Assignment 2 Technical Report

Contents

[Architecture and technology 1](#_Toc41162875)

[MVC Pattern 1](#_Toc41162876)

[Introduction 1](#_Toc41162877)

[Model 1](#_Toc41162878)

[View 2](#_Toc41162879)

[Controller 2](#_Toc41162880)

[External JS/CSS Files 2](#_Toc41162881)

[MongoDB 2](#_Toc41162882)

[PUG 2](#_Toc41162883)

[NodeJS 2](#_Toc41162884)

[Functional Requirements 2](#_Toc41162885)

[Main/Landing Page 2](#_Toc41162886)

[Aesthetics 2](#_Toc41162887)

[Article Analytics Functionality 2](#_Toc41162888)

[Overall Analytics 2](#_Toc41162889)

[Individual Analytics 2](#_Toc41162890)

[Author Analytics 5](#_Toc41162891)

[Non-Functional Requirements 5](#_Toc41162892)

[Performance 5](#_Toc41162893)

[Security 5](#_Toc41162894)

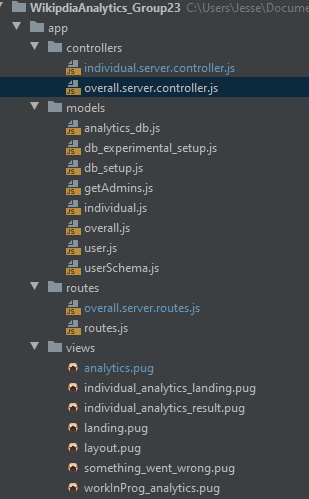
[Usability 5](#_Toc41162895)

# Architecture and technology

## MVC Pattern

### Introduction

We follow MVC design patterns for our website, this is to ensure modularity, so that all members can integrate different parts of the website without too much overlap.



### Model

The model area of the website contains all the database queries, as well as some classes that sets up our database. All of these methods are exported for the controller to use.

### View

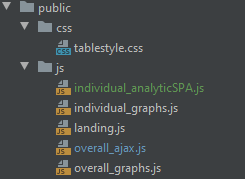
For our view technology, we use PUG as our preferred way to render the website, some of these pug files receive data server side which is render blocking in some cases, however some parts of the website is able to render dynamically via js using ajax and jquery.

### Controller

The controller handles most of the complicated methods where it retrieves data from the model and either returns as a json for external js files to retrieve dynamically or to send directly to pug files to render.

## External JS/CSS Files

We use JS/CSS files to better aesthetics and usability of the website where JS files are mostly used to dynamically render pages and css files control the standard design.



## MongoDB

MongoDB is used as the core back end database for this assignment. Members had to set up their own MongoDB servers locally as per week8 tutorial and insert revisions in various ways. Some members followed the tutorial to insert revisions while others used nodejs scripts.

## PUG

PUG is the template viewer of choice for the project as members think this engine is intuitive.

## NodeJS

NodeJS is the back end language used to write most of the files in MVC folders.

# Functional Requirements

## Main/Landing Page

Lakshesh

## Article Analytics Functionality

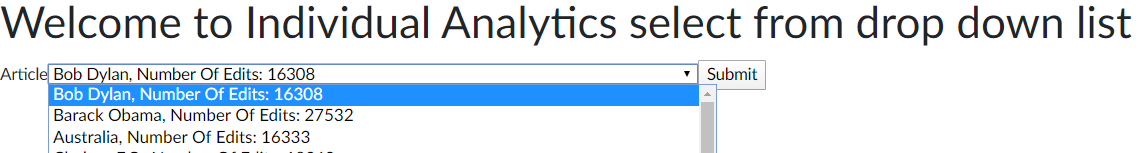
### Overall Analytics

Carl

### Individual Analytics

For this sections of analytics, most functional requirements are implemented with the exception of the ‘year filter’ for graph selection. The main problems that will be discussed will mainly be involved with dynamic rendering.

