Details of Shem-Tov Shapiro’s completion of the assignment

Original instructions:

Goal : Create a cloud api available in public

1. Create a cloud environment with a database (AWS - Azure - Gcloud - Firebase)
2. Create a login system with a jwt token
3. Create 3 table/collections persons, animals, membership

Technical review:

Persons is a collection of unique people, which have an unique id, firstName, lastName, phoneNumber, city, country.

Animals is a collection of unique animals, which have unique id, name, species

Membership is a collection of relationships between a person and animals.

One person can have many animals

One animal can be with only one person

1. Create apis
   1. Person : Create/Delete/Retrieve
   2. Animals : Create/Update/Retrieve
   3. Create a membership between a person and an animal
   4. Retrieve a membership between a person and his animals
   5. All api need to be secured and verified by jwt token
2. Your apis need to be available via postman that you share with us and your code needs to be available in git. The name of the repository needs to be your fullName\_balink.

Bonus :

\_ use microservice architecture with serverless function (highly recommended to do)

\_ use Algolia or Elastic Search for a secondary database

Details from Shem-Tov

1. I solved it alone with the help of the internet.
2. I created Public Facing APIs (connected to a database) on AWS
3. The environment I created is for LAMP (linux, etc.)
4. I did add SSL for security, but did not use a jwt token. I did so to make things a little easier, so that I could concentrate on the other parts of the exercise.
5. The application and database are running on an AWS EC2 instance
6. ssh to the EC2 instance is secured by a .pem Key pair generated by AWS
7. Database = MariaDB running on the EC2. I called it species.
   1. The animals table has id as a primary key and is an autoincrement field
   2. Same for the persons table
   3. membership table has animals\_id as its primary key, and persons\_id as a regular not null column. It also has a foreign key from the animals\_id column into the animals table. This is what implements the one to many relationship, as requested by the exercise.
   4. It has another foreign key on the persons\_id to insure data integrity.
   5. The database credentials are stored in a env.ini file
   6. I make use of the AWS parameter store to combine the DB password with a secret key, to add a layer of security.
8. I created an Elastic IP so that the IP is static, and we don’t have to worry about it changing.
9. I connected the IP to the EC2 instance.
10. I registered a domain and connected it to the static IP
11. I applied an SSL to the site including a valid certificate.
12. The url for the application is shemtov-shapiro.com .
13. The full code is in the repository but the only two important folders are
    1. html/api/posts
    2. repos (and only the file post.php
    3. core/auth.php
    4. The file env.ini
14. The DB code is in balink-shemtov\_schema.sql
15. The name of the repository is shemtovShapiro\_balink