**Lab3 - Express server**

**1- Provide REST API implementing CRUD operations on Employees**

* Use a filesystem, same as Lab01.
* You must use middleware to validate inputs.
* You must return descriptive status codes in case of errors.
* Routes:
  + - GET /employees
    - GET /employees/${id}
    - POST /employees
    - DELETE /employees/${id}
    - PATCH /employees/${id}

**2- Provide a Static files server**

* Download this [image](https://i0.wp.com/www.kingtutballoons.com/wp-content/uploads/2016/06/balloon.jpg?fit=1800%2C1200&ssl=1) and serve it using express static.

3- **Serve SSR html page on route GET /**

* Render the current employees list in styled components
* Serve css from separate file

**Bonus:**

1- Can filter employees using queryparam (one filter)

Ex: /employees?name=”ahmed”&salary=10000&...

**Must read before Day4:**

* [**Introduction to Mongoose for MongoDB**](https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/)
* [**REST API Best Practices – REST Endpoint Design Examples**](https://www.freecodecamp.org/news/rest-api-best-practices-rest-endpoint-design-examples/)
* [**Authorization vs Authentication**](https://stackoverflow.com/questions/6556522/authentication-versus-authorization)
* [**Hashing vs. Encryption vs. Encoding**](https://cheapsslsecurity.com/blog/explained-hashing-vs-encryption-vs-encoding/)
* [**JWT**](https://jwt.io/introduction/)
* [**CORS**](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS)