Terminology

Server:

Runtime environment is the environment in which a program or application is executed. It's the hardware and software infrastructure that supports the running of a particular codebase in real time. Typically the runtime system will have some responsibility for setting up and managing the stack and heap, and may include features such as garbage collection, threads or other dynamic features built into the language.

npm (node package manager) is the package manager for the Node JavaScript platform

yarn is a software packaging system developed in 2016 by Facebook for Node.js JavaScript runtime environment that provides speed, consistency, stability, and security as an alternative to npm (package manager).

Node.js is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. The runtime is intended for use outside of a browser context (i.e. running directly on a computer or server OS). As such, the environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries.

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a *library* that implements the Object-Relational Mapping technique, hence the phrase "an ORM". An ORM library is a completely ordinary library written in your language of choice that encapsulates the code needed to manipulate the data, so you don't use SQL anymore; you interact directly with an object in the same language you're using.

Micro ORM is basically mapper that creates objects based on database query. How we are going to interact with database.

express is a Node Package Manager (npm) module (a.k.a. a package). it builds a rest api. The purpose of the express is that you don't have to repeat same code over and over again. Node.js is a low-level I/O mechanism which has an HTTP module. If you just use an HTTP module, a lot of work like parsing the payload, cookies, storing sessions (in memory or in Redis), selecting the right route pattern based on regular expressions will have to be re-implemented. With Express.js, it is just there for you to use.

apollo-server-express is the Apollo Server package for Express, the most popular Node.js web framework. It enables you to attach a GraphQL server to an existing Express server.

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.

GraphQL is not an ORM, because it doesn't understand the concept of DBs. It just gets the data from a "data source", which could be static, from a file, etc. Nor can it figure out how to get data once you point the source at it. You have to write resolver functions that tell the DB how to find the value of each field.

It's also not a REST replacement, rather an alternative, and you can utilize both in the same project!

GraphQL is an opinionated API spec where both the server and client buy into a schema format and querying format. Based on this, they can provide multiple advanced features, such as utilities for caching data, auto-generation of React Hooks based on operations, and optimistic mutations.

GraphQL resolver is a collection of functions that generate response for a GraphQL query. In simple terms, a resolver acts as a GraphQL query handler.

Client:

next.js is a React framework for developing single page Javascript applications. (https://snipcart.com/blog/next-js-vs-react)

chakra_ui is a simple, modular and accessible component library that gives you the building blocks you need to build your React applications.
urql is a lightweight, extensible GraphQL client for React. It is built to be easy to use, performant and functional, logical default behaviour and caching and easily extensible.

Basic setting for server

- **npm init -y** // initiate a project (create package.json)
- **yarn add -D @types/node typescript** // add typescript and node and typescript behave each other so add dependency
- Add script in package.json
  - **ts-node** typescript execution engine and REPL for Node.js. But this is slow, and in the production, usually they compile to javascript and run with node.
  - **nodemon** listens for changes in files and reruns
  - **tsc-w** take ts file and make js file in dist folder
- **yarn add -D nodemon** // How to install nodemon
- The set up for node and typescript
  - **yarn watch** = tsc -w in one terminal recompile type http://file3.instiz.net/data/cached_img/upload/2022/04/25/12/83334e61b1c7e855e01320a8728ccc6d.jpg script to javascript file
  - **yarn dev** = nodemon in the other terminal re-execute the updated javascript file

Postgresql (in separate terminal)

- https://www.sqlshack.com/setting-up-a-postgresql-database-on-mac/
- **brew services start postgresql**
- **brew services stop postgresql**
- **psql -U postgres** // log in database with username postgres
- password: Juho19971106
- **CREATE DATABASE lireddit;** // create database

Mikro-orm

- **yarn add @mikro-orm/cli @mikro-orm/core @mikro-orm/migrations @mikro-orm/postgresql pg** //How to install micro ORM version 4 for postgresql
- command line (CLI):
  - **npx mikro-orm migration:create** # Create new migration with current schema diff
  - **npx mikro-orm migration:up** # Migrate up to the latest version
  - **npx mikro-orm migration:down** # Migrate one step down
  - **npx mikro-orm migration:list** # List all executed migrations
  - **npx mikro-orm migration:pending** # List all pending migrations
  - **npx mikro-orm migration:fresh** # Drop the database and migrate up to the latest version

Create server

Graphql

- **yarn add express apollo-server-express graphql type-graphql** //add few dependencies. Express is the server we are going to be using. apollo-server-express is a package that allow us to use graphql and create graphql endpoint very easily.
- **yarn add -D @types/express** //Install typescript type for express
- **yarn add reflect-metadata** //library that is used somewhere
- **yarn add argon2** //password hashing https://github.com/ranisalt/node-argon2