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. Bugs Found and Fixed 10
6. Prototype 11
7. Final Snapshots and Descriptions 12

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 and 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, each of them have their own form of communication whether it's through the use of Instant Messaging, posting through a social network like Facebook, or even using an issue tracking tool like JIRA which already cost money. For beginner developers who are starting from scratch, this can be a lot to manage and work with on it's 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, tools, and profiling to provide an easy and manageable experience. By having different devs work together in "Collabs" for projects that can be public, protected, or private, they are free to have 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 section 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. The features we include in our initial product as well as future releases 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 define 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
- Project collaboration system
- Free lance work system
- Profiles to view key information about users like biographies, projects, connections, social media links, and functionality to add/remove connections.

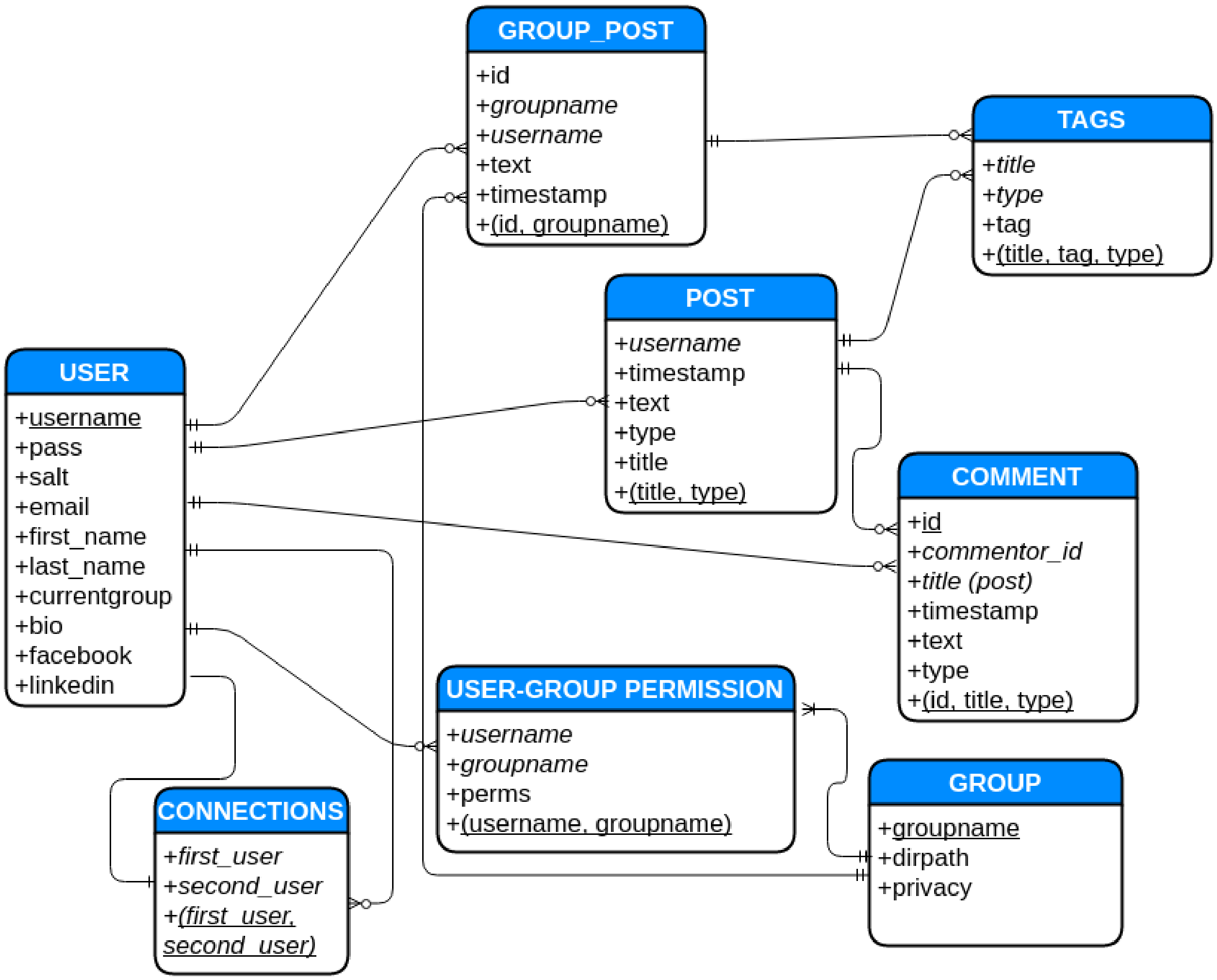
- Robust Search tool that can search the entire website for all user created data based on tags, text, or useremail.
- Social media integration
- Direct messaging system
- Use Collabedit to facilitate code editing between users of the site

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.
+primary_key	Uniquely identifies the row entry in the table.
+*foreign_key*	Uniquely identifies a row of another table.



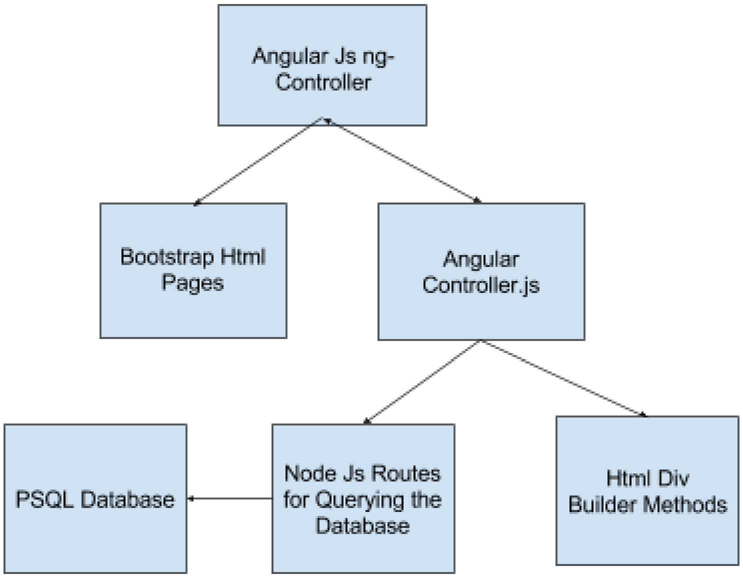
The Entity-Relationship Diagram (ERD) shown describes the minimum that we perceive needing from our database in order to store our information. This can clearly change depending on some organization or perhaps adding more attributes, however these changes are miniscule. For instance, our USER will have username, email, and password at minimum, but this can also include their first name, last name, gender, date of birth, profession, status, and more. These attributes are simply additional information that we can implement into a user’s individual 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 USER is their username, e-mail, and password which must be unique and not 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 establish a relationship. An example would be if a user makes 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 a post or a comment. Similarly, TAGS will have a relational foreign key which will give it reference to its associated post which can be utilized 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 the main focus of our site and will more than likely not be added unless done as an additional feature.

A USER can make multiple POSTs, and each POST must have one USER that posted it (poster_id). A POST will have description text as well as a type (Collab, Q&A, Freelance, etc.) that will be a number defined by an enumeration. A POST will also have a timestamp to mark when it was posted to better sort by chronological 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. At this point, we are focusing less on media types and more on basic plaintext which we can later update with more features.

We have added CONNECTIONS to USERS so that different USERS may connect to one another on their profile page. We have also expanded our POSTS so that we have GROUP POSTS and regular POSTS. Regular POSTS are used by the Q&A and Freelance format, allowing us to reference the table for those types of questions. The GROUP POSTS allows us to reference different USERS and GROUPS related to each post. Furthermore, each GROUP and USER has a USER GROUP RELATION which allows a USER to be part of any one GROUP where each GROUP has a unique USER.

UML Diagram



The UML diagram pictured above visualizes the basic structure of the Nodes Connect platform. These five modules are facades for all the classes in the platform but show the basic structure of calls throughout the website. The Bootstrap Html Pages represent all the pages on the website that have dynamic elements which are populated by the database. These elements are listened to by the ng-Controller which call the appropriate method in the Controller to update the divs in the Html Page. The Controller uses the appropriate database Route method to query the database for information. With the information from the database the Controller uses the appropriate Html Div Builder method to construct divs for all the information that needs to be displayed. The controller then uses the ng-Controller to to update the Html in the page with the new divs.

As seen in our UML Diagram, we have a level-based system starting with our database, going down to our html using AngularJS, JQuery, and Bootstrap HTML. We define our layered stack as the PEAN stack, using PostgreSQL as a backend database model that is being controlled by our server. The server itself is NodeJS using Express to route our API and allow us to use different modules such as Node-Postgres, which connects us to the backend database. We may also send information from our database as JSON using our Express and NodeJS, which is used by AJAX calls and AngularJS $http methods. The AngularJS uses its controller to configure what data is being sent to our HTML view, as well as sending data back to our server using our API. We may also use JQuery to populate our HTML with different posts. By doing these steps, we have gone from our model (PostreSQL) to our controller (NodeJS, Express, AngularJS, JQuery) onto our View (HTML, CSS, Bootstrap).

Bugs Found and Fixed

Freelance / Q&A

- Open client connections (which were never closed) on API calls causing the server to crash after a while
- Some minor issues with adding tags on a post

Groups

- Permissions - Users could essentially make themselves an admin of a group, remove other admins and then delete the group.

Searching

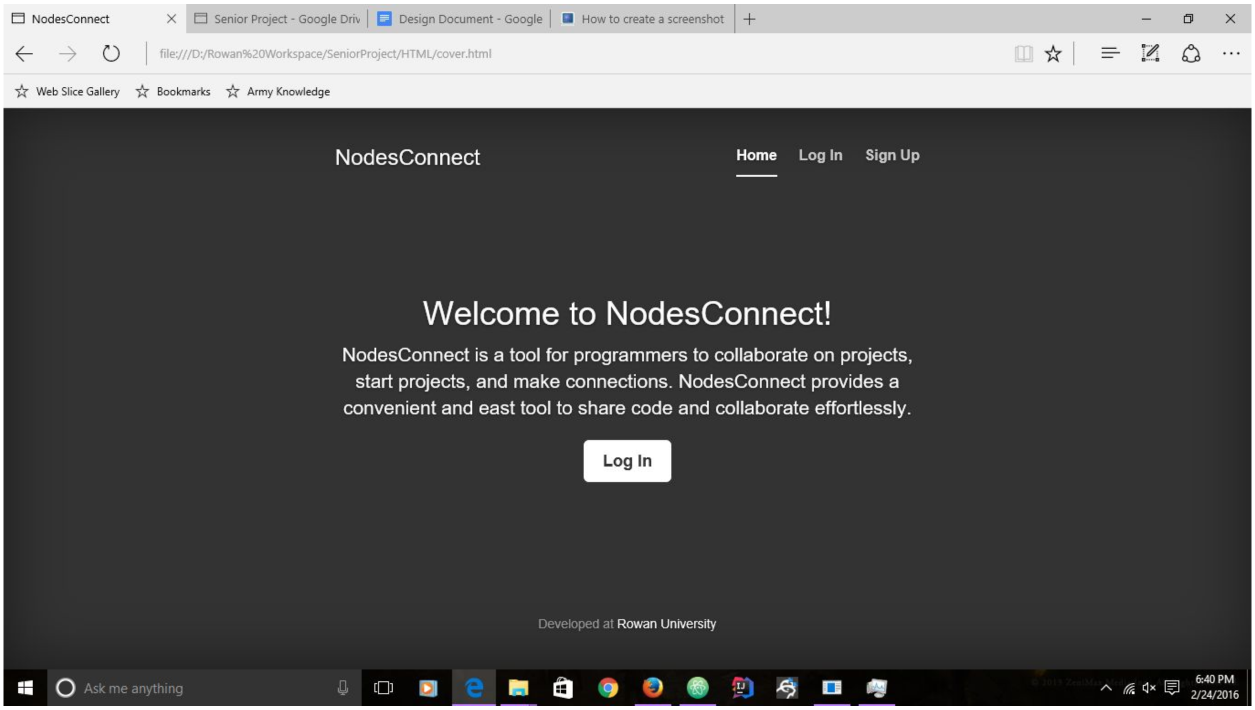
- The creation of a link in the results for groups when a group was private
- Only one result showing up when a search returned multiple objects
- Searched string were case sensitive and not returning expected results because of database formating

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”. At this time we will go into full detail on all the pages and features since they have been completed for our final product.

Final Snapshots and Descriptions



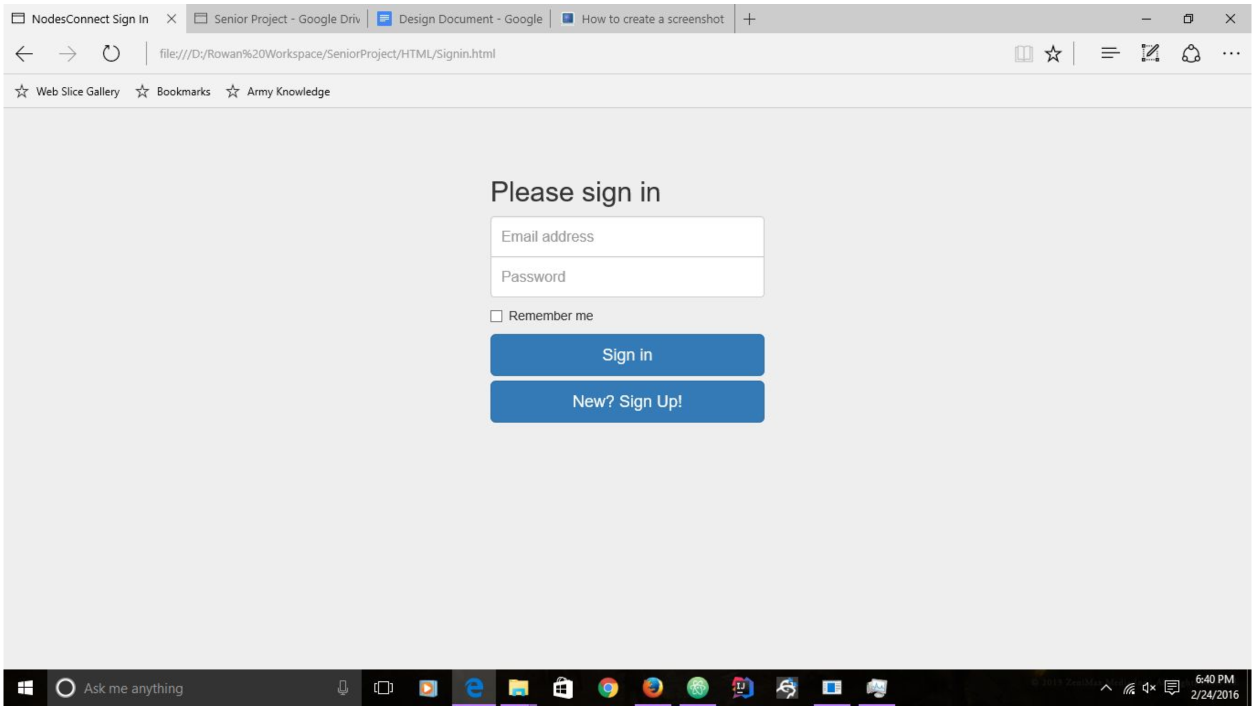
Welcome to NodesConnect!

NodesConnect is a tool for programmers to collaborate on projects, start projects, and make connections. NodesConnect provides a convenient and east tool to share code and collaborate effortlessly.

Developed at Rowan University

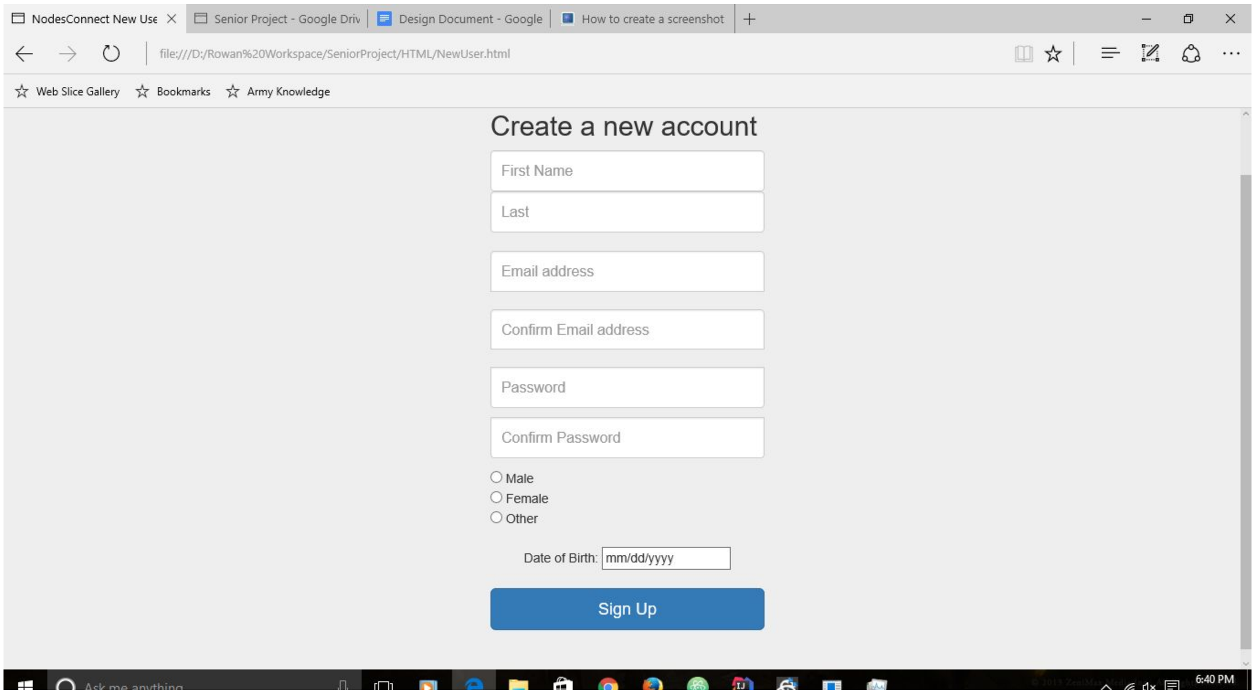
Index Page : Welcome

This front page is acts as the basic login/signup for anyone who visits our site. This page provides new users with a brief overview of our site along with navigation to sign-up or log in.



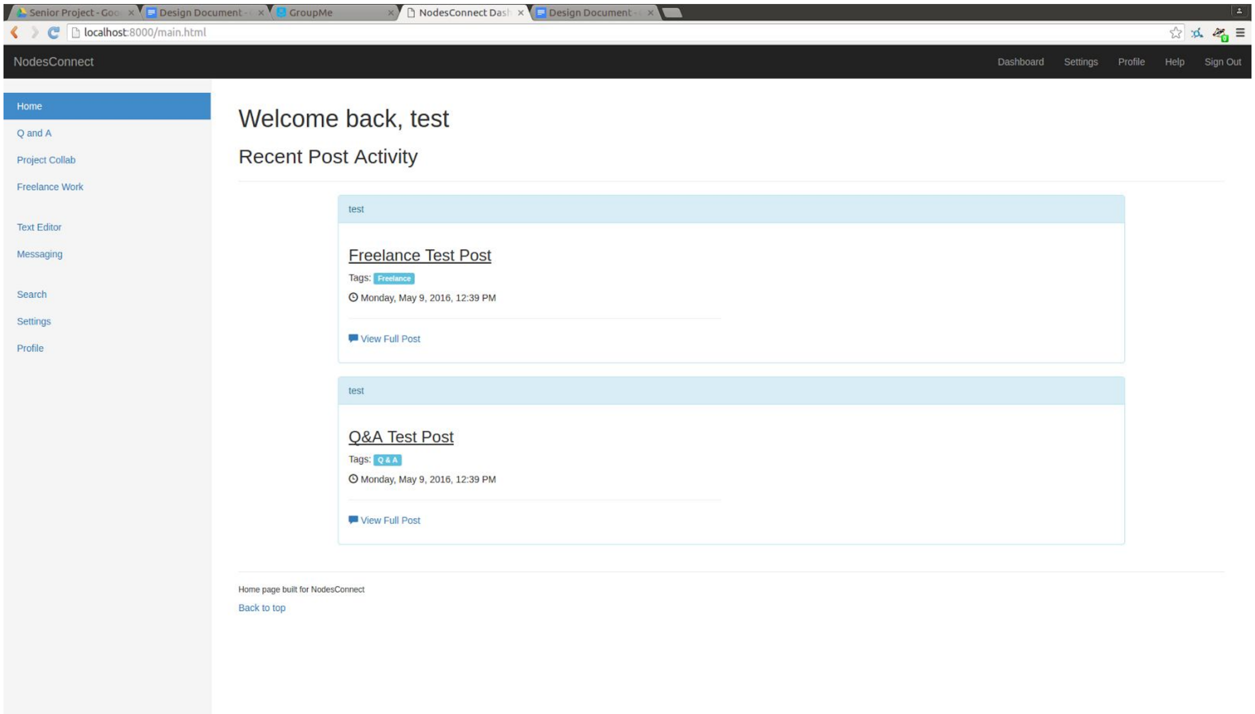
Sign In Page

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. If a user attempts to log in with false information the site will provide a pop up error message indicating the reason for the error. This page also allows users to navigate to the sign-up page if they have not yet created an account. Additionally, we could add more dynamic features such Captcha locks, but this isn't the main focus for our site.



Sign Up Page

This is our current sign up page for new users which asks basic information that would be needed to create an account, all of which is stored in our database. This includes first name, last name, e-mail address, password, gender, and date of birth. More features in this section could be added on such as a secret questions in case of a user loses their password, or more questions to fill a profile. We felt that these features were not a top priority for the final product. Similar to the login page, the user will be provided with error messages if their information is invalid. Invalid information can include passwords that don't match or an email that has already been used. After signing up for an account, the user will be redirected to the login where they may enter their credentials to be brought to their home page. The home page is where a lot of the core functionality of the product can be accessed.



