

Max Objects Database

Using last terms project and the mock-ups provided by Professor Manzo, a redesign of the front-end was done using Bootstrap, a “CSS framework directed at responsive, mobile-first front-end web development.” Bootstrap was chosen for its simplicity, speed and rich component set.

Views were initially written using React, which requires knowledge of JavaScript and HTML. The team opted to use the Handlebars templating engine express-handlebars to create dynamic page views. It was chosen because it requires knowledge of html and a small set of easy to remember helper functions. This will allow future groups who may not have programming skills to easily add new pages.

Handlebars has a useful concept for creating reusable widgets called partials. Partials can be embedded in other templates.

Handlebars partial for representing the news widget:

```
<div class="container py-2">
  <h1 class="widgetHeader py-2 fw-bold text-center">News</h1>

  <div class="card mb-1">
    <div class="card-body">
      <p class="card-text pl-4">{{news.body}}</p>
      <p class="card-text"><small class="text-muted">Last updated {{news.date}}</small></p>
    </div>
  </div>
</div>
```

Partial added to the Max Objects Database homepage:

```
<div class="container mt-4">
  <div class="row row-cols-1 row-cols-sm-1 row-cols-md-2">
    <div class="col mb-4">
      {{> news}}
    </div>
    <div class="col mb-4">
      {{> featured}}
    </div>
  </div>
</div>
```

Server code to render the home view and news partial:

```
router.get('/', async (req, res) => {
  if (req.user) {
    res.render('home', {
      home:true,
      featuredItems: featuredItems,
      news: await db.getNews(),
      user: req.user
    })
  }
})
```

The server was written using Express a “minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.”

Routes were identified and broken down into express routers which helped organize and encapsulate server routes. Routes were added for administration but need to be implemented.

This project used PostgreSQL. A library for node called pg-promise was used to interact with the databases, which were hosted locally and on Digital Ocean. A new schema needs to be

developed to match the data requirements of the new site and then data must be migrated. The databases for the new site and the old site can be found in the databases folder.

For more information, dependencies, and a roadmap for future development, please visit the [README](#) for this project.