

Capstone

Visualis

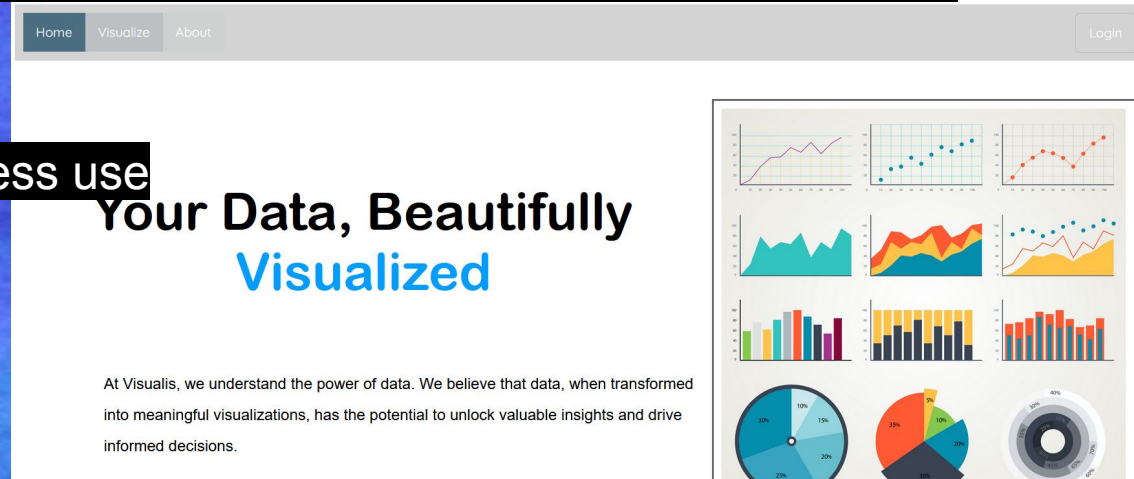
Fonz Hamilton

What does the app do?

- A Data visualization website
- Users need account to use
- Users can upload their own data
- Users can choose how they want to visualize their chart

Business use

- Can be a paid model where users pay for access to make professional level charts
- Could have a free and pro version with different features for each version
- Can create charts for business use



How does it work?

- The app utilizes Java Spring Boot, Javascript, Html/CSS, and MySQL
- Users are required to register to have access to the visualize page
- Once registered, Users are added to a database
- They are then able to upload a file which is stored server-side with a database connection between the user and the file

How Does it work? -Cont-

- Data entity stores name, User, and id of DataInfo entity
- DataInfo entity stores name, user description, file type, and where the data is located server side
- The user has access to the files they have stored and can choose one to view
- A table is created with the values of the data file
- The user then has the option to choose which data point will be the X-axis and which Data point will be the Y-axis

How it works -Cont-

- The user can then choose a chart to visualize the data
- In the back end, Spring boot directs and controls the security, file processing, and database queries.

- Javascript handles the visualizations using d3.js

```
UserPrincipal userPrincipal = (UserPrincipal) authentication.getPrincipal();
User user = userPrincipal.getUser();
user = userRepository.findById(user.getId()).orElse(null);
try {
    // save the file using FileUtil
    String filePath = fileUtil.saveFile(file);
    String fileName = StringUtils.cleanPath(file.getOriginalFilename());
    fileName = fileDuplicateChecker.checker(fileName);

    String realFileName = fileUtil.getNewFileName(); // gets the weird UID generated n

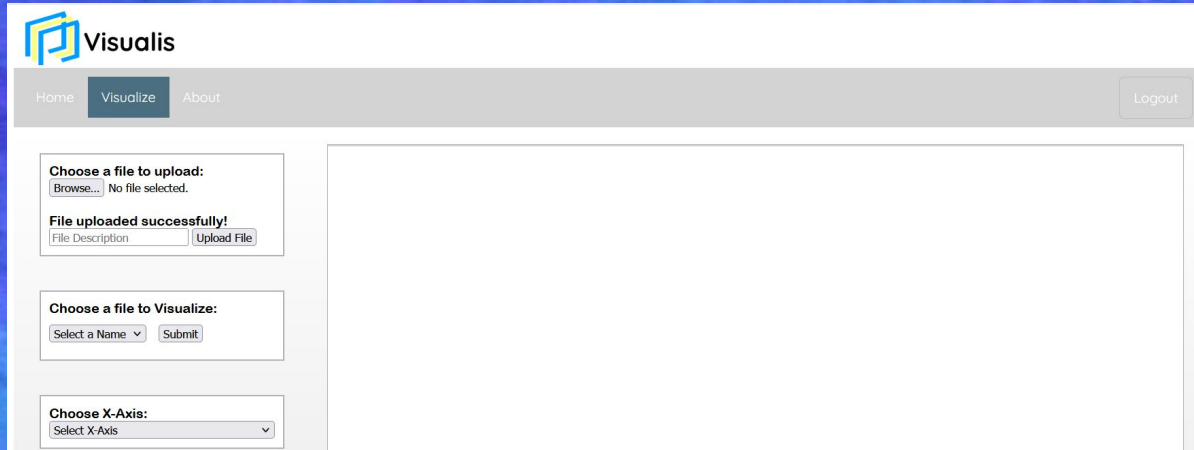
    // create DataInfoDTO with file information
    DataInfoDTO dataInfoDTO = new DataInfoDTO();
    dataInfoDTO.setSourceLink(filePath + "\\" + realFileName);
    log.debug("SourcePath for file {}", dataInfoDTO.getSourceLink());
    dataInfoDTO.setName(fileName);
    dataInfoDTO.setDataType(fileUtil.getFileType(file));
```


What was learned?

- This project taught me a lot!
- I learned a lot about what goes into security
- I learned a lot of javascript to make the visualizations
- I learned a lot about file processing
- I learned how to make Java and Javascript share data between each other
- I learned how to make dynamic web pages with thymeleaf

Hardest part?

- Dealing with Spring security!
- Learning D3.js
- Keeping track of the code



The screenshot shows the Visualis web application interface. At the top, there is a navigation bar with the Visualis logo (a stylized 'V' made of blue and yellow squares) and the text 'Visualis'. Below the logo, there are three navigation links: 'Home', 'Visualize' (which is highlighted with a dark blue background), and 'About'. On the right side of the navigation bar, there is a 'Logout' button. The main content area is divided into two columns. The left column contains three sections: 1. 'Choose a file to upload:' with a 'Browse...' button and the text 'No file selected.' 2. 'File uploaded successfully!' with a 'File Description' input field and an 'Upload File' button. 3. 'Choose a file to Visualize:' with a 'Select a Name' dropdown menu and a 'Submit' button. The right column is currently empty.

Additional Features?

- So many!
- I plan to continue working on this for a while
- I would like to add more chart types
- I would like to animate the charts
 - Make sliders and have the data more customizable
 - I had an idea I wasn't able to implement that I really want to continue on
- I want to display the user descriptions the users wrote for their data
- I want to add a user dashboard and an admin dashboard