

b2match interview task

The app is written in node.js. The api is inside /routes/api.js.

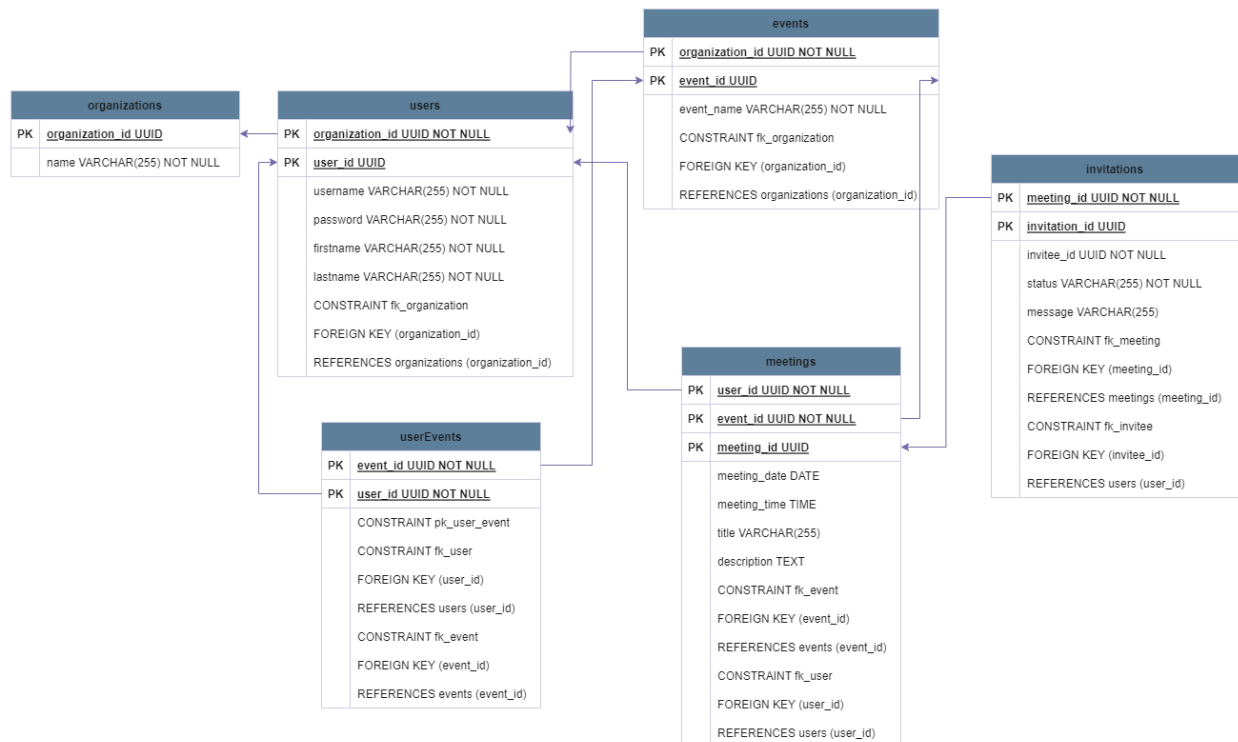
Even though the calls look like “/users/:user_id/meetings” they are and should actually be “/api//users/:user_id/meetings”. The api is lacking because I’m using an api router inside app.js file.

```
var apiRouter = require('./routes/api');
```

```
app.use('/api', apiRouter);
```

The file webapi.js and /files/site.js should be ignored. Initially I wanted to create a user interface as well but due to the lack of time, and the fact that it wasn’t part of the task I ended up not completing it.

