**MY TRAINING REPORT**

1/ Git

Các lệnh thường dùng:

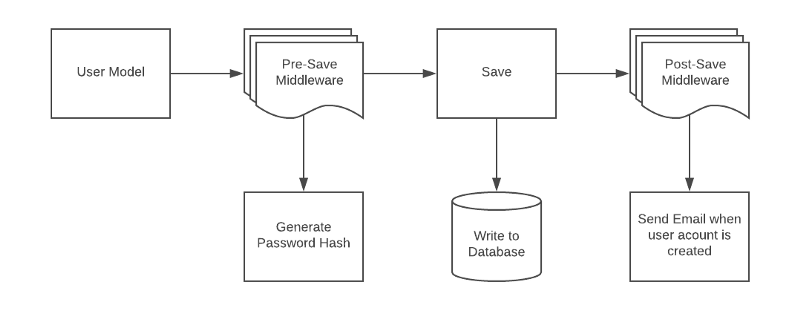
* Init : khởi tạo git
* Status: kiểm tra trạng thái git (3 trạng thái woking directoty | staging | repository)
* Add : thêm 1 file hoặc . để thêm tất cả files
* Commit: lưu thay đổi của repository
* Commit –amend : dùng để combine 2 commit lại
* Push: cập nhật ref
* Push –force : overwrite all history
* Log
* Branch: liệt kê nhánh, tạo nhánh mới với tên
* Checkout: chuyển nhánh
* Merge : merge code từ nhánh này qua nhánh kia
* Branch –d: xóa branch
* Rebase : move to master
* Reset
* Pull request (github)
* Cherry-pick: copy commit

2/ Typescript:

* Cơ chế khai báo biến
* Kiểu dữ liệu
* Class
* Interface
* Sự khác biệt của type và interface
* Một số khái niệm: tham số sau có thể lấy giá trị của tham số trước

3/ Mongodb – Mongoose:

* Connect to db
* Create model
* Middleware: (models have pre and post functions)



* CRUD operation
* Query

VD : UserModel.find() // find all users

.skip(100) // skip the first 100 items

.limit(10) // limit to 10 items

.sort({firstName: 1} // sort ascending by firstName

.select({firstName: true} // select firstName only

.exec() // execute the query

.then(docs => {

console.log(docs)

})

.catch(err => {

console.error(err)

})

* **NOTE! A few features only available on cloud**

**Data Foundation:**

* MongoDB Atlas: cloud database for modern applications
* Atlas Search
* Atlas Data Lake: a serverless, scalable data lake
* Realm Database: The data foundation is extended to the edge by Realm Mobile Database, a lightweight database embedded on the client

**Platform Integrations:**

* Management
* Security
* Data Connectors
* Development Tools

**Application Services**

* **Entity: thực thể có những thuộc tính. ⬄ row (mysql)**

VD thực thể học sinh có thuộc tính tên, lớp

* Nguyễn văn a, lớp 10 là 1 instance of entity.

4/ Hapi:

* Create server, plugin
* CRUD
* Write api-doc by swagger-ui
* Using Joi to validate

5/ Agenda:

* Create job to check database every minutes

6/ Bull:

* Using bull, redis, hapi to handle the master/worker processing

7/ Practice 2.3.1 : Voucher application

* Read db transactions
* Create event and voucher collection
* An endpoint to request voucher code (vouchers table) against an event (events table)
* Sending email

8/ Practice 2.3.2: