

Nodes Connect

Social Network for Developers

Authors

Boyce, Scott

Bucknam, John

Middleton, Mike

Ortiz, James

Rivera-Lau, Stephen

Wild, Sebastian

Table of Contents

1. Introduction	2
2. Project Goals & Features	3
3. Database Design	5
4. UML Diagram	8
5. Prototype	9
6. Prototype Snapshots and Descriptions	10

Introduction

It is common for developers to use tools such as GitHub or BitBucket to store repositories and information on projects to be passed on to other users. However, this difficulty grows with larger members are larger projects. For instance, GitHub only supports public accounts freely and BitBucket limits users to 5 members per a private repository. Although there are other options, there is then the thought on communication whether through Instant Messaging or posting through some social network like Facebook, or perhaps wanting to use issue tracking tools like JIRA which already cost money. For basic developers starting from scratch, this seems like too much on its own.

Then comes ***Nodes Connect***, a social network site used by developers for developers. Our social network hopes to bring closure to independent developers by allowing for better collaboration and tools, not to mention better profiling for developers. By having different devs work together in “Collabs” for projects that can be public, protected, or private, they are free to has as little or as much control as they want with their project members as well as their resources. Members will also be able to ask questions in the Q&A and look for other work in the Freelance section of our site to view what is available for them either as a resume booster or a small-source of income.

Project Goals & Features

Although we can't make a perfect service within our timespan, we hope to take steps forward in developing key features. However, these features and future features should follow these basic goals:

- Create a website resource for coders that allows easy collaboration on coding work of all degrees.
- Build a comprehensive search tool used to find relevant projects and coding examples.
- Provide a utility for project collaboration that connects users to projects they are interested in working on.
- Create a forum-like system for users to receive help with coding from other users.
- Provide an in browser text editor for easy posting of small size code; our editor will allow other user to get these small code bits.

These goals were previously mentioned in our project specification and defines our overall deliverable in the end. Because of these goals, our project features must reflect thusly, having our features becoming:

- Stack overflow esque tutor mentor system with large filter and search options. Way to present search tool
- Project collaboration finder and searcher
- Free lance work system
- Profiles reminiscent of a person's webpage to put resume, projects, and careers on display
- Social media integration

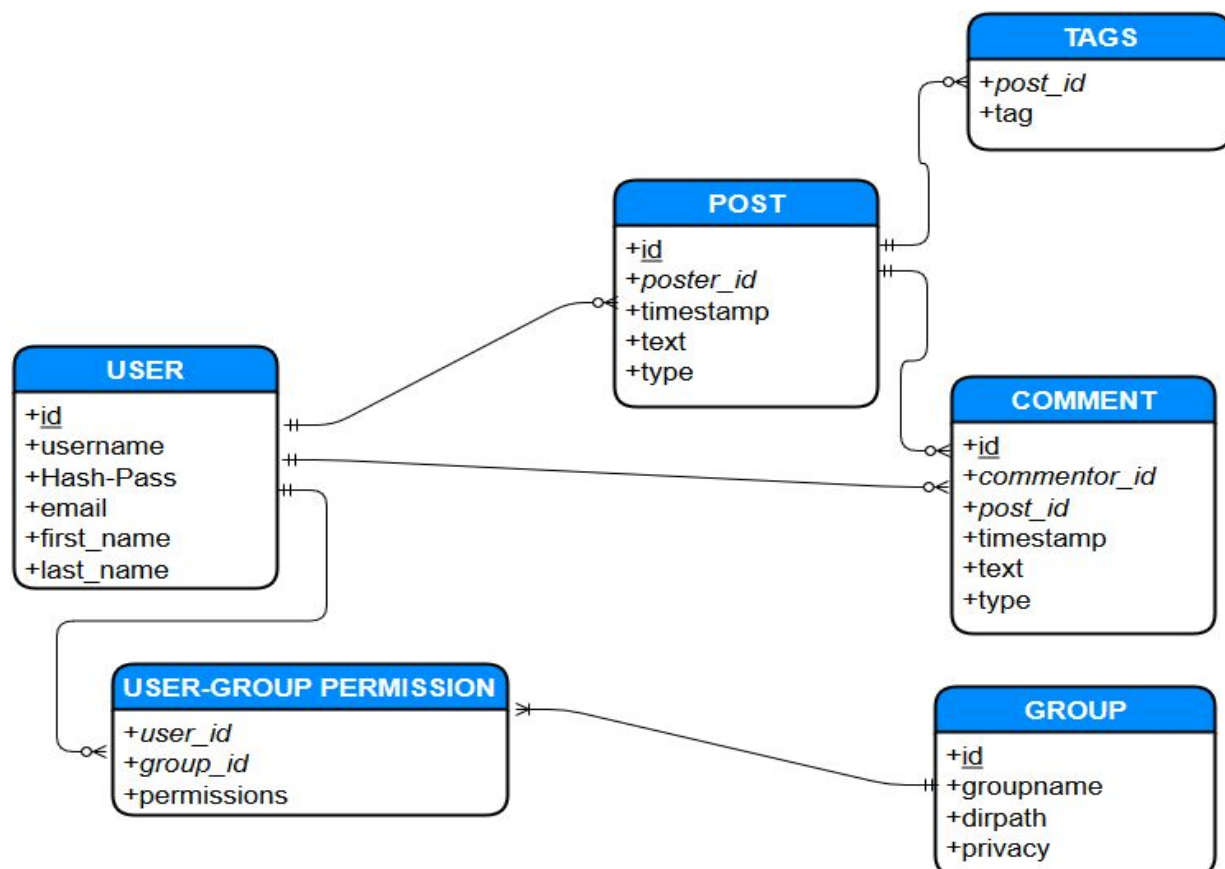
- Direct messaging system
- In browser text editor

Note that not all of these features are explicitly defined in our goals, such as social media integration. However, by accomplishing each of these features, we are then showing that all of our goals will be fulfilled.

Database Design

ER Diagram Legend

+attribute_name	Indicates the name of attribute; related to what will generally be stored in that column.
+ <u>primary_key</u>	Uniquely identifies the row entry in the table.
+foreign_key	Uniquely identifies a row of another table.

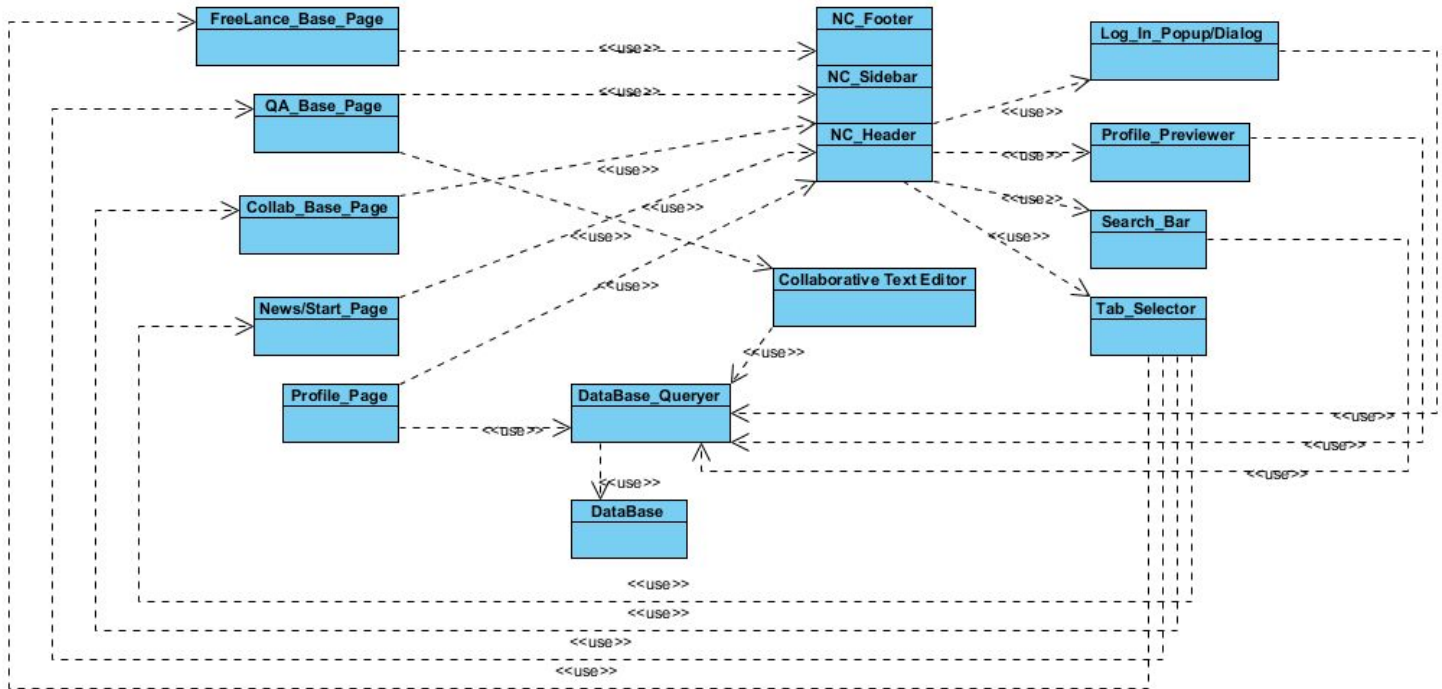


The Entity-Relationship Diagram (ERD) shown describes the minimum that we perceive needing from our database in order to store our information. This can clearly change depending on some organization or perhaps adding more attributes, however these changes are miniscule. For instance, our USER will have username, email, and password at minimum, but this can also include their first name, last name, gender, date of birth, profession, status, and more. However, these attributes are simply additional information that we can implement in some regards possibly in their profile.

USER will contain all basic information on individual developers and users in our site. Although this can include their first name, last name, phone number, date of birth, and other miscellaneous information, the main components of this USER is their username and e-mail address which must be unique and not empty, as well their password which must not be empty. When added to the database, a user is given a unique identification number which will be stored and referenced in other tables in order to create a relationship. An example would be if the user has made a post on the website. Their post will be added to the database and the original owner of the post will have their unique number added as a foreign key (seen as italicized text on the ER-Diagram) in order to pair it together. This will easily allow to look up the owner of posts as well as comments. Similarly, TAGS will have a relational foreign key which will give it reference to its associated post for searches. USER_GROUP_PERMISSIONS will contain a reference to the USER whom it belongs to while GROUP contains necessary information about the group such as name and privacy level, as well as filepath. Note that one GROUP is one Collab Project, however this can change to be a group of contacts over time. This is not a main focus of our site and will more than likely not be added unless done as an additional feature.

A USER can also make one more POSTs, and each POST must have one USER that posted it (poster_id). A POST will have the text itself as well as a type (Collab, Q&A, Freelance, etc.) that will be a number defined by an enumeration. It will also have a timestamp to mark when it was posted to better date in relevance. COMMENT is similar to POST, having the same relationship to USER, except that COMMENTS can be added to POST in response. We could add other attributes for things such as images, but depending on our database, we may be better off either storing the image as a binary file or some other setup. This is a small feature though since most images can be linked and it may be better for us to simply read text in a specific format such as Markdown or HTML.

UML Diagram



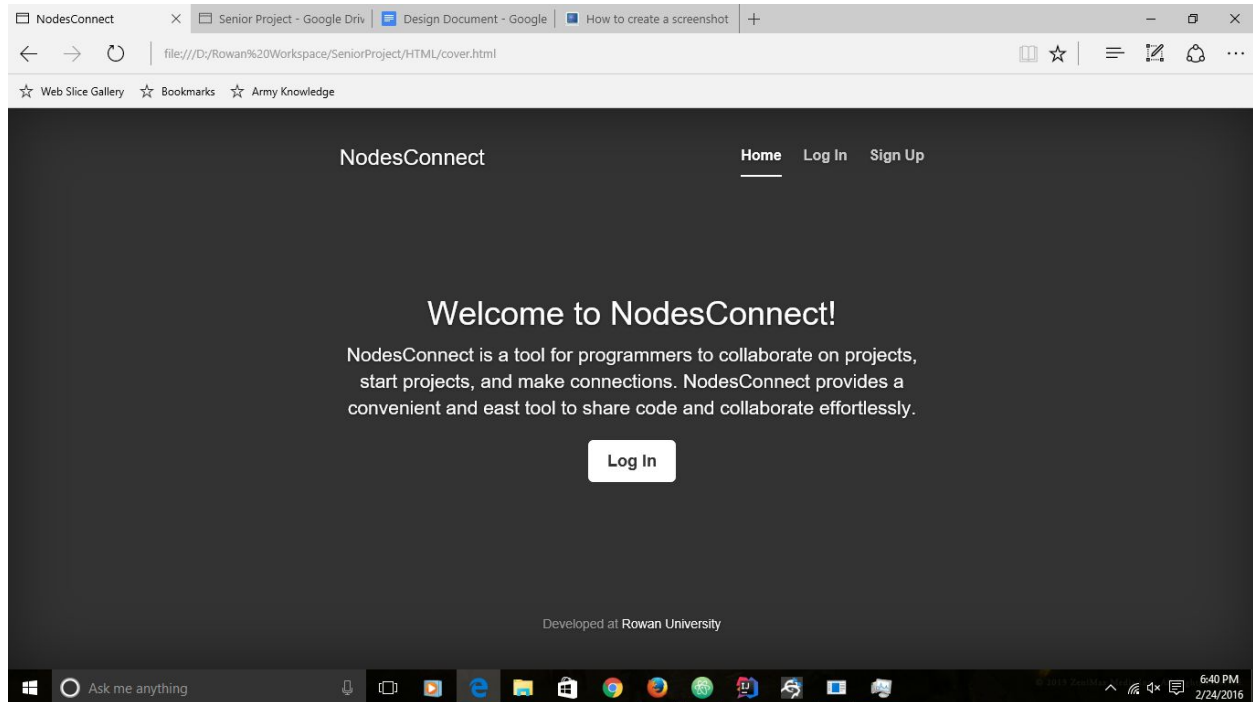
The UML diagram pictured above visualizes the basic structure of the Nodes Connect platform. In the upper right there are classes that represent the five main pages of our website. Each of those pages and nearly any other web page on our site will use the NC header, footer, and sidebar. These controls give every page on the site a sense of familiarity and will allow for easy maneuvering between sub sections. Although it is not explicitly shown in the diagram every web page will query the database for information to be shown on the page, though the search will send the request for the data to be displayed most of the time. The database queryer is a facade for the actual interaction with the database, it is used to concisely represent the need to pull information from the database.

Prototype

The prototype for Nodes Connect is a series of html pages “linked” together. Starting from a cover page, a login and signup are found, which later on will allow users access to the main dashboard. The main dashboard allows easy access and navigation to the other features and pages of Nodes Connect, such as the profile page, Q&A page, collaboration page, freelance work page, and quick accesses to the text editor, current projects, searching, and messaging. The web pages are designed using html, CSS and javascript.

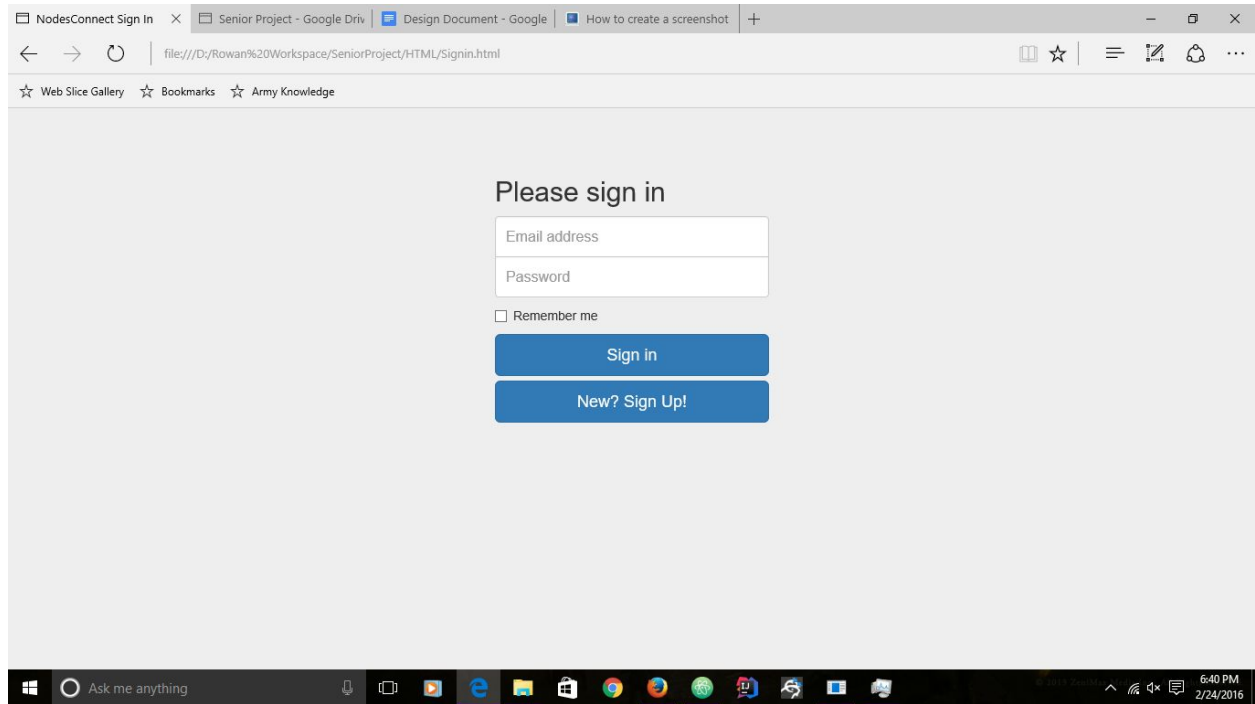
We will be going through each page step-by-step, starting from the front page to the user’s “dashboard.” Although this is may be incomplete, keep in mind that this is a prototype for our final product. There are also features that will stay consistent, such as the navbar for user pages and other tools in navigation.

Prototype Snapshots and Descriptions



Index Page : Welcome

This front page is acts as the basic login/signup for anyone who visits our site. This probably won't change much over time since it is the least significant part of our website and only acts to access the login or signup pages.



