

ModelKB DB - Project Increment 1

Team Members

Brett Recker - Class ID: 22

Redwan Al Badawi - Class ID: 1

Increment Goals and Objectives

For this increment, our main objectives were to research similar systems, get an idea for the database schema, and investigate technologies that can be used to implement our project.

Research

We reviewed two papers regarding similar systems:

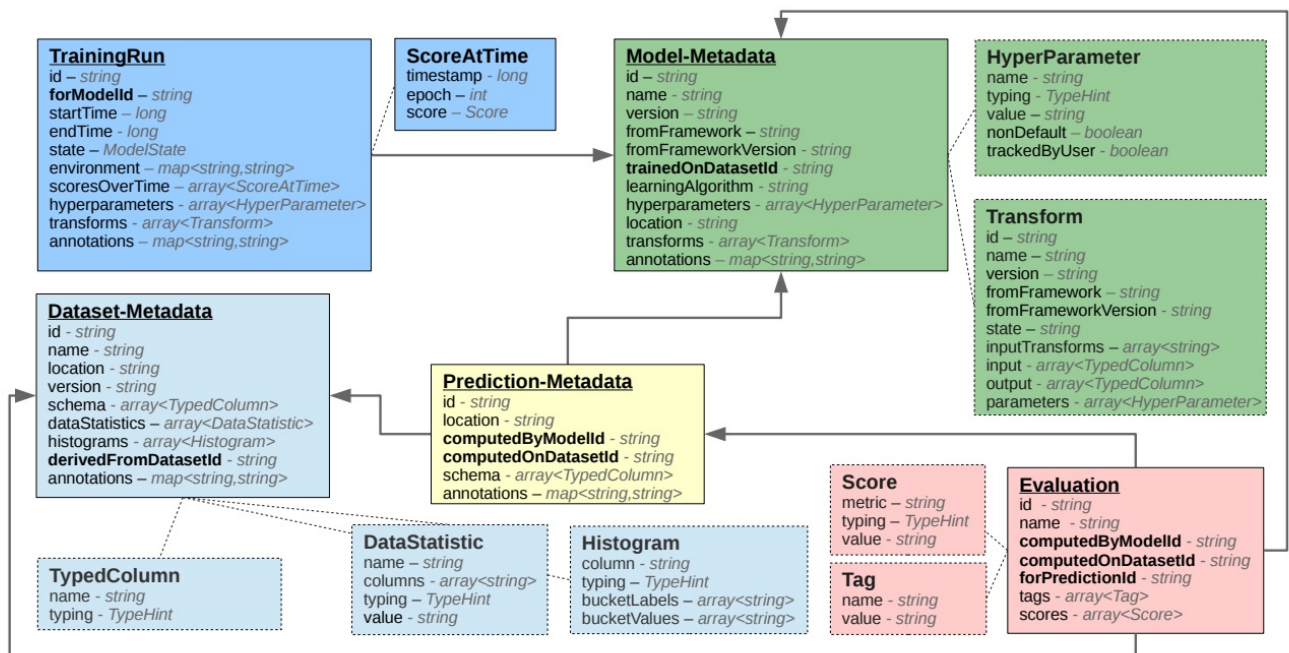
ModelHub - <https://arxiv.org/pdf/1611.06224.pdf>

Amazon's ML Model -

http://learningsys.org/nips17/assets/papers/paper_13.pdf

These papers provided insight on the architecture and functionality of a deep learning database. Amazon's paper provided a great guide for a

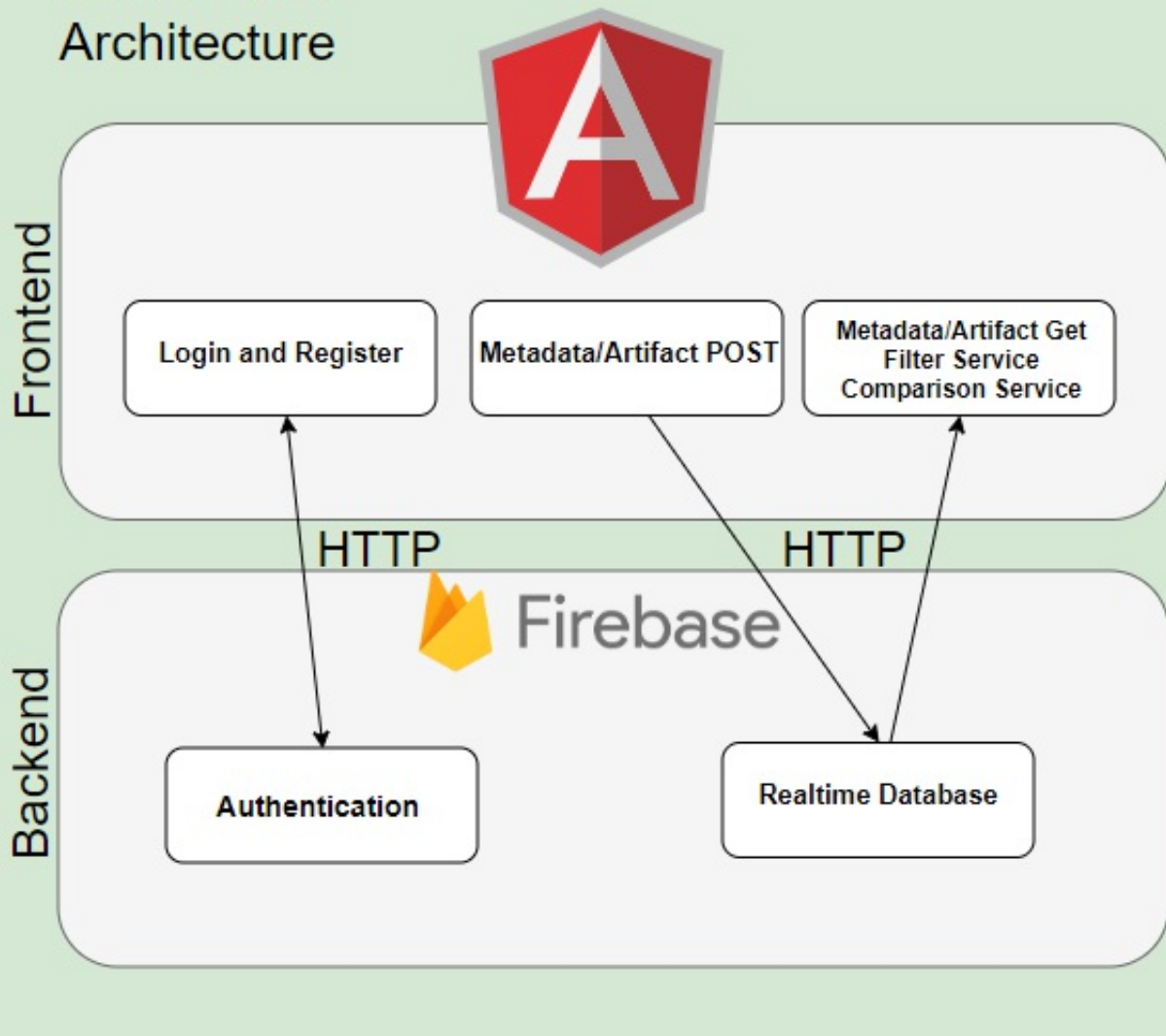
database schema that can be used to store these deep learning artifacts:



Throughout that process, we created a preliminary database model diagram, as well as UI mockups of the login and register pages.

