# Mini Assignment

**Prerequisites**

**Software(s)**

1. VS Code/ PyCharm
2. Python
3. Flask
4. Local DynamoDB - [Deploying DynamoDB Locally on Your Computer - Amazon DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DynamoDBLocal.DownloadingAndRunning.html)
5. AWS CLI
6. Connect to Local DynomoDB

**Plugins**

1. Sonarlint
2. AWS toolkit – Optional

**Hands-on**

1. Create Flask application
2. Create a connection to Dynamo DB using AWS SDK (Use local DynamoDB)
3. Create tables and CRUD REST APIs to the Dynamo DB
4. REST APIs can be accessed only when correct JWT token is provided.

**Problem statement:**

Read the data from the CSV file (which contains the IMDB movie data) and store it in DynamoDB

*Input*: should accept the argument in the command line | REST APIs

*Output*: Print output of params: Title, published year, budget, user reviews, country, genre, duration

**No extra library should be used for the assignment**

1. Titles directed by given director in the given year range e.g : generate titles report for director D.W. Griffith and year range 2010 to 2020

2.Generate report of English titles which have user reviews more than given user review filter and sort the report with user reviews by descending

3.Generate highest budget titles for the given year and country filters

4.Create a TimingMiddleware that prints time taken to execute each request. Return the time taken in a “X-TIME-TO-EXECUTE” header on the response.

5.Create an AuthTokenMiddleware that returns a 401, unless a “x-access-token” header is present on the request.

6.Sync new additions from the file to each datastore.( Every new movie details in the file should be synced to database (or other datastores, if necessary) without any intervention by the user (Calling APIs, hitting command etc).)

**Coding standards expected:**

1. Exception handling
2. Adequate comments
3. Logging
4. Proper project structure