FINAL PROJECT REPORT

Week 6

Abstract

This Is the last report of my project in this report I tie everything together.

Mohammad Hossein Movahedi

Movahedi.m@northeastern.edu

Table of Contents

Introduction

My project is about an academic database between Wikipedia and Google scholar. Scholars write their articles in forms and use the built-in function to add their citation materials, not put them in the text. So, behind the website, there would be a vast dynamic database of articles that can be easily updated if the source of the paper changes.

I believe in this app, and I think it should be free to use so It would run like Wikipedia with the support of its users.

This idea first came to me in an article many years ago about how some mathematics articles may be incorrect because they use other mathematicians' proofs as part of their own instead of proofing some features independently, and how if the first scholar made a mistake in their proof, the conclusion of all proofs based on the first proof would become faulty immediately.

Business Analysis

My app would be a scholarly website with articles that never become outdated or irrelevant. Additionally, producing, updating, and citing would be quicker, faster, more efficient, and more accurate. Here are five rules I have thought about them so far I would be more than happy to add to them.

1. inspectors should be always be referenced by a current inspector or admin

2. comments should always refer to something, and if the thing that they are referring to is deleted following comments will automatically be deleted

3. scholar can become an author or an inspector, but in that case, they have to provide an academic email address and be referred to by a current inspector

4. inspectors can't rate an article that they wrote

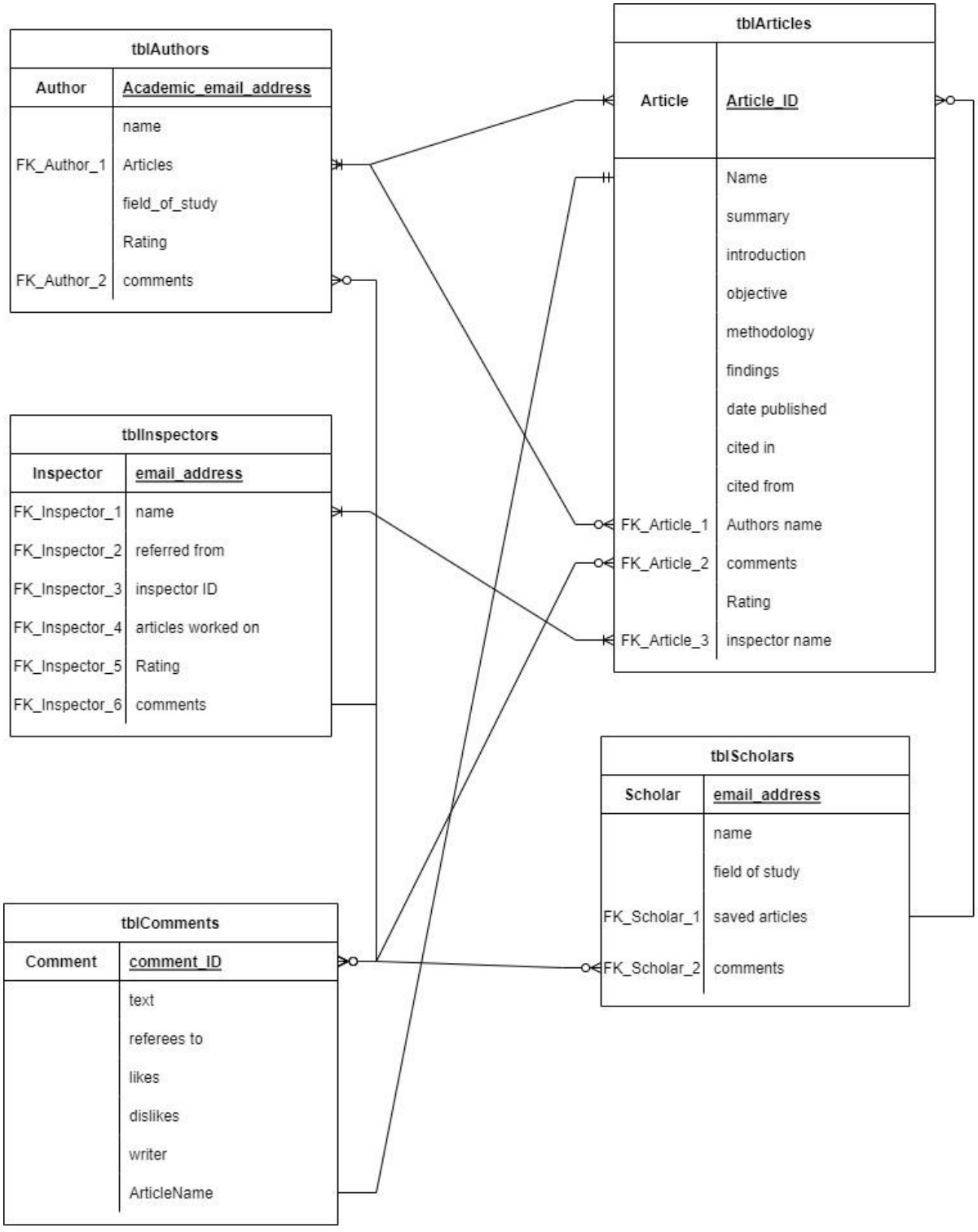
Also, I thought of the comments like the way Reddit uses them; however, as you mentioned, it's nice for an academic site to have objective and informative words. So far, I have come up with three ideas, but they have potential problems.

1. I can give authors access to the comment section of their articles to delete or promote comments, but in that case, I would sacrifice the freedom of speech on my site.
2. I can implant some reward system for comments, but people might start spamming again to get the rewards.
3. I can sort comments not by date but by the rating of the authors, which I think is the best option here, but this won't address the lack of comments on unpopular articles.

Table Design and Analysis

ER Diagram

My Project ER Diagram showed below.



Creating Tables

```
CREATE TABLE "tblArticles" (

        "Article" INTEGER NOT NULL UNIQUE,

        "Name"    TEXT NOT NULL,

        "Summary "          TEXT NOT NULL,

        "Introduction"      TEXT NOT NULL,

        "Objective"         TEXT NOT NULL,

        "Methodology "      TEXT NOT NULL,

        "Findings"          TEXT NOT NULL,

        "DatePublished"     TEXT NOT NULL,

        "Citedin" TEXT NOT NULL,

        "CitedFrom"         TEXT NOT NULL,

        "AuthorsEmail"      TEXT NOT NULL,

        "Comments"          TEXT,

        "Rating"  NUMERIC NOT NULL,

        "InspectorID"       NUMERIC,

        CONSTRAINT "FK_tblArticles_tblComments" FOREIGN KEY("Comments") REFERENCES
"tblComments"("CommentID"),

        CONSTRAINT "FK_tblArticles_tblInspectors" FOREIGN KEY("InspectorID") REFERENCES
"tblInspectors"("InspectorID"),

        CONSTRAINT "FK_tblArticles_tblAuthors" FOREIGN KEY("AuthorsEmail") REFERENCES
"tblAuthors"("AcademicEmailAddress"),

        CONSTRAINT "PK_tblArticles" PRIMARY KEY("Article" AUTOINCREMENT)

);
```

```
CREATE TABLE "tblAuthors" (

        "AcademicEmailAddress"       TEXT NOT NULL UNIQUE,

        "Name"    TEXT NOT NULL,

        "Articles"          TEXT NOT NULL,
```

```
          "FieldOfStudy"      TEXT,

          "Rating"  NUMERIC,

          "Comments"          TEXT,

          CONSTRAINT "FK_tblAuthors_tblComments" FOREIGN KEY("Comments") REFERENCES
"tblComments"("CommentID"),

          CONSTRAINT "FK_tblAuthors_tblArticles" FOREIGN KEY("Articles") REFERENCES
"tblArticles"("Article"),

          CONSTRAINT "PK_Authors" PRIMARY KEY("AcademicEmailAddress")

);
```

```
 CREATE TABLE "tblComments" (

          "CommentID"         INTEGER NOT NULL UNIQUE,

          "Text"    TEXT NOT NULL,

          "likes"   INTEGER,

          "Dislikes"          INTEGER,

          "Writer"  TEXT,

          "ArticleID"         TEXT,

          CONSTRAINT "FK_tblComments_tblArticles" FOREIGN KEY("ArticleID") REFERENCES
"tblArticles"("Article"),

          CONSTRAINT "PK_tblComments" PRIMARY KEY("CommentID" AUTOINCREMENT)

);
```

```
 CREATE TABLE "tblInspectors" (

          "EmailAddress"      TEXT NOT NULL,

          "Name"    TEXT NOT NULL,

          "ReferredFrom"      TEXT NOT NULL,

          "InspectorID"       INTEGER NOT NULL UNIQUE,

          "ArticlesWorkedOn"  TEXT NOT NULL,

          "Rating"  NUMERIC,

          "Comments"          TEXT,

          CONSTRAINT "FK_tblInspectors_tblArticles" FOREIGN KEY("ArticlesWorkedOn") REFERENCES
"tblArticles"("Article"),

          CONSTRAINT "FK_tblInspectors_tblComments" FOREIGN KEY("Comments") REFERENCES
"tblComments"("CommentID"),
```

```
        CONSTRAINT "PK_tblInspectors" PRIMARY KEY("InspectorID" AUTOINCREMENT)

);
```

```
 CREATE TABLE "tblScholars" (

        "EmailAddress"     TEXT NOT NULL,

        "Name"    TEXT NOT NULL,

        "FieldOfStudy"     TEXT,

        "SavedArticles"    TEXT,

        "Comments"         TEXT,

        CONSTRAINT "FK_tblScholars_tblArticles" FOREIGN KEY("SavedArticles") REFERENCES
"tblArticles"("Article"),

        CONSTRAINT "FK_tblScholars_tblComments" FOREIGN KEY("Comments") REFERENCES
"tblComments"("CommentID"),

        CONSTRAINT "PK_tblScholars" PRIMARY KEY("EmailAddress")

);
```

