Stack Overflow

# Intro

As we kept moving forward in the 21st century, more and more of our solutions to daily problems have been relying on modern software. The building blocks used in creating software has in turn also suffered major changes as to accommodate the many different type of applications. With this new approach to problem solving, we managed to create a new problem, and that is what happens when we don’t know how to use or misuse these building blocks. To combat these 21st century problems, we will come up with a 21st century solution, the solution is to create a community where users can place questions and other users can answer them thus helping in building better and more reliable software.

# Tech Stack

The technology that makes this app a reality is node.js + express.js for the backend and MySQL and TypeORM for the data base. Since this is a web app, this allows for easy cross platform development as most modern devices now have the needed requirement, a web browser, to run this application,

# Use Case Example

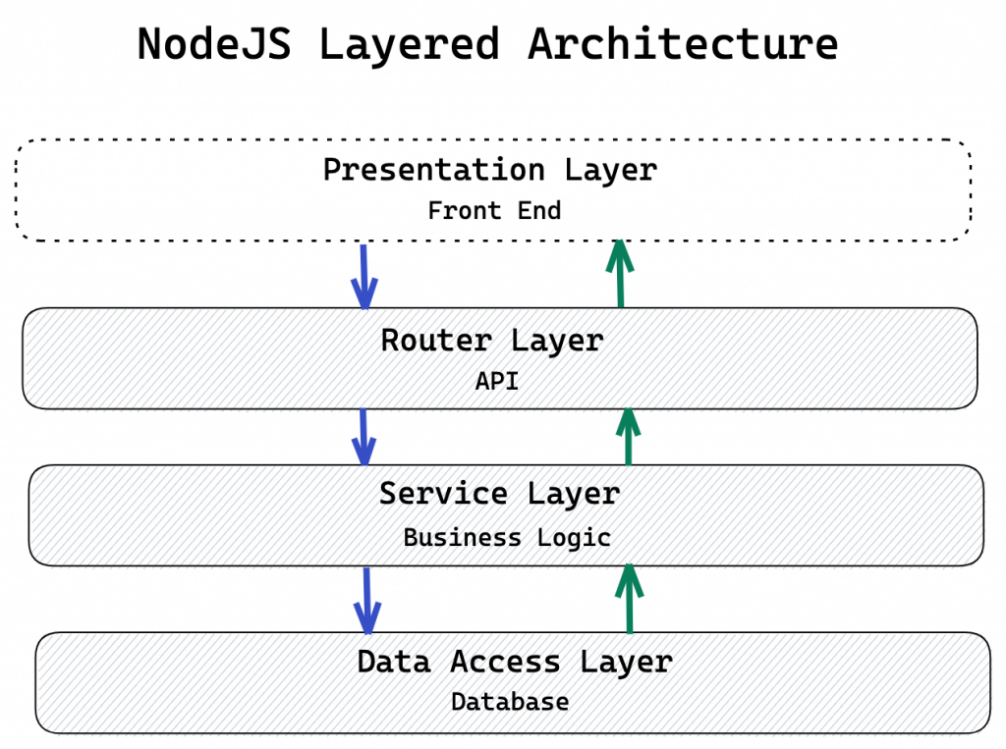
In the following diagram we will represent some use cases, followed by a description of the login use case:

# 

I will describe the way that the login use case works. The client creates a post request containing his email and password in the body. The route handler will then pass the data along to the service layer. The service layer will then try to retrieve the user using the data access layer, if user is successfully retrieved, then the given password is compared to the password in the data base, if the passwords match, then the service layer will create a JWT token and store it using the data access layer. The created token is return to the route handler, and further on to the user who created the request. If any of the steps fail, the message “Login failed”, will be return to the user.

# Architecture

Behind all great apps, there are great architectures, and this app is no different. This app will feature the layered architecture.



As can be seen in the above diagram, this architecture is made up of 4 layers, 3 for the backend and 1 for the frontend. The router layer holds the endpoints and routes the data to the service layer which manipulates the data. The data access layer is then used to save the data or retrieve new data. The reworked items are then sent back up all the way to the presentation layer.

# Database diagram

The following diagram represents our database structure

A picture containing graphical user interface

Description automatically generated

# Bibliography

<https://app.diagrams.net/>

<https://ctrly.blog/nodejs-layered-architecture/>

<https://dev.to/santypk4/bulletproof-node-js-project-architecture-4epf#architecture>

<http://expressjs.com/en/guide/using-middleware.html#middleware.application>

<https://www.section.io/engineering-education/speakeasy-two-factor-authentication-in-nodejs/>

<https://typeorm.io/#/>