh token)

2) Sign up

           Add an user entry

           Create a new organization (Get organization name and organization details as input )

           Add member entry with owner role

3) Reset password

4) Invite member

5) Delete member

6) Update member role

===============================

APIs for stats

1)  Role wise number of users

2) Organization wise number of members

3) Organisation wise role wise number of users

4) add from and to time filter and status filter to both APIs 3 and 4

Email APIs test

Use any Email API (twillio,resend, brevo etc). Email API is left to the candidate’s choice

1) Send an invite email on sign up and invite with a generated link

2) Send an password update alert email

3) Send login alert event email