Database Implementation

Then I import some sample data.

```
 PRAGMA foreign_keys = 0;

INSERT INTO tblComments (Text,likes,Dislikes,Writer)

VALUES ('The quick brown fox jumps over the lazy dog',10,15, 'Mark Dunn');

INSERT INTO tblComments (Text,likes,Dislikes,Writer,ArticleID)

VALUES ('so good ',100,11, 'Caesar Rodney','Declaration of Independence');

INSERT INTO tblScholars (EmailAddress,Name,FieldOfStudy,SavedArticles,Comments)

VALUES ('a@gmail.com','Mark','IT', 'Declaration of Independence',4);

INSERT INTO tblInspectors
(EmailAddress,Name,ReferredFrom,InspectorID,ArticlesWorkedOn,Rating,Comments)

VALUES ('b@gmail.com','Rex','Randy',1,'Declaration of Independence',5,'The unanimous Declaration
of the thirteen united States of America');

INSERT INTO tblAuthors (AcademicEmailAddress,Name,Articles,FieldOfStudy,Rating)

VALUES ('c@gmail.com','alexander hamilton','Declaration of Independenc','theater',5);

INSERT INTO tblArticles ("Name","Summary ","Introduction","Objective","Methodology
","Findings","DatePublished","Citedin","CitedFrom","AuthorsName","Comments","Rating","InspectorI
D")
```

```
VALUES ('Declaration of Independence,’' the unanimous Declaration of the thirteen united States
of America','We hold these truths to be self-evident','all men are created equal',' whenever any
Form of Government becomes destructive of these ends, it is the Right of the People to alter or
to abolish it','let Facts be submitted to a candid world','2020/2/2','the Rotunda at the
National Archives Museum','Wikipedia','c@gmail.com',3,5,1);

PRAGMA foreign_keys = 1;
```

The first select function selects the name of the author by the articles

```
 SELECT tblAuthors.Name ,AcademicEmailAddress



FROM tblAuthors



JOIN tblArticles

on tblArticles.AuthorsEmail = tblAuthors.AcademicEmailAddress



WHERE tblArticles.Name = 'Declaration of Independence'
```

The second select function selects the Scholars by their likes

```
 SELECT tblScholars.Name , tblScholars.EmailAddress



FROM tblScholars



JOIN tblArticles

on tblArticles.Name = tblScholars.SavedArticles



WHERE tblArticles.Name = 'Declaration of Independence'
```