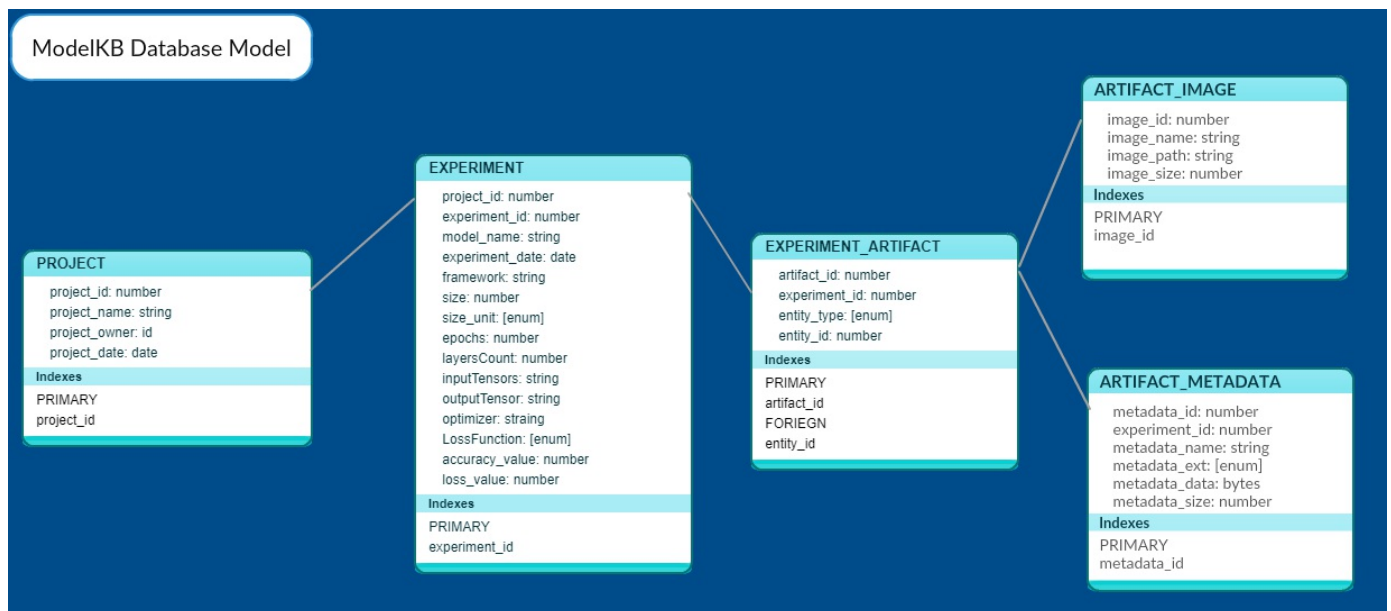
Basic Architecture

ModelKB DB Architecture



Database Model

Using Amazon's ML model as guidance, we created the following database schema diagram for our project:

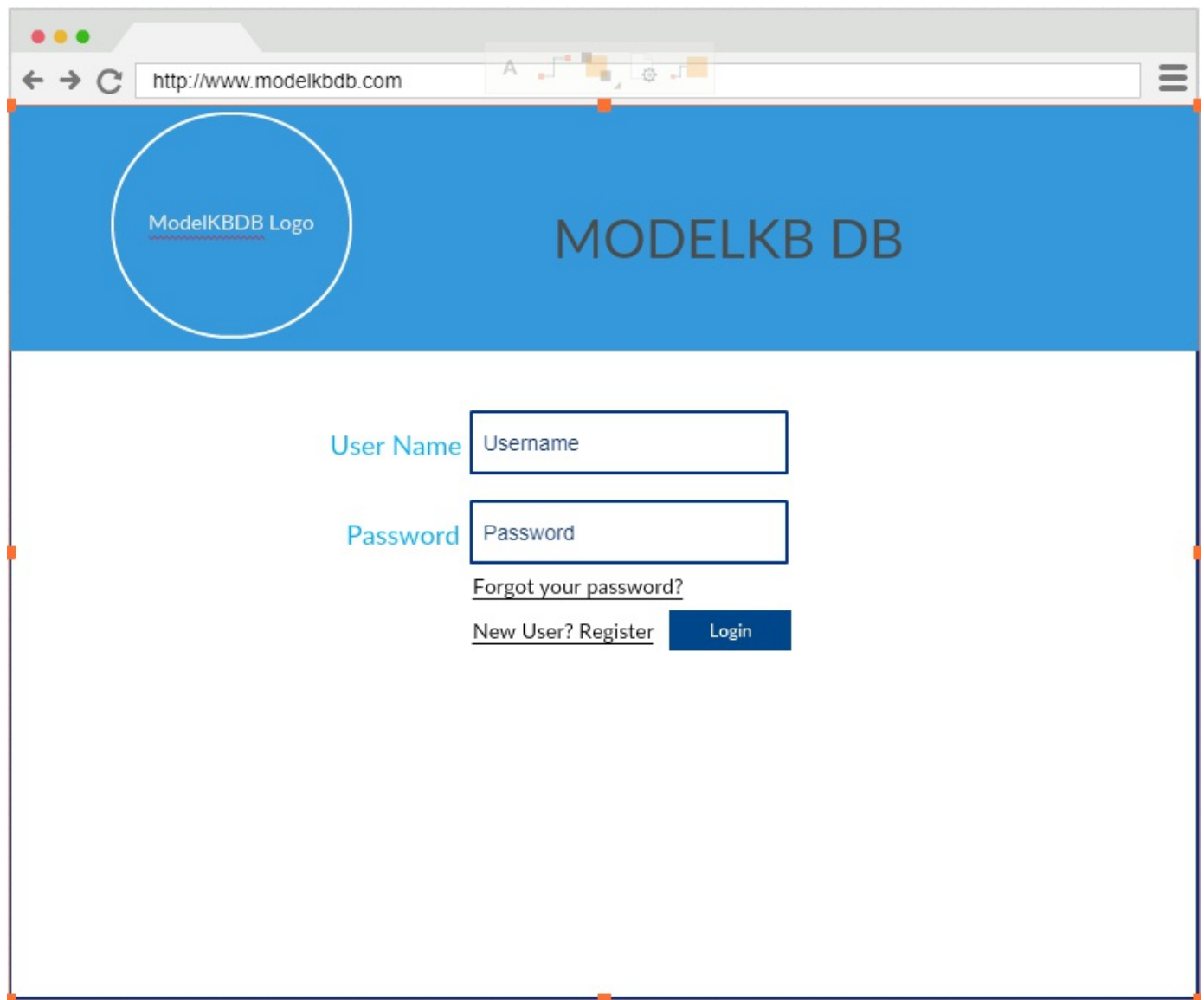


The main table in our database is **PROJECT**. This table will hold the project details (name, owner, date) as well as a unique id. Each project will have 1 to many experiments, making the **EXPERIMENT** table the direct descendant. This table will contain all the information parsed from the metadata flat file. Next, the **EXPERIMENT_ARTIFACT** table will store the additional artifact's type and id which can then be linked to **ARTIFACT_IMAGE** and **ARTIFACT_METADATA**. These tables will store pertinent data to each artifact type.

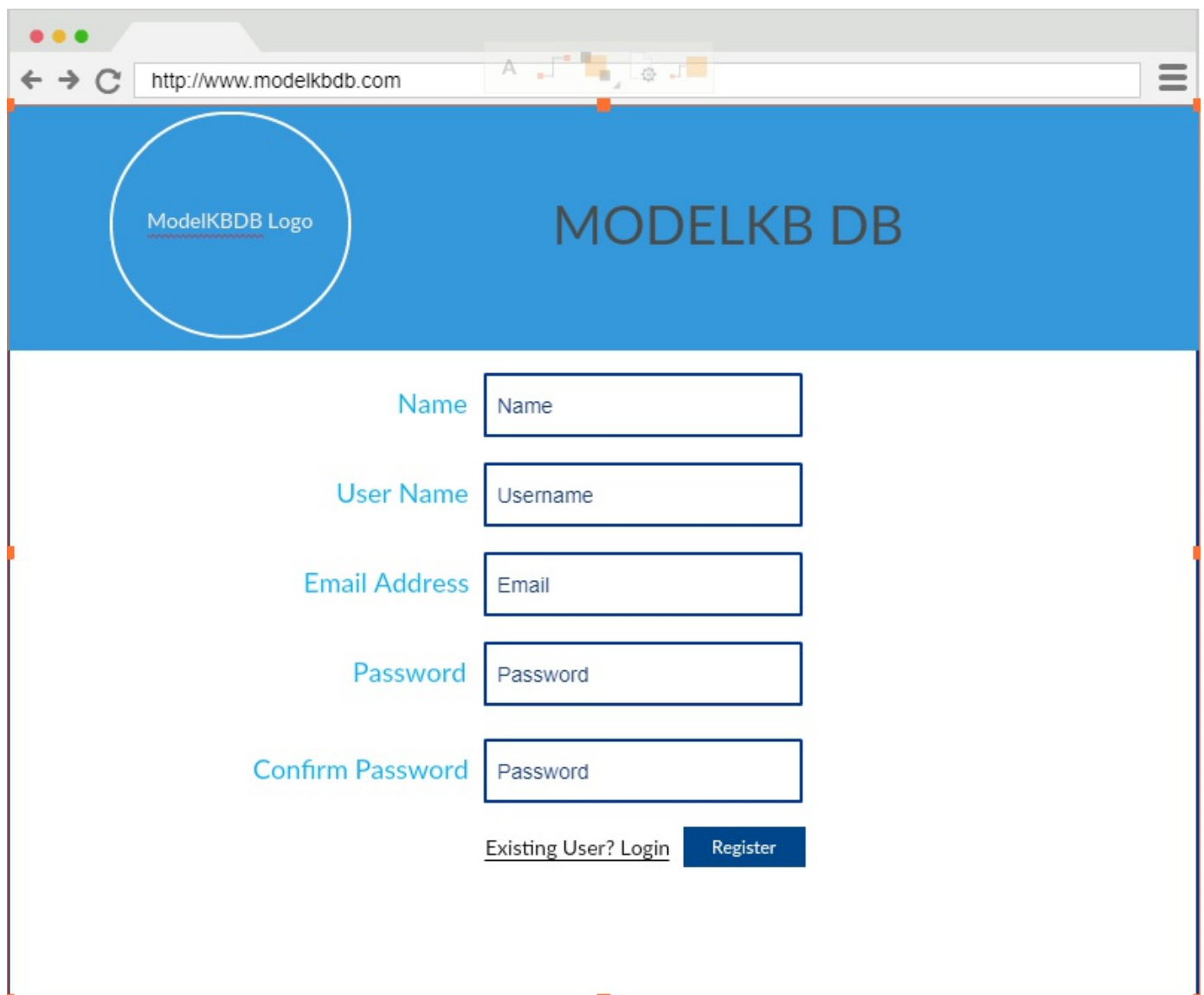
This schema is a work in progress, and will be reviewed with our project manager to determine what changes should be applied.

