Thai Tanic

Thai food for the easy minded built with Ionic and Strapi

Tech Stack

Tech	Version	Comment
Ionic	5	Used as framework to build PWA, mobile and or just plain web apps
Angular	11	Used under the hood for framework of ionic
Strapi	3.5.4	Used as backend
Node	14.15	
NPM	6.14	Could also use yarn for more granular package management
Yarn	1.22.10	For strapi
Auth0	5.0.2	
Formly	5.10.17	Used for easier use of forms (Scaled to be use json forms)
Docker	20.10	Used for deployment
Docker Compose	128	Used for easy development of backend with PostgresSQL
Git		Used for version control
Github		Used for hosting
PostgreSQL	12	Used as backend for project

Frontend setup

NOTE: Please make sure you are in the root of the project folder/

1. Execute the following command in the terminal of your choice `npm install`
2. To start the server you can run `ionic serve` PS: If you want to see how it looks for mobile you can run `ionic serve --lab`

Unit Testing frontend

1. Please run `ng test` to run automated tests with Karma

Running Deployed app

If you want to automate most and don't want to run the ionic server you can run the provided docker image i created

1. Please run `docker build -t thaitanic:latest .` PS: You can rename `thaitanic` to anything you want
2. Once completed you can run this by executing `docker run -d --rm -p 80:80 thaitanic:latest` PS: If you renamed it please replace `thaitanic` with anything you want it to be
3. Please visit `http://localhost` and you should be presented with the front end. PS: To make sure it's fully working and to login please make sure the backend is also setup.

Backend setup

To run and work with the backend please make sure you are int he `backend` folder.

Note here, currently the choise of using postgres would be that this could be a big store which would require relationships for data. For this demo it could also be used as a single mongoDB but i chose Postgres due speed in production env etc. Again the use of strapi would let me use `sqlite` `PostgreSQL` `MongoDB` `MySQL` `MariaDB`

A NoSQL database would be great for this mini project but i prefer here to use postgres.

1. Please cd into `backend`

2. Please create a `.env` (Where the `docker-compse.yml` is located) to hold out docker files (This could also been used with secrets file but for simplicty i chose to use this) PS: If not using docker we could use the `.env` directy with strapi located inside app.

3. Paste the content below and replacing XX with random strings PS: You can use `openssl rand -base64 32` to generate the random string

```
DATABASE_CLIENT=postgres
DATABASE_NAME=strapi
DATABASE_HOST=postgres
DATABASE_PORT=5432
DATABASE_USERNAME=strapi
DATABASE_PASSWORD=REPLACE ME WITH BASE 64 random string
SECRETKEY=REPLACE ME WITH BASE 64 random string
JWT_SECRET=REPLACE ME WITH BASE 64 random string
ADMIN_JWT_SECRET=REPLACE ME WITH BASE 64 random string
ENVIRONMENT=development
```

4. Please run `docker-compse up -d` this will run in the background. PS: it will take a few minutes first time start up as it will require to download postgres and yarn install and bootstrap the application remove the `-d` to keep an eye on the docker-compose for any errors and first time setup PS: You can also do `docker logs strapi -f` to keep an eye on the logs live

- Coffe Break - Go get a coffee while docker does it's thing

5. Once started you can setup your own admin user by visiting `http://localhost:1337/admin`
6. You can now create a an admin user (Note that in strapi admin user and a normal user are seperated)
7. Permissions needs to be given in Admin Settings for the front end to work correctly
    - Settings > Roles > Authenticated (Press the pencil to edit)
    - Under **Application** > **PRODUCTS** please give all permissions by using **select all** (count is not needed so for security feel free to uncheck this)
    - Under **USERS-PERMISSIONS** > **USER** please give the following permissions:

      ☒ find
      ☒ me
      ☒ findOne
      ☒ update
8. Once completed please scroll to the top and press SAVE

The backend should now be ready to be used so feel free to startup the frontend by checking out Running Deployed app

Tech Test Questions

1. What is the difference between ++a and a++?

    > One will increment the number BEFORE the current expression the other one after.

2. Can you explain what you consider to be the key differences between object-oriented programming and functional programming?