The third select function selects the Inspectors by their Article

```
 SELECT tblInspectors.Name,tblInspectors.EmailAddress



FROM tblInspectors

JOIN tblArticles

ON tblArticles.InspectorID = tblInspectors.InspectorID
```

```
WHERE tblArticles.Name = 'Declaration of Independence'
```

The fourth select function selects the Article by their Comments.

```
 SELECT tblAuthors.Name , tblAuthors.AcademicEmailAddress



FROM tblAuthors



JOIN tblArticles

on tblArticles.AuthorsEmail = tblAuthors.AcademicEmailAddress



JOIN tblComments

on tblComments.ArticleID = tblArticles.Name



WHERE tblComments.ArticleID = 'Declaration of Independence'
```

The fifth select function selects the Article by their Comments

```
 SELECT tblArticles.Name , tblArticles.Rating



FROM tblArticles

JOIN tblInspectors

ON tblArticles.InspectorID = tblInspectors.InspectorID



WHERE tblInspectors.Rating >= 3
```

Analytics, Reports, and Metrics

There are lots of analytics that can be run on this project one of them that I think would be the most beneficial is the rating of the articles if the rating of an essay is low, it can be said it is not a good article should be revised.

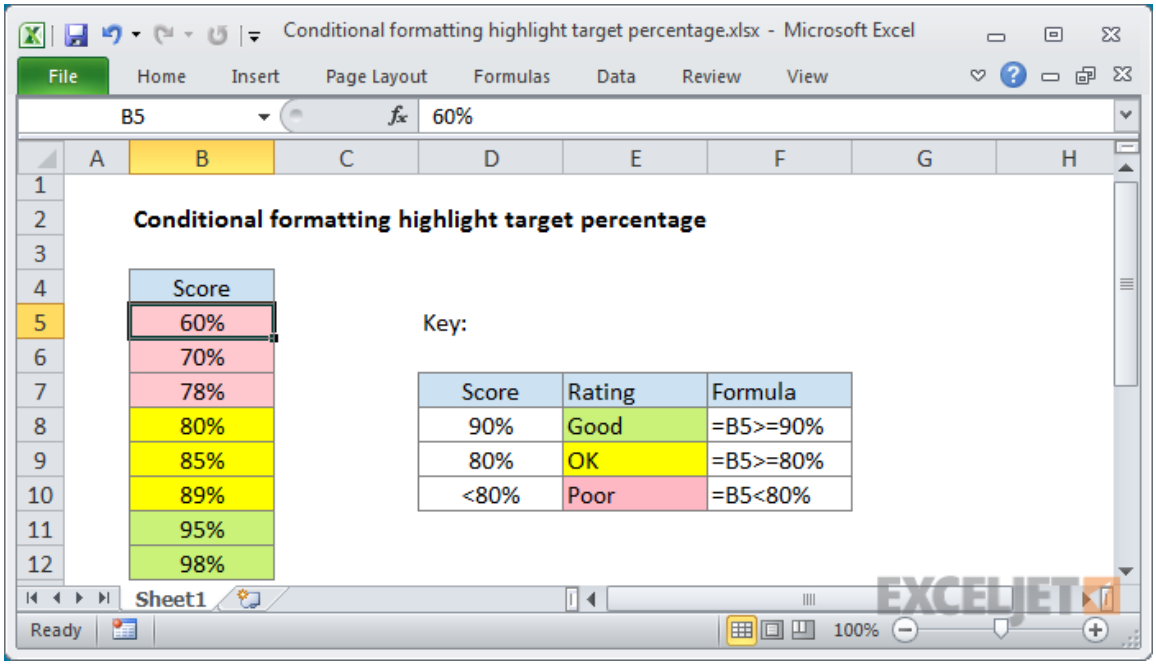
The SQL code for this KPI can be something like this

```
 SELECT tblArticles.Name , tblArticles.Rating

From tblArticles

WHERE tblArticles.Rating <= 3
```

For Example, the final user can create a panel like the image below to leverage this KPI.



Security Concerns

The nature of this dataset and website is not very security-based, and it's safe to say that my website doesn't need a heavy security plan to protect its content because every content on the website is free to use.

