

The background of the slide is a semi-transparent red overlay on a photograph of the Iowa State University campus. The image shows several large, classical-style buildings with domes and columns, surrounded by trees. The text is overlaid on this background.

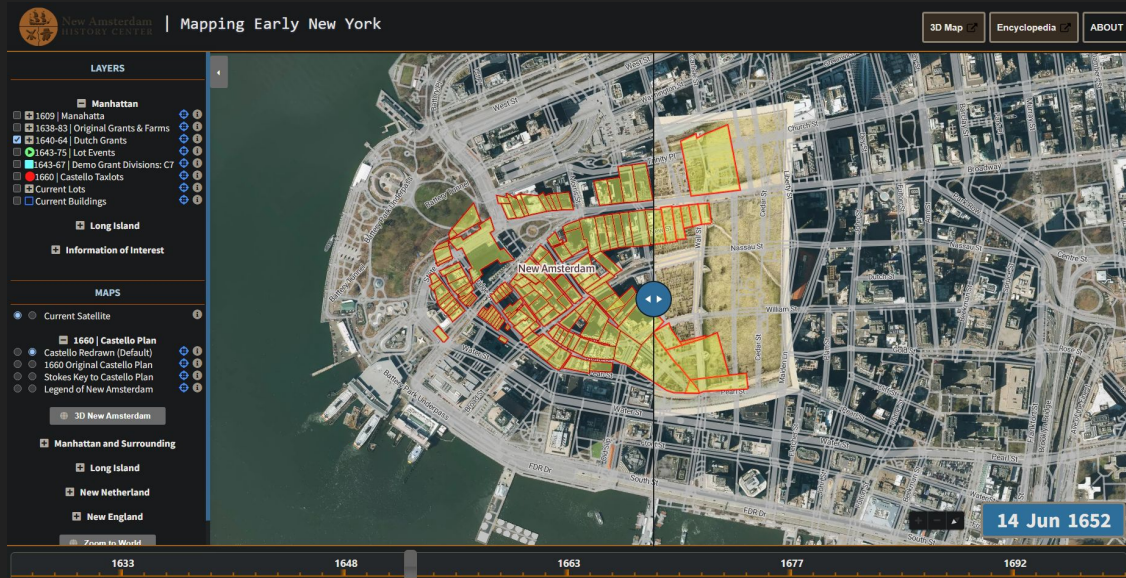
IOWA STATE UNIVERSITY

Group 8 web-map

Isaac, Zak, Haakon, Anbu, Abhi

The Product - Mapping Early New York (MENY)

- This app allows for the user to add maps from various times.
- They can then annotate these maps with areas, points, and add notes associated with these annotations.



The Problem Being Addressed

- Implementing an alternative method to generate and edit maps in MENY
- Our client is struggling with his current method which is done by manually adding json objects to his frontend files.
- There is no existing database for maps/filters/etc. And there is no way for a user to add data from the actual app
- Our job is to make it easier for the client and his devs to maintain and continue the project by converting the project to a more maintainable stack and adding in ways to create maps/layers/notations from the app.

Client Goals

- Document existing code.
 - Written overview
 - Short [video](#) explanation on layer duplication
- A new menu to add/edit maps.
- A new menu to add/edit layers.
- More maintainable codebase / database.



Add Layer

Name	<input type="text"/>
Source name	<input type="text"/>
ID	<input type="text"/>
Type	<input type="text" value="▼"/>
Database	<input type="text"/>
Group	<input type="text"/>
Hover	<input checked="" type="checkbox"/>
Click	<input checked="" type="checkbox"/>
Sidebar	<input type="checkbox"/>
Slider	<input checked="" type="checkbox"/>

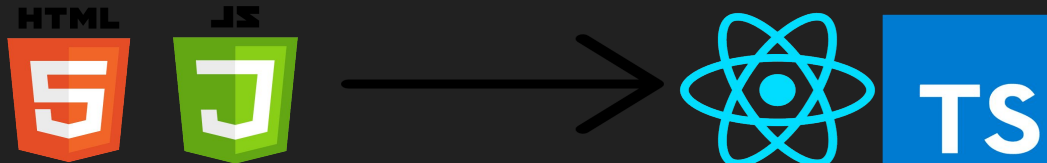
Rough starter form mockup, full field list can be found [here](#)

Forms Requirements

- Create editing page driven by login system
- Add, edit and remove layers/maps with buttons, **without the need to code**
- Create features (points, lines, polygons), without the need for GIS programs
- Adding a data slider, and connecting to the added data
- Adding information for each feature
- Formatting the appearance of information
- Uploading data directly, without the need for an external site

