

Improving current features

- **/getUser**: Get a single user based on the query string (this is already there but its in GQL. Mode it to REST)
- **/getAllBoats**: Make this accept parameters -
 - **filters**: shared, completed
 - **sorts**: date, crew size. should also take in asc/dec
 - **search**: searching boat by name. right now it can just be a simple string matching for boat name
 - **page**: pagination. default?
- Search images API (unsplash) is not working

New Features

- **/boats/crew**: GET all crew data for a boat. UPDATE too.
- **/custom-login**: Custom login. Takes in `email` and `password`
- **/sign-up**: Custom Sign up. Takes in `name`, `email`, `password1`, `password2`. Validate password1 and password2, hash and store.
- **/forgot-password**: Take in an email, send back a 200. Generate a 16-dig token (5min exp time), send a link to the entered email (eg: https://sailboat.com/reset-password?token=<16_dig_string>)
- **/reset-password**: Takes in new `password1`, `password2` and `token`. Verify token. Validate, hash, update user obj, and send back 201.

Widget Design

```
interface Boat {  
    // ...  
    widgets: Array<Widget>;  
}  
  
interface Comment {  
    author: string; // id of crew member  
    body: string;  
    likes: Array<string>; // ids of crew members  
}  
  
enum WidgetType {  
    Date = 'date',  
    Location = 'location',  
}
```

```
interface Widget {  
  id: string;  
  order: number; // position of the widget in the frame  
  responses: Array<Crew>; // members that have voted  
  comments: Array<Comment>;  
  /**  
   * this should be user specific.  
   * if they haven't voted this should be true  
  */  
  actionRequired?: boolean;  
  description: string;  
  type: WidgetType;  
  deadline: Date; // will be used to send reminders  
  data: Array<WidgetData>;  
  selected: id | null; // selected data id (by captain)  
}
```

```
interface WidgetData {  
    id: string;  
    option: string;  
    description?: string;  
    votes: Array<string>; // id of crew member  
}
```

APIs

- [PUT] **/boats/widgets**: Add/update a widget to boat
- [GET] **/widget/:id**: Get data of widget by providing the ID. Pass the boatId as well

- [POST] **/widget/vote**: Params - `widgetId`, `widgetDataId`, `boatId`, `crewId`
- [POST] **/widget/select**: Params - `widgetId`, `widgetDataId`, `boatId`.
Update `selected` field in widget
- [POST] **/widget/comment**: Post a coment to `widgetId` & `boatId`
- [GET] **/widget/comments**: Get all comments of a widget. Use `page` to paginate.