However, some security concerns are needed to be addressed. First of all, the real identity of inspectors should always be kept hidden from the public. Also, there are some concerns about spamming in comments which is the most common problem on the web nowadays, so we have to address that by checking the provided emails before letting a secret word.

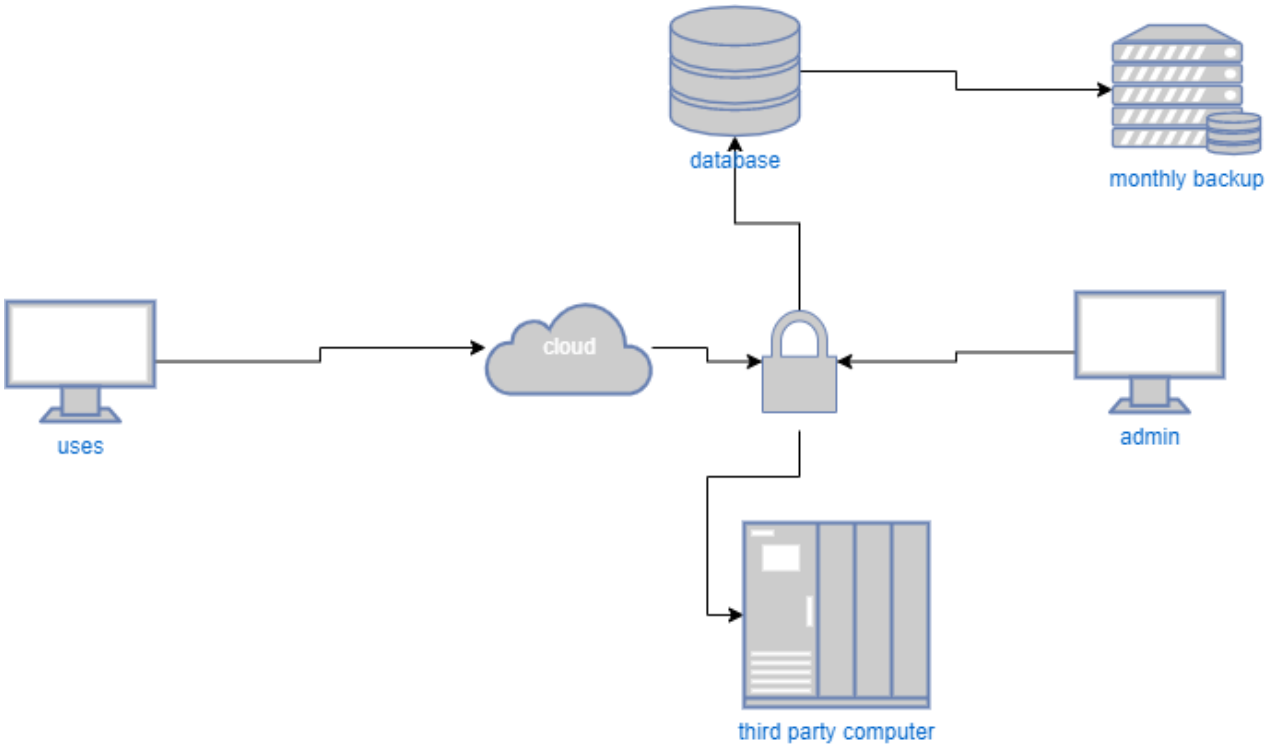
My database architecture

Due to the nature of the website and the need for it to be expandable and reliable, the best choice for the hosting model would be the cloud-based solution.

The website will start small with just one university as a pilot and then scale up. Still, because the nature of the project is heavily data-centred, maybe we would need a medium-size space on a server for the database at the beginning.

The architect of the project is the trickiest part because I want my website to have the ability to use peer-to-peer architecture. At the same time, it stays efficient and straightforward because I don't want to make it always depend on a third party for required computation. Maybe a hybrid solution would be the most beneficial. We start with the server-host model, but we move to the P2P model as we scale up. If this becomes possible, we can have the best of two worlds.

Also, there should be a monthly backup, so if anything happened to the cloud server, there would be another copy of the articles in this way. The authors can trust us better.



Project Wrap-up and Future Considerations

I learned a lot during this project, and I believe something like this project can one day be possible to exist I hope when someone create some thing like my vision every one use it