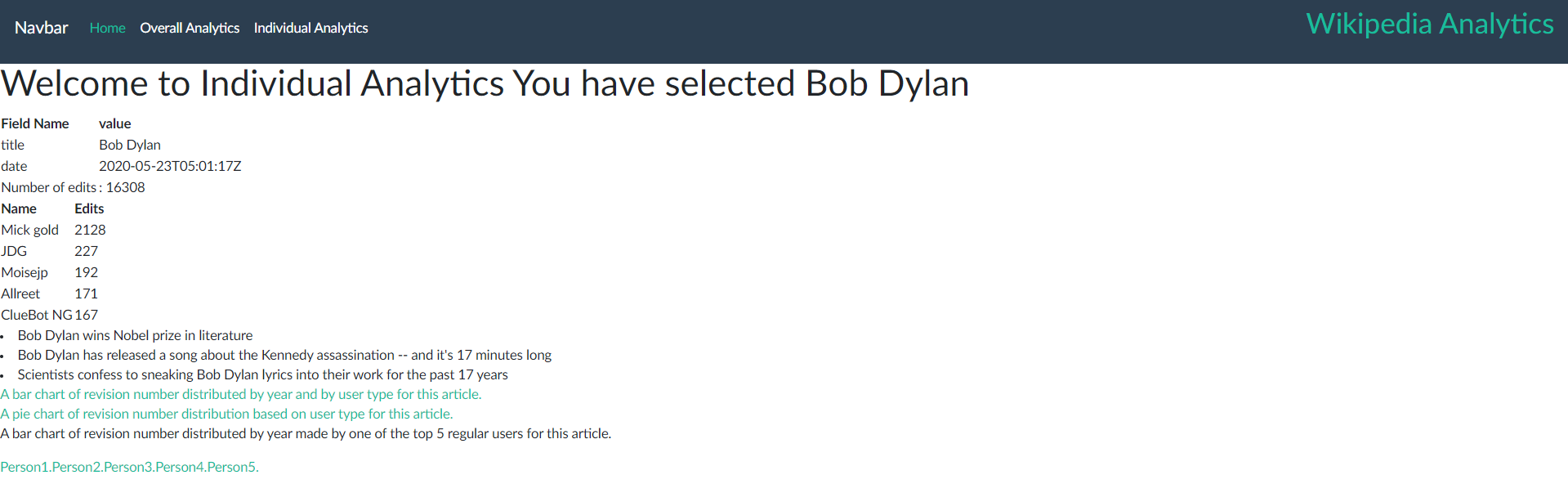
First thing that you’ll see when routed to /Individual, is a form which allows you to select an article, to help you choose a selection, the number of revisions or ‘edits’ is shown alongside the article in question. Once an article is selected and the user presses the submit button, it sends the article name to /Individual/results to finally show the summary for Individual Analytics



In terms of the technical implementation, when routed to /Individual, the controller method retrieves an array of all the article names within ./Dataset\_22\_March\_2020/revisions. The controller then calls a model method to retrieve the sum of revisions grouped by the article name. the results are put into a list then sent to pug to render.

The problem with this implementation is that its not dynamic, the controller calls the model method however should have rendered the page without passing any variables and simply allowed an external js to retrieve the data and then render dynamically.

The second page features most of the functional requirements for Individual analytics and is parsed in such a way that long queries are dynamically rendered allowing for a better user experience.



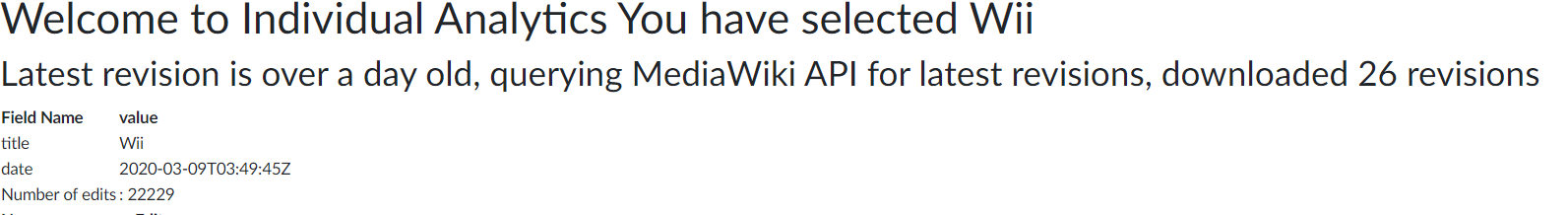
The design of the analytics page extends another pug template which imports the css and navigation bar for the page.

The core features are all represented aside from ‘year selection’ for the graphs. We have the title of the article as well as total number of revisions for the article. Top 5 users based on their revisions numbers ranked according to the number of revisions. Top 3 all time reddit post titles from /r/news as a list format. As well as the 3 graphs, with the last graph being able to choose between the 5 users as person1-5.

