# Introduction

## System introduction

This system is a typical three-tier web application mainly designed and implemented using the MEAN stack.

The data source is a data set that contains the revision history of featured Wikipedia articles from the English Wikipedia page[1], which contains many JSON format files. This system stores the data set in a MongoDB non-relational database. By querying and summarizing the information in the data set on the server-side, the summarized summary and statistical information results are returned to the front-end pages, and the concise and clear chart information is generated and displayed on the pages.

This webpage program can also interact with third-party websites through published web APIs. By invoking publicly available API endpoints, the application is able to fetch the latest dataset directly from Wikipedia as well as displaying the hottest news topics related to the selected article.

This web application has been deployed on the Heroku Cloud Application Platform, which can be opened directly via <https://com5347angulargourpsix.herokuapp.com/>.

## Division of labor

Front-end development: Shanzheng Liu

Back-end development: David Dai & Qizhen Zhu

# Feature design

## Function module introduction

### User module

In this system, the user's login and password modification functions are provided. After providing the email and password, the user can successfully register an account. The user must log in to perform a series of operations on the website. The system also provides the option to reset the user's password based on the authentication answer. After completing the query access operation on the page, the user can safely log out of the account.

A screenshot of a social media post

Description automatically generated

Figure1.1 Home page

### Overall analytics module

In the article's overall analysis section, we have compiled the given data set to analyze it in an overall manner and extract constructive information, such as "the first two articles with the most revisions and the number of revisions." And the information is clearly displayed in the form of text.

In addition, by reading administrators.txt and bots.txt, we analyzed the types of users who modified the article, displayed four different types of user information in the form of pie charts, and added some web page interaction functions.

This module also provides a bar chart to show the relationship between the revision year and the user type, and can also switch between the line chart and the bar chart.

A screenshot of a cell phone

Description automatically generated

Figure1.2 Overall analytics page

### Individual article analytics module

In this module, we conduct specific analysis of individual articles based on the articles selected by the user. We first check whether the history of the article is up-to-date, if not, update it through the MediaWiki API and pop up a message.

In addition, we display and analyze some basic information for selected articles, such as headlines and related news from third parties.

We also provide three charts. When the user selects a specific range through the year filter, the chart can analyze the relationship between the year, user type, and user in this article according to the range selected by the user.

A screenshot of a cell phone

Description automatically generated

Figure1.3 Individual article analytics page

### Author analytics module

In this module, users are able to search the author they interested in. After they input some letters into the input box, the autocomplete component will show the names relating to the keywords, and users can select the name directly from the name list. After pressing the search button, the article edited by the author you searched and the revision number will be shown in the article list. In the next step, users can select the article shown in the article list and click the button “select”. The timestamp of the revision record relating to the author and article will be shown in the timestamp list.

A screenshot of a social media post

Description automatically generated

Figure1.4 Author analytics page

## Database design

### Database structure



﻿ Figure2.1 ﻿Collection revinfos structure



﻿ Figure2.2 ﻿Collection ﻿users structure

### Database installation instructions

**Data import script install**

For data importing, a python library script has been deployed in PiPy. Before using this script, python >=3.6 should be installed on your computer.

1: Open the command in Windows or terminal in  Mac OS and input the command

pip install MongoFileImport

2: Find out your MongoDB URL and transfer it into python format. It looks like mongodb+srv://username:password@cluster0-xxxxx.mongodb.net

3:Find out your name of the database and the name of an **existing** collection.

4:Get the absolute path of your data file which only contains JSON files.

5:Open your command or terminal again, and input the command like:

MongoFileImport --mongourl mongodb+srv://username:password@cluster0-xxxxx.mongodb.net --db test --dir d:/Downloads/Dataset\_22\_March\_2020/Dataset\_22\_March\_2020/revisions/ --cn collectionname

for this command, “--mongourl” refers to your MongoDB URL,  “--db” refers to the name of the database. “--cn” refers to the name of the collection and the “--dir” means your path of the folder which you want to import.

If you have any questions, you can visit <https://github.com/shanzhengliu/MonogdbFileImport> to get more details and star it.

**Frontend project install**

1: use Command or Terminal to get into the root path of the frontend project “web-angular”.

2: run command “npm install” to install the dependencies and library for the project.

3: run command “npm start”, after compiling and rendering, the project will start.

A close up of a green background

Description automatically generated

Figure2.3 Frontend project install

**Backend project install**

1: use Command or Terminal to get into the root path of the backend project “group-back”.

2: run command “npm install” to install the dependencies and library for the project.

3:run command “npm run start:server”, and the express server will start.

A screenshot of a cell phone

Description automatically generated

Figure2.4 Backend project install