## **PROJECT: FACE-BOOK**

You’ve come a long way, congratulations! At this point you should feel comfortable with building new Express applications and using MongoDB to model and store data. This project will require you to put all of that knowledge to the test. It’s not going to be easy, but should be well within your capabilities and it will be a *great* portfolio piece.

You’ll be building Facebook. As with our previous lessons, how much effort you want to put into the styling and front-end is up to you. The important stuff is the data and backend. You’ll put together the core features of the platform like users, profiles, posts, “liking”, “friending”, and the news feed. You’ll also implement signing in with the real Facebook using our old friend passportJS.

Some features of Facebook are things we haven’t been exposed to such as chat, realtime updates of the newsfeed and notifications. You won’t be responsible for those unless you’re feeling really confident in your skills at this point. (It’s not *that* hard.. [look here](https://socket.io/) if you want to see what’s involved.)

### **Assignment**

Build Facebook! You’ll build a large portion of the core Facebook user functionality in this project. We won’t be worrying about some of the more flashy front-end stuff unless you really want to, but you shouldn’t need it to get a nice user experience.

This project will give you a chance to take a relatively high level set of requirements and turn it into a functioning website. You’ll need to do some of your own research and read the documentation for a few of the modules we’ll be using. Keep the following requirements in mind. We’ll cover specific steps to get started below this list.

1. Users must sign in to see anything except the sign in page.
2. Users should be able to sign in using their real facebook details. This is fairly easily accomplished using PassportJS, and you should be able to use the knowledge you already have to figure it out [from the documentation](http://www.passportjs.org/docs/facebook/).
3. Users can send friend requests to other users.
4. A user must accept the friend request to become friends.
5. Users can create posts. (begin with text only)
6. Users can like posts.
7. Users can comment on posts.
8. Posts should always display with the post content, author, comments and likes.
9. Treat the Posts index page like the real Facebook’s “Timeline” feature – show all the recent posts from the current user and users she is friends with.
10. Users can create Profile with a photo (you can get this from the real facebook when you sign in using passport)
11. The User Show page contains their profile information, profile photo and posts.
12. The Users Index page lists all users and buttons for sending friend requests to those who are not already friends or who don’t already have a pending request.
13. Deploy your app to Heroku!

#### **Extra Credit**

1. Make posts also allow images (either just via a url, or by uploading one.)
2. Allow Users to upload and update their own profile photo.
3. Make it pretty!

#### **Getting Started**

1. Think through the data architecture required to make this work. There are lots of models and the relationship between them is more complicated than anything you’ve done before. How are you going to model a user’s list of friends and friend requests? Posts should be able to have likes and comments associated with them, how are you going to model that? Take some time to plan your approach before diving in.
2. Start your app however you like, using the express-generator or from scratch.
3. Work your way down the list above! Each step will involve a new challenge, but you’ve got the tools.
4. You can populate data like users and posts with fake data using the [Faker](https://github.com/marak/Faker.js/) module from npm. To accomplish this, create a new JavaScript file named seeds.js which imports your mongoose models and uses the faker module to generate and save a bunch of new users.

### **Examples**

* <https://brackenl.github.io/odinbook_frontend/#/login>
* <https://nijepa.github.io/odinbook-frontend/#/>
* <https://fcloneodin.herokuapp.com/#/>
* <https://gentle-savannah-57372.herokuapp.com/>