C.A.R journal

Conflict, Action, Results

Clone of bookings page for hostelworld.com

My module is availability/bookings section

Conflict – Schema Design (for MONGODB)

-Schema design: could not properly mine data from the database. Had all the data on one flat schema.

Actions

* Redesigned Schema several times - eventually realizing that just like in SQL, mongoDB should persist with the concept of one to many relationships, so every data group should have its own dataset. Ended up with a 3 layered schema

Results

* Allowed me to grab data I needed and flexibly allocate it into props at the correct places. Learned that proper database construction plays a huge role in data useability

//if I have time

Conflict – Parsing Dates

After settling on a schema and moving forward to front-end work, I realized that mongoDB dates are stores in an ISOdate format which is read as a simple string in javascript. Due to this mismatch of datatypes, I could not compare the dates of my raw data and filter it to be useable in the frontend.

Actions – Parsing Dates

At first I tried to query by subdocument (due to my schema design, the dates were held within a 2nd layer of nesting, making them a subdocument of a subdocument) but quickly realized that MongoDB/Mongoose can only use subdocuments as conditions to query documents.

I then decided to take all the raw data and, in the front end, parse the strings of dates into useable javascript date objects. I would then filter the data by finding the indexes of the first and last dates specified by the user and only grab data from within those dates.

Results – Parsing Dates

Ended up being much more complex. After grabbing data from within those dates, in order to obtain the max number of reservable beds I had to run reduce on the array of each dataset to find the lower number of available beds in each set.