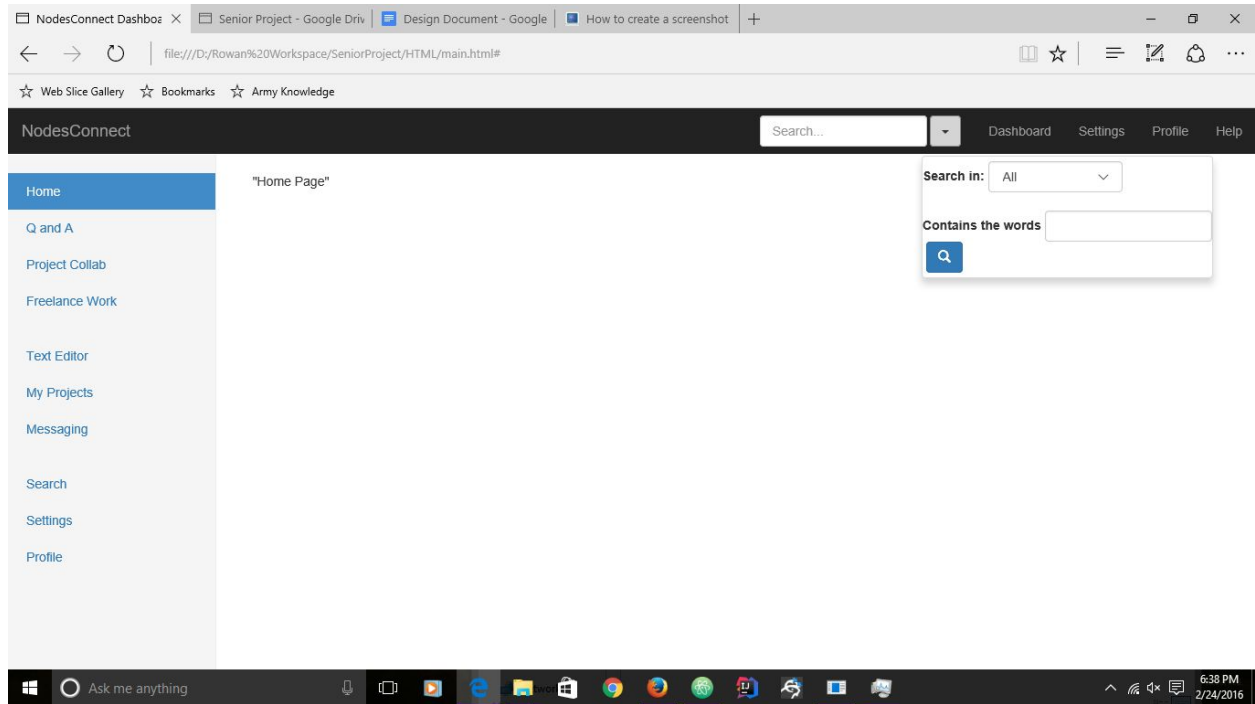
Sign In Page

This is the basic sign in page that simply asks for the email address and password of a user. It's only job is to redirect to the user page when given the correct information, otherwise keep asking until it fails. Obviously we could add more dynamic features such Captcha locks, but this isn't the main focus for our site.

The screenshot displays a web browser window with multiple tabs. The active tab is titled 'NodesConnect New Use'. The address bar shows the file path 'file:///D:/Rowan%20Workspace/SeniorProject/HTML/NewUser.html'. The main content area features a 'Create a new account' form. The form consists of the following elements: a title 'Create a new account', input fields for 'First Name', 'Last', 'Email address', 'Confirm Email address', 'Password', and 'Confirm Password', radio buttons for 'Male', 'Female', and 'Other' (with 'Other' being selected), a 'Date of Birth' field with a placeholder 'mm/dd/yyyy', and a blue 'Sign Up' button. The browser's status bar at the bottom shows the time as 6:40 PM.