    > OOP is a stateful programming model while functional is stateless

3. What is the difference between a closure, a callback, a lambda, and a promise?

    > Callbacks are functions that are pased into other functions as arugments, Closures are nested in other functions, Mostly used to avoid Scope clashes

4. Explain logic short-circuiting, and how it can affect the code you write.

    > Not sure about this

5. What are your thoughts on composition versus inheritance?

6. How would you choose between using a regular expression, a parser, or a simple string search? Give examples.

   > As i'm mostly used to write JS i prefer regular expressions but i presume lower end langauges and or more algorhyms would have other ideas for speed and what is best.

7. Can you explain how dependency injection helps when writing unit tests? > In simple term it helps in the way that you dont need to modify code of any object

8. Give an example of how you would use defensive programming techniques (other than to sanitise user input).

   > The use of defensive design / programing would apply when getting data or any places where you have async code.

9. Do you think it is good or bad to commit "built" files? (E.g. the output of SCSS, etc.) Explain why.

   > I prefer NOT to commit built files as it pollutes the repo, any developer and or CI/CD can easy build these files. Though some extend i would say that it's nessary if you do not have any build tools (so scss ->) and the deployed version uses the CSS (things like tailwind), i would say it's fine to keep them.

10. When would you use fully-normalised form, and when would you use JSON columns? > Simple simple forms with not much data or state can use simple forms, preference for me is to use JSON powered forms > as this will give me more flexibility, though simple forms will take less time to write if complexity is needed.

11. When would you use a stored procedure and why?

    > Not Sure

12. When is it inadvisable to rely upon ORM? > There might be time for a company having security policies where we are unable to use ORM, but mosty it's a good thing to use ORM

13. What was the mostuseful feature that was added to the latest version of your chosen language? Please include a snippet of code that shows how you've used it. > I would say the Template Literals as types for Typescript 4.1 is quite cool and new.

```
type Suit = "Hearts" | "Diamonds" | "Clubs" | "Spades";
type Rank =
  | "Ace"
  | "Two"
  | "Three"
  | "Four"
  | "Five"
  | "Six"
  | "Seven"
  | "Eight"
  | "Nine"
  | "Ten"
  | "Jack"
  | "Queen"
  | "King";
type Card = `${Rank} of ${Suit}`;
const validCard: Card = "Three of Hearts";
const invalidCard: Card = "Three of Heart"; // Compiler Error
```

14. What is your preferred approach to responsive design? > Mobile first, bigger the screen more the flare.

15. Please describe yourself using JSON.

```
{
  "firstName": "Simen",
  "lastName": "Daehlin",
  "age": 35,
  "email": "simen@dehlin.dev",
  "latestExperience": [
    {
      "title": "Lead Developer",
      "company": "Cropdesk Technologies",
      "whatIDo": "Designing and leading a team of developer where we build agricultural soft
    },
    {
      "title": "Student Mentor",
      "company": "Code Institute",
      "whatIDo": "Mentor new people to code, these are people that might never have touched
    }
  ],
  "hobbies": ["Gaming", "Tech Geek", "Spending time with my wife"]
}
```

- How long did you spend on the coding test?

  Several hours (I prefer do a good job rather then quick smash and grab jobs to have it be redone)

- Did you manage to cover everything that you wanted to?

  As i did spend more time on it then just a few hours yes i did cover most things, i do feel i could do a bit more error handling on things and or adding interceptors to deal with most errors and or loading spinners and feedback to users etc.

- What would you add to your solution if you had more time? > Due scalability of this **AMAZING** app and the Thaitanic reputation **NGRX** store would be next choice when scalability is needed
- Would you choose different technologies if this were to become a reliable enterprise system? Why? Or, why not? > For a mobile app flutter would be good, but for a WEB application any PWA could be good as well

What did i think of the test

A fun test for me, i did enjoy getting my hands dirty with one of my favourite framworks Maybe less of the questions or have the questions before having a tech test to save on some of the time and or if it would fit the team person.

I feel a CRUD application is a great way to have a person show their skills of, it's quite open so fair indeed. It's up to the user to show best practices etc. Overall i did really enjoy it hence the extra hours on it and attention to deployment and details