For the database I used PostgreSQL and I hosted it on Render.com to make the application calls functional without a locally running database. The database dummy information is listed below as well as an ER diagram for an easier understanding of the database.



```

-- Insert dummy organizations
INSERT INTO organizations (organization_id, name)
VALUES
  ('11111111-1111-1111-1111-111111111111', 'Organization A'),
  ('22222222-2222-2222-2222-222222222222', 'Organization B');

-- Insert dummy users
INSERT INTO users (user_id, username, password, firstname, lastname, organization_id)
VALUES
  ('11111111-1111-1111-1111-111111111112', 'gbrkic', 'password1', 'Goran', 'Brkić', '11111111-1111-1111-1111-111111111111'),
  ('11111111-1111-1111-1111-111111111113', 'ancil506', 'password2', 'Ana', 'Brkić', '11111111-1111-1111-1111-111111111111'),
  ('22222222-2222-2222-2222-222222222223', 'ficofilip', 'password3', 'Filip', 'Botički', '22222222-2222-2222-2222-222222222222');

-- Insert dummy events
INSERT INTO events (event_id, event_name, organization_id)
VALUES
  ('11111111-1111-1111-1111-111111111114', 'Event 1', '11111111-1111-1111-1111-111111111111'),
  ('11111111-1111-1111-1111-111111111115', 'Event 2', '11111111-1111-1111-1111-111111111111'),
  ('22222222-2222-2222-2222-222222222224', 'Event 3', '22222222-2222-2222-2222-222222222222');

-- Insert dummy userEvents
INSERT INTO userEvents (user_id, event_id)
VALUES
  ('11111111-1111-1111-1111-111111111112', '11111111-1111-1111-1111-111111111114'),
  ('11111111-1111-1111-1111-111111111113', '11111111-1111-1111-1111-111111111114'),
  ('22222222-2222-2222-2222-222222222223', '22222222-2222-2222-2222-222222222224');

-- Insert dummy meetings
INSERT INTO meetings (meeting_id, event_id, user_id, meeting_date, meeting_time, title, description)
VALUES
  ('11111111-1111-1111-1111-111111111116', '11111111-1111-1111-1111-111111111114', '11111111-1111-1111-1111-111111111112', '2023-06-01', '10:00:00', 'Meeting 1', 'This is Meeting 1'),
  ('11111111-1111-1111-1111-111111111117', '11111111-1111-1111-1111-111111111115', '11111111-1111-1111-1111-111111111112', NULL, NULL, 'Meeting 2', 'This is Meeting 2'),
  ('22222222-2222-2222-2222-222222222225', '22222222-2222-2222-2222-222222222224', '22222222-2222-2222-2222-222222222223', NULL, NULL, 'Meeting 3', 'This is Meeting 3');

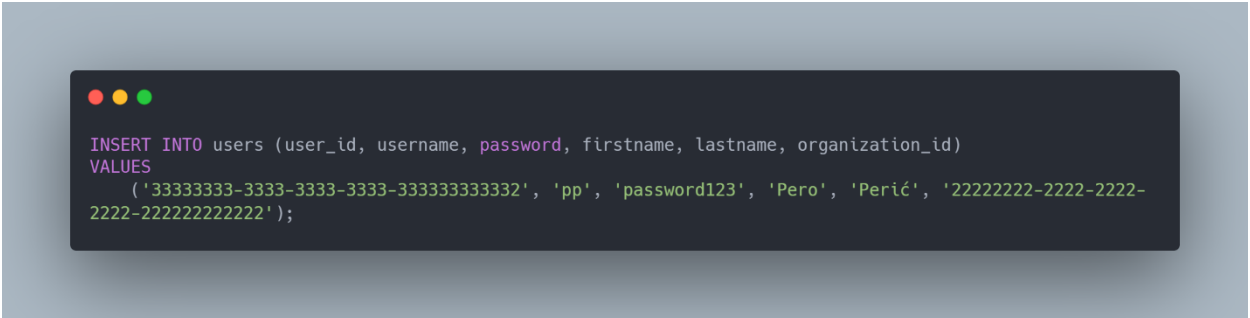
-- Insert dummy invitations
INSERT INTO invitations (invitation_id, meeting_id, invitee_id, status, message)
VALUES
  ('11111111-1111-1111-1111-111111111118', '11111111-1111-1111-1111-111111111116', '11111111-1111-1111-1111-111111111113', 'Pending', 'You are invited to Meeting 1'),
  ('11111111-1111-1111-1111-111111111119', '11111111-1111-1111-1111-111111111117', '22222222-2222-2222-2222-222222222223', 'Accepted', 'You are invited to Meeting 2');

```

I used Jest for testing.

At the time of writing this I have deleted a userevent entry for user with the id of '11111111-1111-1111-1111-111111111112' so that when you run 'npx jest' you test confirms 2 correct tests.

I also added a new dummy user for you to manipulate with, id '33333333-3333-3333-3333-333333333332'. He's not in any other table so you should be able to add him anywhere if you were to do some testing.



```
INSERT INTO users (user_id, username, password, firstname, lastname, organization_id)
VALUES
('33333333-3333-3333-3333-333333333332', 'pp', 'password123', 'Pero', 'Perić', '22222222-2222-2222-2222-222222222222');
```

When testing the GET all scheduled meetings for a user, it will return only meetings that are scheduled. So for user 11111111-1111-1111-1111-111111111112 1 meeting will return as 1 of the 2 are scheduled, and none will return for 22222222-2222-2222-2222-222222222223.

The following image shows the idea for the architectural design.



Frontend UI



API Gateway / Proxy



Authentication Service



Authorization Service



API

(User, Organization, Event, Meeting, Invitation APIs)



Database

Presentation Layer

Business Logic Layer

Data Access Layer