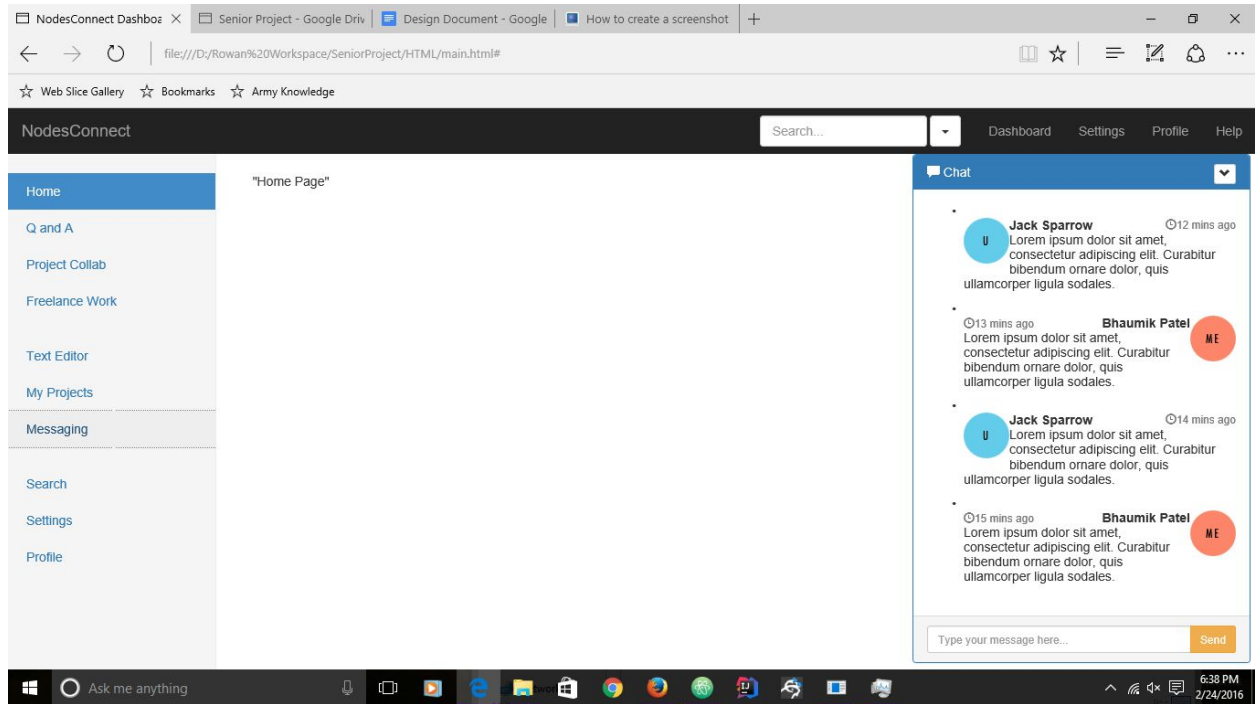
Sign Up Page

This is our current sign up page for new users and asks basic information to be stored on our database. This includes first name, last name, e-mail address, password, as well as gender and date of birth. More could be added on if necessary, such as secret questions in case of a user loses their password, or more questions to fill a profile. Those types of data can be added on later when we have more time. After signing up for an account or signing in, the user can then access their page, having the ability to edit their information and use other services.



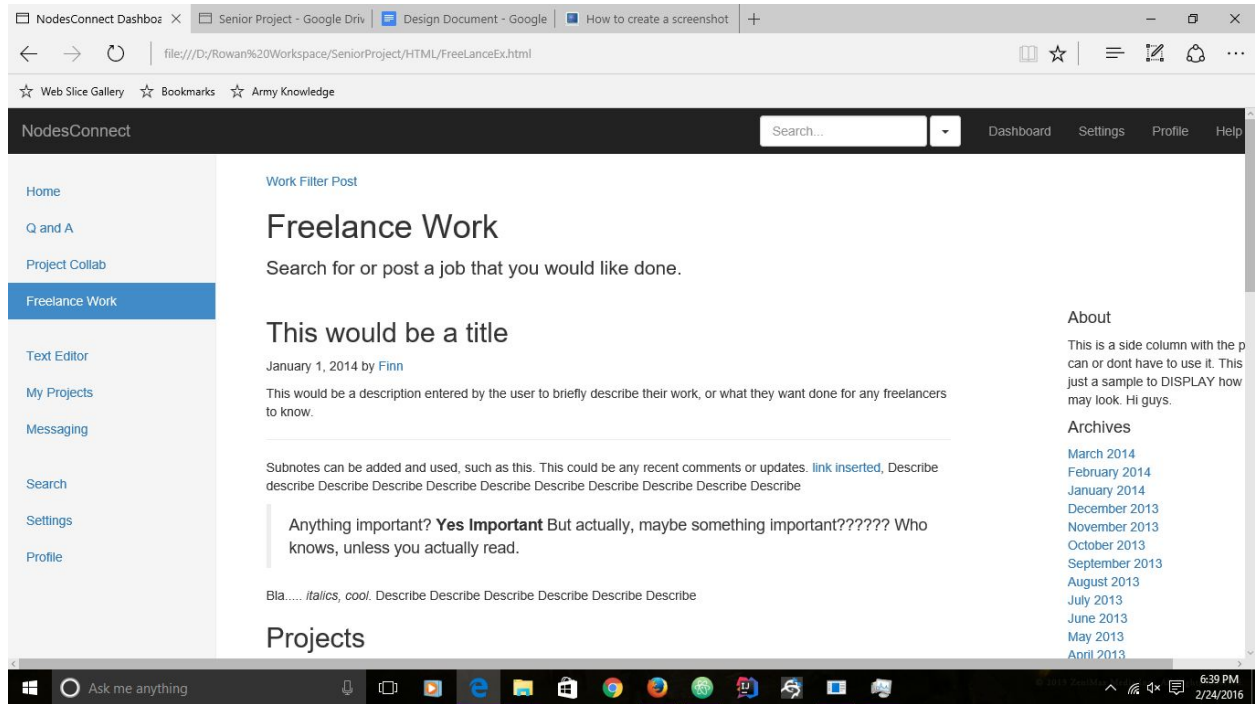
User Home Page

The Home page should contain a summary of the user's network life from different projects, collabs, freelance work, as well as questions asked and answered. Of course at this given stage it is fairly bare, but shows how most other pages will look as well, with shortcuts on the top and side and the main page being tugged to the right. We will have some type of features or components that could be added to this page that will give more light to the user's progress not just in projects and freelance work, but as a community member as well. Ways we can in fact bring light to this idea of "community builder" is by giving out karma points or something similar that shows some type of responsibility or growth over time. This of course is an additional feature and will be added after all main features are implemented.