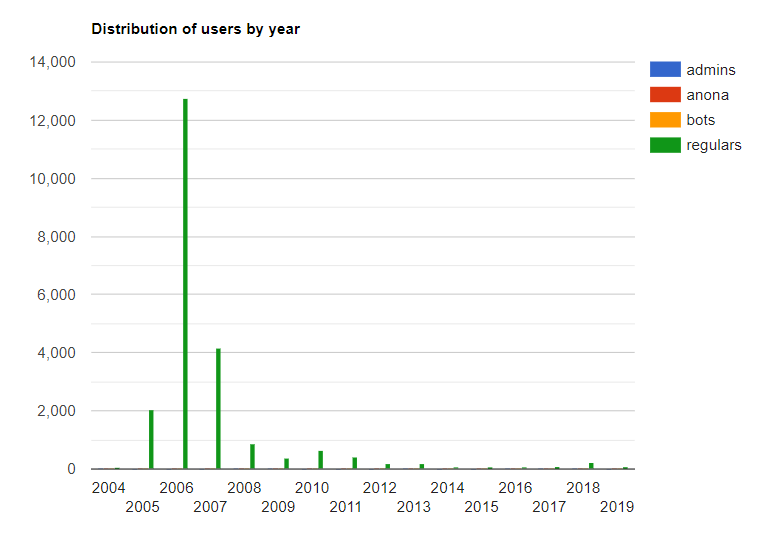
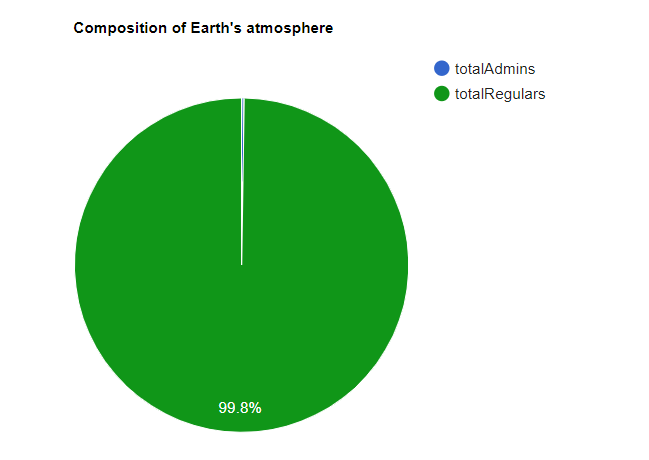
One thing to note is another core functionality that’s may not be visible and that’s querying for MediaWiki and showing the total number of downloaded revisions to be inserted into the database. That is because it checks the date of the latest revision if its over a day old. In this case for article Bob Dylan, we can see that the date is 23/05/2020 the same date at the time of writing this paragraph. The latest revision is not over a day old, so we never query MediaWiki API

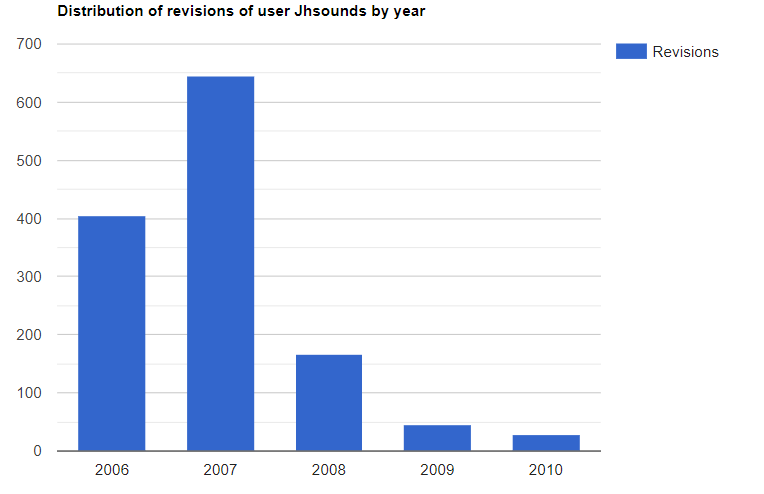


When we select an article that is over a day however, the following blurb ‘Latest revision is over a day old, querying MediaWIKI API for latest revisions, downloaded x revisions’ will show, the revisions are downloaded and inserted into the database for use.



Taking a look at the graphs, we used google charting for our representation of the graphs and aside from implementing selection of years, this requirement is mostly present





### Author Analytics

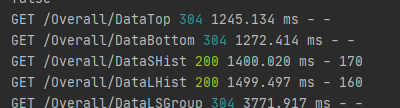
Carl

# Non-Functional Requirements

## Performance

Overall, the website performs at a speed to where its usable but the user will recognize some areas where performance is noticeably slow, this is mainly in querying data to build the graphs. The slowest areas of our website however is rendered dynamically so the user is still able to have a usable experience when accessing the website.





## Security

LakShesh

## Usability

In terms of usability, does not allow free access until the user signs up or logs into the website. Any attempt to access any part of the site other than the landing page will redirect to the landing page.

The website features a navigation bar which routes home where Overall analytics and author analytics are displayed on one page and Individual analytics on a separate routing..