Welcome back, test

Recent Post Activity

test

Freelance Test Post

Tags: Freelance

Monday, May 9, 2016, 12:39 PM

View Full Post

test

Q&A Test Post

Tags: Q & A

Monday, May 9, 2016, 12:39 PM

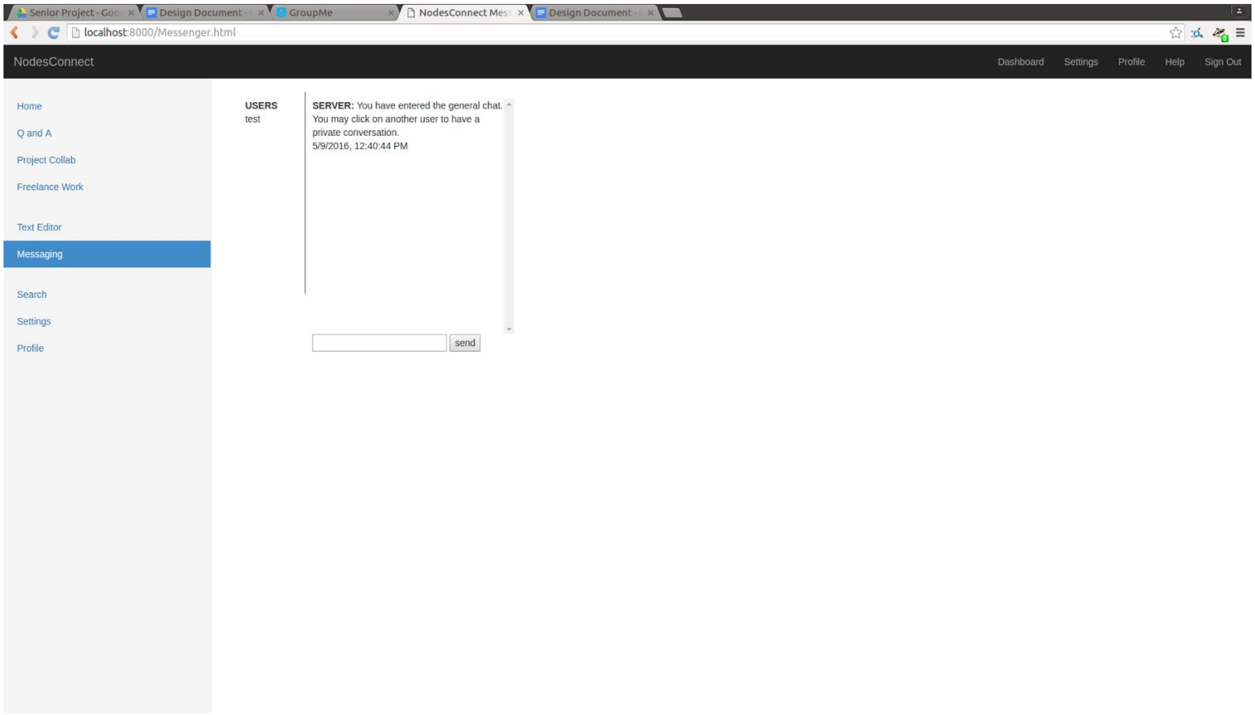
View Full Post

Home page built for NodesConnect

Back to top

User Home Page

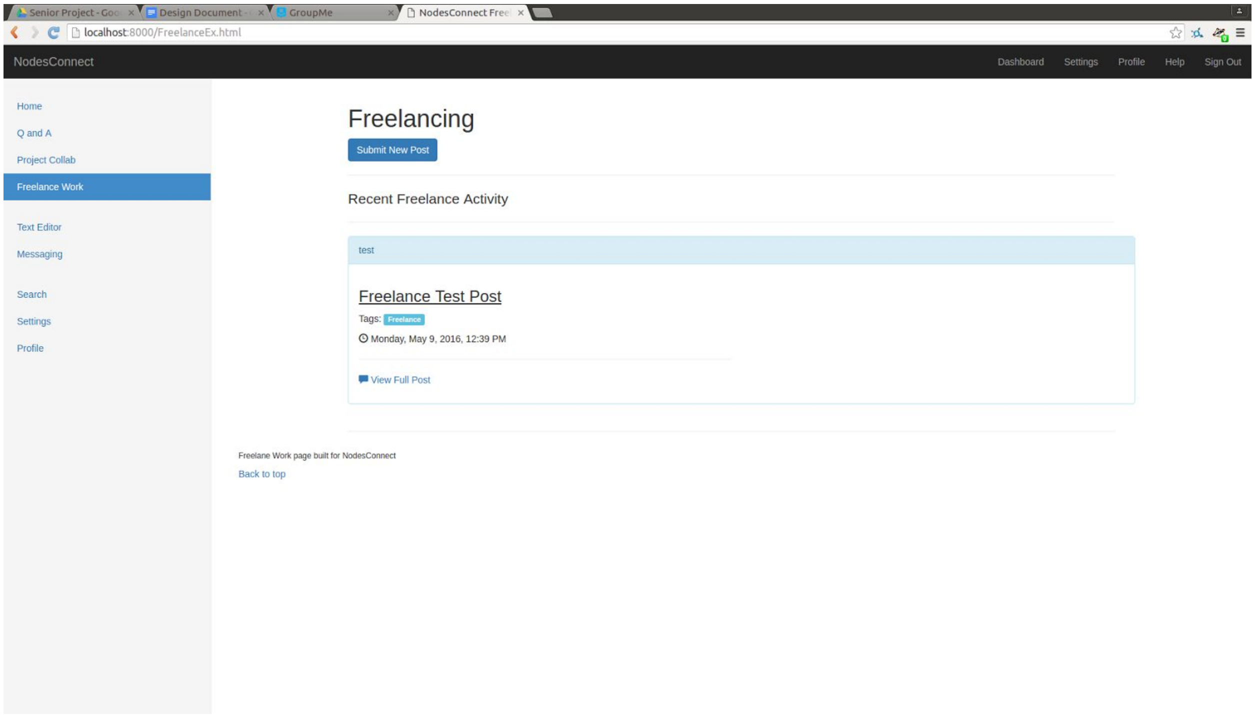
On the User Home page the user will be greeted by the system as well as show them recent post which they have made, if any. This would allow a user to stay update with the posts they have made on the site by provide a single section to view only their posts.



SERVER: You have entered the general chat. You may click on another user to have a private conversation.
5/9/2016, 12:40:44 PM

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. Users have the option to talk with individuals privately or even chat with everyone that is currently on the messaging system.



