Chrystal Quek Wan Qi

A0206350U

**Basic use cases**

* Simple Todo application for individual use
* Authenticated using JWT
* Help users keep track of their tasks
* CRUD enabled
* Two models:
  + Users
    - Id
    - Username
    - Email address
    - Password
  + Todos
    - Id
    - Title
    - Description
    - Tag
    - Category
    - User references

**Execution plan**

* Rails and react application stored in different files, uses ports 3001 and 3000 respectively
* Rails is an API only app which uses postgresql (for future deployment to Heroku)
* Rails controller methods when called render json containing user specific todos, status, errors (if any) etc
* Use axios GET PUSH POST to query for information and DELETE to delete information, following predefined routes by rails
* Use local storage to store JWT, decode JWT to obtain user\_id
* Routes
* scope '/api' do
* resources :users, only: [:create] do
* resources :todos
* end
* end
* post "/login", to: "auth#login"
* get "/auto\_login", to: "auth#auto\_login"

**Suggestions/plans (in progress)**

* Styling - drop down lists, datepicker etc using either react or bootstrap
* Stronger and more efficient authentication using JWT-token
* Searching/filtering
* Navigation bar

**Current Issues**

* Previously used session data to authenticate users but only recently realised that this approach was not scalable. Trying to convert to JWT instead.
* Not enough styling
* Spotting inefficiencies