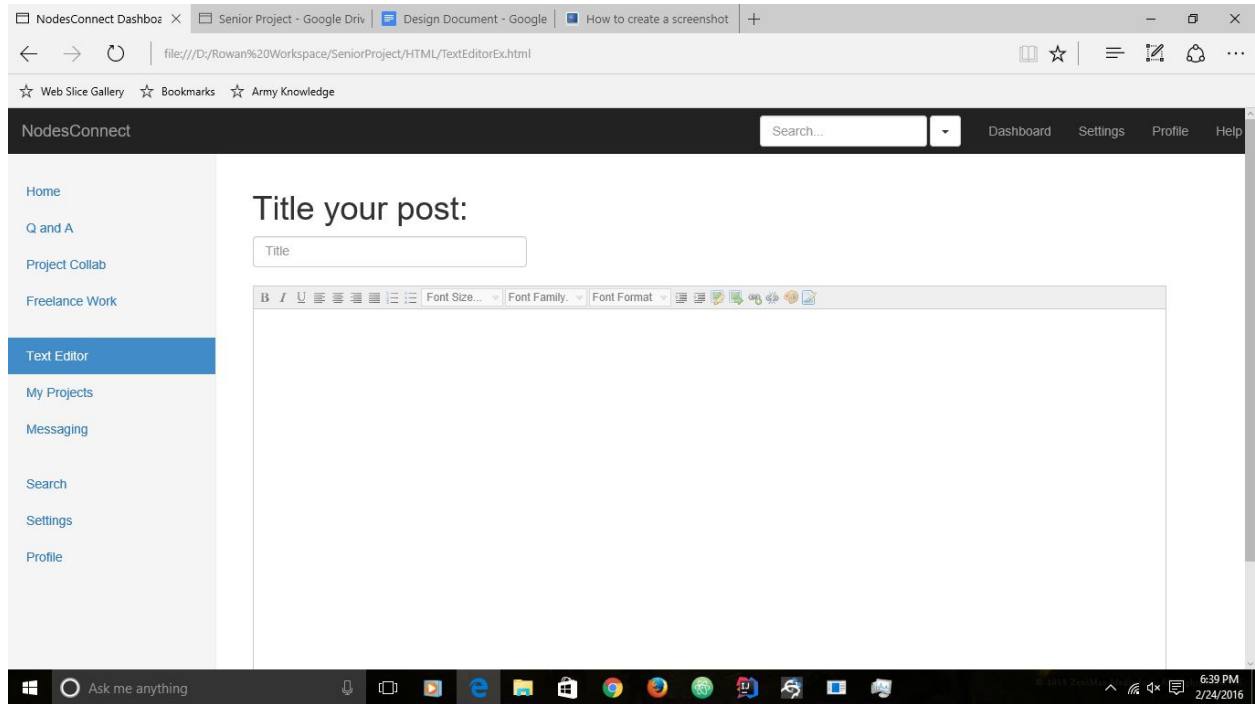
Message System / Chat

This is a basic messaging system that pops up when pressing the messaging nav-button on the left side. This allows our users to chat with one another without having to go through some other system of messaging such as Skype, Slack, Google Hangouts, or Facebook where it can be cumbersome or clunky at times.