Freelancing

Recent Freelance Activity

test

Freelance Test Post

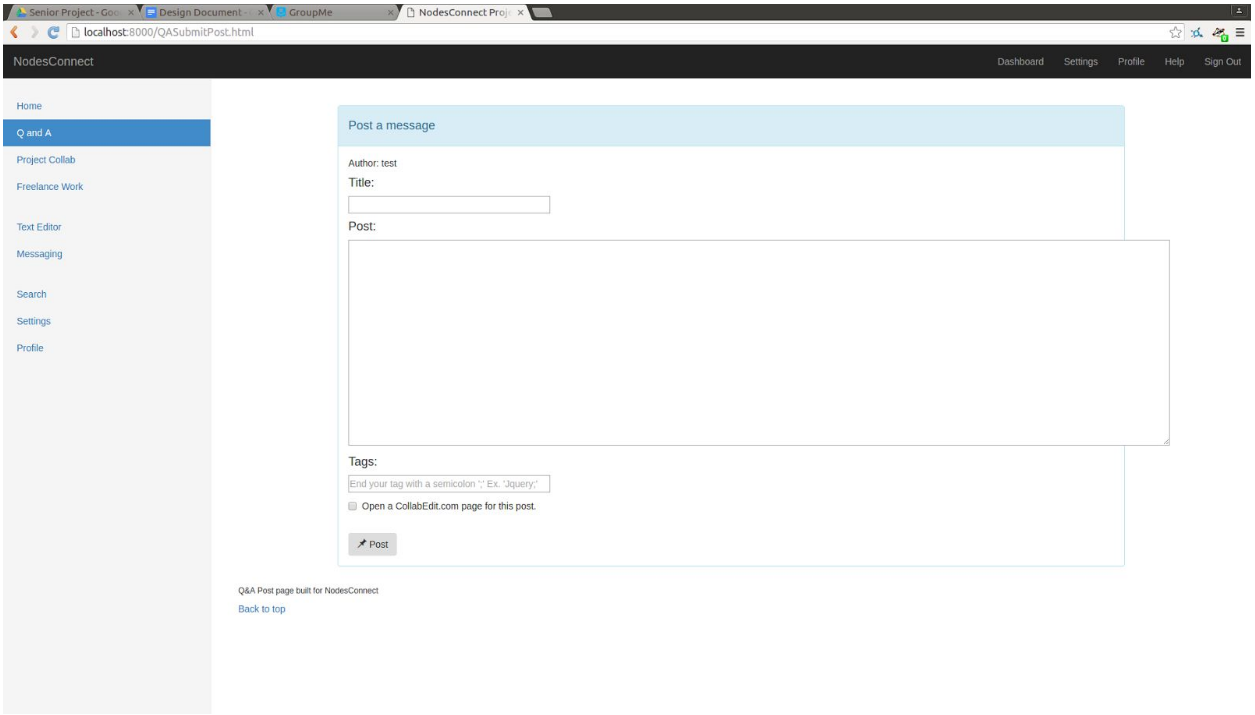
Tags: Freelance

Monday, May 9, 2016, 12:39 PM

Freelane Work page built for NodesConnect

Freelance Work / Posts

This is our freelance work page, displaying a sample post. Here users can find freelance work that other users wish to have done either for free or a small amount that they must discuss through some chat such as our messenger or through the Project Collab post system.



Post a message

Author: test

Title:

Post:

Tags:

End your tag with a semicolon ';' Ex. 'Jquery;'

