1 PROJECT DOCUMENTATION

1.1 Hardware & Software Requirements

1.1.1 Hardware Prerequisites

The MVP Platform is a web application with search and user functionalities requiring a database. As such, it requires a decently powerful machine.
The platform has been successfully used on a machine with

1.1.2 Software Prerequisites

To begin setup of the MVP Platform, there are a few things that need to be installed.

- NodeJS and NPM: NodeJS is the runtime for the application itself, and NPM is the package manager included with it for installing 3rd-party libraries.
- MongoDB: This is the database used to store things like user favorites and login tokens.
- ElasticSearch: this is the search engine behind the search feature.

1.2 Installation Instructions

1.2.1 Setup

1) Clone the source code:
```
$ git clone https://github.com/CS461-MVP/mvp_platform.git
```
2) Move to the source code directory:
```
$ cd mvp_platform/dev
```
3) Install all scrap library dependencies:
```
$ cd scrapjs && npm install && cd ..
```
4) Install all backend Node dependencies:
```
$ cd backend && npm install && cd ..
```
5) Install frontend dependencies:
```
$ cd frontend && npm install && jspm install && cd ..
```

1.2.2 Running the backend

To run the server, simply run:
```
$ npm start
```

This will start the server running on localhost, port 8000. To verify that the server is running, in another terminal window run:
```
$ curl http://localhost:8000
```

This will return a 404, as there is no root page, if the server is successfully running.

1.2.3 Running the frontend

The frontend is a static, single-page webapp, but must first be built. To build it, run `gulp build`. To serve the resultant static site, a standard `nginx` or `Apache` static file server is recommended; alternatively, for testing, the pages can be served with Python's built-in HTTP server:

```
$ cd /path/to/clone/mvp_platform/dev/frontend
$ python -m SimpleHTTPServer 8080
```

The website may now be visited at `http://localhost:8080`.

1.2.4 Deploying for Production

For detailed instructions regarding setting up `Nginx`, creating a service, and setting up a reverse proxy, please find the nearest sysadmin. Such a setup is outside of the scope of this documentation. In brief:

1) Create a service that runs the backend as shown above.
2) Install `nginx`.
3) Create a configuration file that:
    - Create a rule that matches requests sent to `/accounts`, `/login`, `/favorites`, `/books`, `/chapters`, `/scraps`, `/images`, `/search`, `/users`. In this rule, reverse proxy the request to the instance of the platform running on port 8000.
    - For all other requests, serve static files from `/dev/frontend`.
4) Restart `nginx` to put the rule into effect.