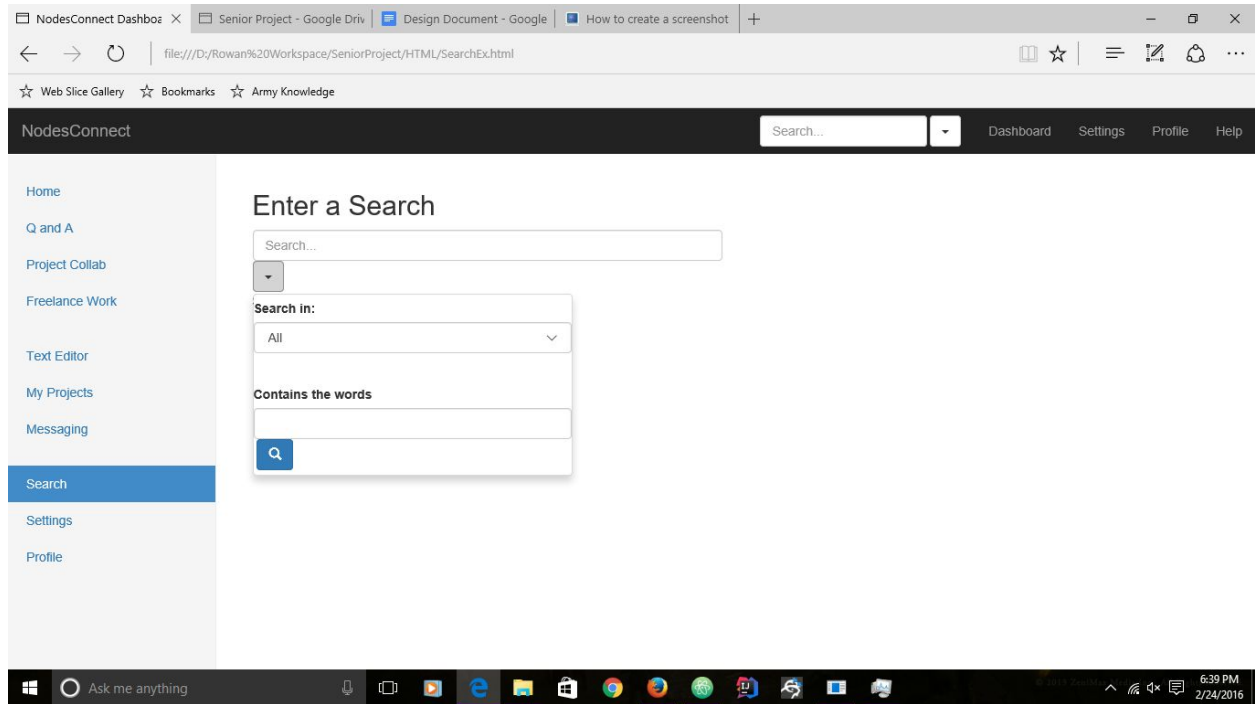
Freelance Work / Posts

This is an example of our freelance work page, displaying all the different possible posts for freelance work that other users wish to have done either for free or a small amount that they must discuss through some chat.



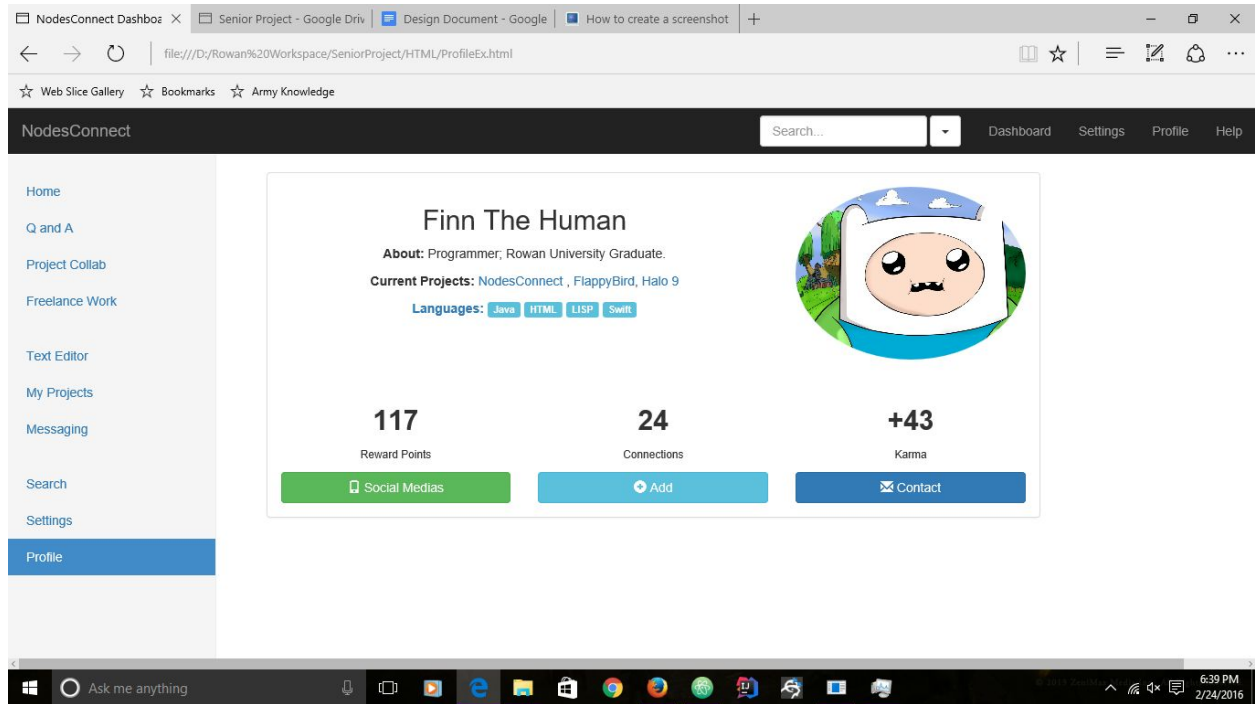
Text Editor

The basic text editor allows our users to create new files, and edit old files, through a basic terminal. Although it will be fairly small, it will allow our users to do quick programming online without any IDEs.



Search Page

The search page allows our users to search different Q and As, Collabs, or Freelance Work that have specific keywords, or ‘tags’, as we refer to. We are then capable of using our database to search for the corresponding tags for those results and from there we are able to return a result list. This example shows the basic text-field that users have with them. Also, on the top of the page, as with every page, there is a search bar that can reference back to the same search tool as well.



Profile page

The profile page is an open summary of a user's work, including current projects, information about themselves, as well as programming languages that they are using. This can be further include collaborators, questions asked, answers given, jobs finished, and more. This also goes into the idea of karma points as an idea of community building, but once again these ideas will be implement last if at all.