Challenges

- Understanding how the program works
 - We have been given a functional project that we are modifying
 - Project exclusively uses html/javascript and is driven by a set of about 30 javascript files
 - Little to no documentation and javascript is inherently difficult to read
- Conversion from HTML and Javascript to React and Typescript
 - Mapbox doesn't have a package for Typescript or React, had to do some unique dynamic package loading
 - All styling and animations are done with JQuery (We have had to remove this as we go)
 - Converting in tandem with producing deliverables and keeping deadlines with client
- Adding onto an existing project
 - Current project frontend has a range of functionality that needs to stay
 - Backend functionality and methodology needs to remain the same with added endpoints
 - Current development is done and controlled by two across seas developers



Before (This is < 1/3rd of the Code)

After


Go to file

- previous versions
- v_0.2
- v_0.3
- mapping early new york reducing
 - css
 - frontend
 - node_modules
 - .gitignore
 - archived.html
 - editor.html
 - eventHandlers.js
 - index.html
 - index.js
 - layerManagement.js
 - location-active.png
 - location.png
 - mapConfig.js
 - package-lock.json
 - package.json
 - projectManagement.js
 - styles.css
 - uiManipulation.js
 - utils.js
 - icons
 - js

Name




- ..
- addMapLayer.js
- generateLayers.js
- generateMaps.js
- getLayer.js
- google-analytics.js
- handle-mobile-devices.js
- index.js
- layerFlags.js
- layerSectionData.js
- layersFunctions.js
- layersList.js
- mapData.js
- mapinit.js
- mobile.js
- modalinfo.js
- popupContent.bak.js
- popupContent.js
- popupDisplayFunctions.js

main ▾ grp_8_wep-map / MapstructorApp / meny_react / app /

 **janky date filtering via existing javascript**
zfleisch authored 2 hours ago

Name	Last commit
..	
api	Map API don
components	merge bran
fonts	feat: adding
helpers	feat: adding
models	layers rende
compare.css	done
favicon.ico	feat: adding
globals.css	Merge bran
layout.tsx	feat: adding
page.tsx	janky date fil
slider-timeline-date.css	handle color

Name	Last commit
..	
forms	layer for
layers	layers re
map-filters	feat: add
map	merge br
maps	layers re
right-info-bar/popups	get rid of
slider	slight fix

Name
..
 section-layer-group-item.component.tsx
 section-layer-group.component.tsx
 section-layer.component.tsx

Issues

- **Firestore access**
 - Getting the access to their firestore database
- **Rebasing the whole old project into React + Typescript**
 - Compare tool
 - Data slider
 - Layer filters
 - Map filters
 - Map rendering
 - Layer rendering
 - Info panel popups
- **Form development**
 - Map/Layer form
 - i. Backend map/layer mockup
 - ii. Frontend map/layer service
- [Current timeline spreadsheet](#)

Frontend Practices & Tools

- Refactoring this codebase: https://github.com/MapStructor/mapping_early_new_york
- Javascript (Migrating away)
- React / Typescript
 - Improves maintainability & scalability
 - Organized UI elements into react components
- MapBox GL
 - Package used to manage Map-related tools such as compare
 - Utilizes Javascript and Typescript
- Formik
 - Open source react library that simplifies the process of building and managing forms
 - Package to allow for the user to create a form and send it to the backend server.
 - useFormik hook creates a form for adding a new layer
 - USER: Define Initial values and submission behavior

Frontend Status + Remaining Work

- Very close to completing Typescript conversion
 - Small Styling changes / minor functionality items.
- Have a basic map form & layer form implemented
 - Need to get finalized form fields / create associated models / connect to endpoints
 - [Final form fields list](#)
- In our weekly meetings with the client, we find out:
 - Our short-term timeline and what we need done
 - Information on the next parts of our project

Backend Practices & Tools

- Goal is to create a Form
- Next.js w/ Typescript
 - Rest API
- Firebase
 - Hosting
 - Authentication
- MySQL -> MongoDB
- Prisma
 - Object Relational Mapper
 - Works for Sql and NoSql databases
 - Used to create the schemas and generates the typescript object type
 - Easily fetch or create entries within the database
- Axios
 - Service for the frontend to reach the backend
 - i. Creates JSON and request to the server based on the GET, POST, PUT, and DELETE requests.
 - ii. Handles the response from the service and returns the appropriate data

Backend Status + Remaining Work

- RESTful API Endpoints added for a Map Controller and Layer Controller.
- We had a local MySQL DB setup to test while we got a long term solution added. We are able to successfully use Postman to add data.
- We have a MongoDB database setup, and we access it through prisma, but don't have it connected to the frontend yet.
- Current map and layer data needs to be migrated to MongoDB

The background of the slide is a photograph of the Iowa State University campus, featuring several large, classical-style buildings and a row of trees in the foreground. The entire image is overlaid with a semi-transparent red filter. The word "Demo" is centered in the middle of the slide in a white, sans-serif font.

Demo

IOWA STATE UNIVERSITY