**Backend**

Use **POSTMAN** (<https://www.getpostman.com/>) for testing the methods. Send requests in JSON format and see the response

**Events API** (./server/routes/events\_api.js)

* Create event method
* Delete event method
* Return event by \_id method
* Get list of events from list of \_ids method
* Get list of events close to a location

**Providers API** (./server/routes/providers\_api.js)

* Register provider method
* Getstatus method (similar user\_api.js)
* Login provider
* Logout provider
* Change fields methods (change logo,change description etc)
* Get provider info by \_id method

**Admin API**

* Delete user, delete provider, reset password methods etc
* Admin login, logout
* How do we manually create admins

**PAYMENTS**

* How do we handle this - no idea.

**Frontend**

* HTML and controller for Event page with placeholders
* HTML and controller for User and Provider profile page with placeholders
* HTML and controller for events\_location page with placeholders (and pins on the map for nearby events)
* HTML and controllers for provider login, register, logout
* HTML and controllers for admin\_providers and admin\_events pages (similar to admin page (should really be called admin\_users))
* Anyone can see any provider’s page, so provider profile url has to have id parameter
* Placeholder service for provider login/logout/register/isLogged in (until provider\_api is ready)
* Placeholder service for events (will make all api calls to events\_api)
* Placeholder logic extension for header\_nav (for if provider or admin is logged in)
* IMPROVE ALL CSS AND HTML for mobile friendliness

**OTHER**

* SEARCH (front-end or back-end?. Depends)
* Node module for watermarking