Login/Register UI Mockup

Below is our Login page UI mockup (more discussion regarding this page later):



Below is our Login page UI mockup (more discussion regarding this page later):



A screenshot of a web browser displaying the registration page for ModelKBDB. The browser's address bar shows the URL <http://www.modelkbdb.com>. The page has a blue header with the ModelKBDB Logo on the left and the text "MODELKB DB" on the right. Below the header, the registration form is centered on a white background. It consists of five labeled input fields: "Name", "User Name", "Email Address", "Password", and "Confirm Password". Each field has a corresponding label in blue text to its left. At the bottom of the form, there is a link "Existing User? Login" and a dark blue "Register" button.

ModelKBDB Logo

MODELKB DB

Name

User Name

Email Address

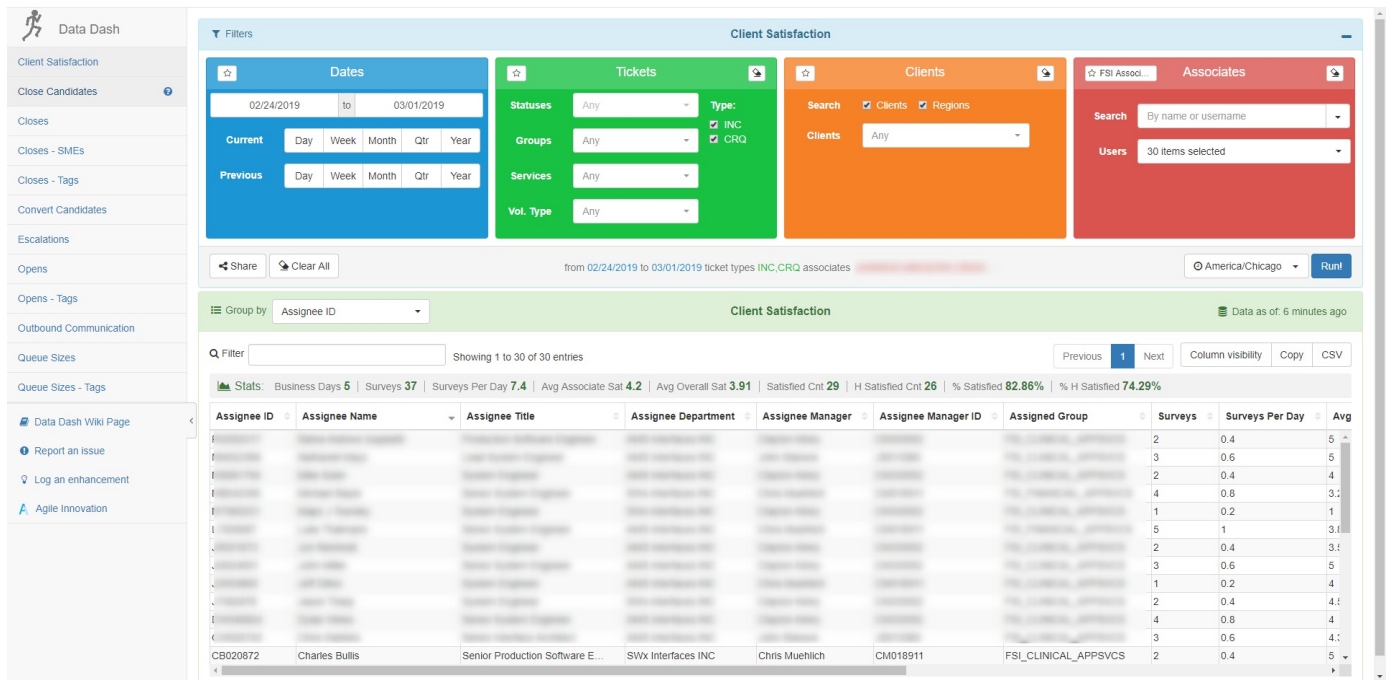
Password

Confirm Password

[Existing User? Login](#) [Register](#)

Home Page Details

Our home page mockup has not been completed yet, but will feature predefined filtering and comparison options, as well graphical output to visualize the deep learning data. In Brett's current role at Cerner, he uses a database web app called Data Dash which our home page will be based on:



Database Details

Our project will utilize Firebase’s Authentication service for login and register actives. Users will be registered with their email address and password.

The ModelKB data will be stored using Firebase’s Realtime Database. This is a NoSQL cloud database which uses key-value pairs. Further investigation will be necessary to ensure our database schema integrates with Firebase appropriately.

Firebase File Metadata storage:

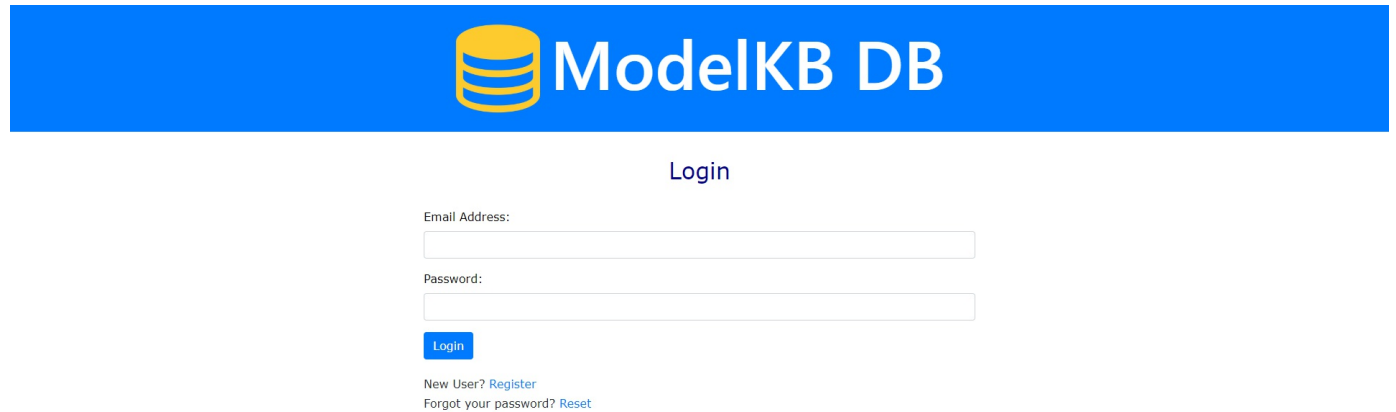
<https://firebase.google.com/docs/storage/web/file-metadata>

Firebase Image Storage:

<https://firebase.google.com/docs/storage/web/create-reference>

Current Implementation

Header/Navigation and Login Components:



ModelKB DB

Login

Email Address:

Password:

Login

New User? [Register](#)

Forgot your password? [Reset](#)

Currently, the header only features a logo and title. Moving forward, this may include app level menus/features so it was built as a separate component. The login page is rather simple, featuring two inputs for user's email and password. Upon successful login the app will route to the home page. This page also has a link for new user registration.

Register Component:

New User Registration

Email Address:

Password:

Confirm Password:

Existing User? [Login](#)

Rather simple as well, this component will validate user inputs upon registration. It will ensure the passwords entered match, and are strong enough. Also, validation will be performed on the email.

Future Direction

Moving forward, the main home component will be stood up with functionality to store ModelKB data. An example of this data can be found using the link below:

[https://github.com/RedwanAlbadawi/CS5551Project/tree/master/ModelKB - Artifacts/test cases](https://github.com/RedwanAlbadawi/CS5551Project/tree/master/ModelKB-Artifacts/test%20cases)

This will require integrating Firebase within our app.

The login and registration authentication service will be fully implemented ensuring to track user sessions appropriately.