□ Open a CollabEdit.com page for this post.

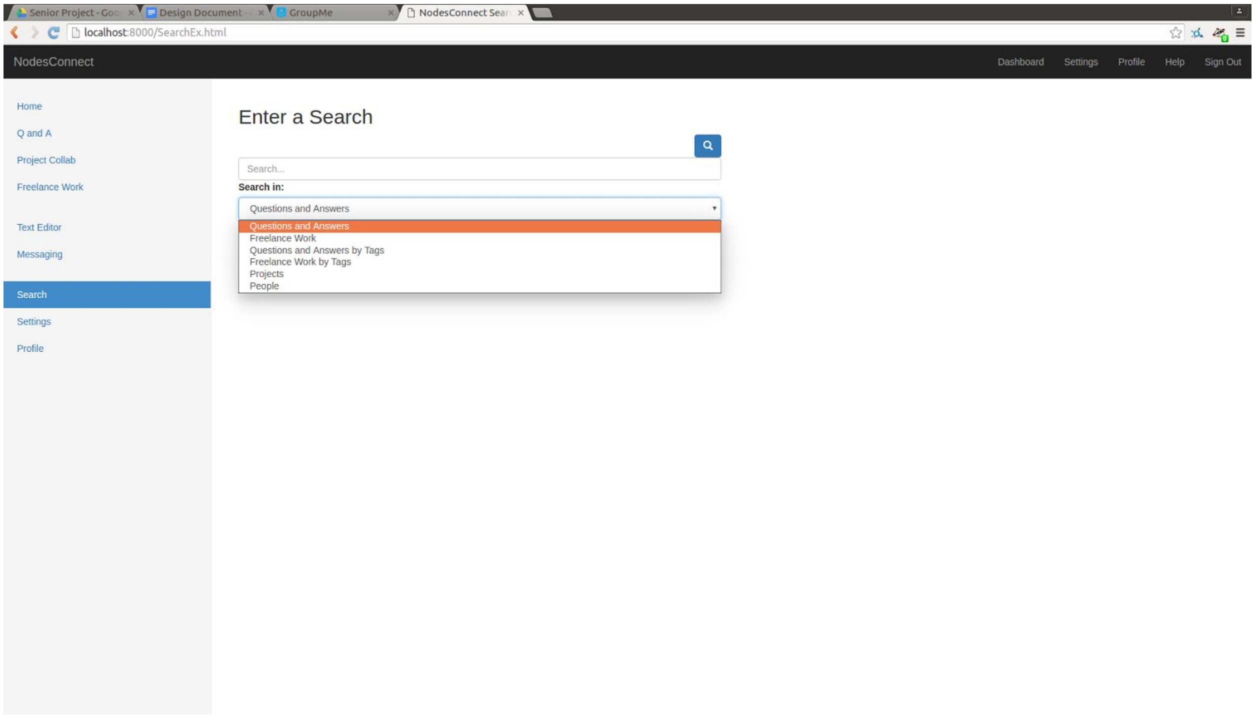
Q&A Post page built for NodesConnect

Post

Our basic post system collects a unique title with the post content to be shown to any other user. We also use tags in order to find aid in searching for specific categories of questions or freelance work. For our Q&A post, we can also open a text editor page. *Note - It should be noted that the javascript used to build tags for posts utilizes a keydown event which Chrome currently has a bug and does not register. Firefox seems to work fine.

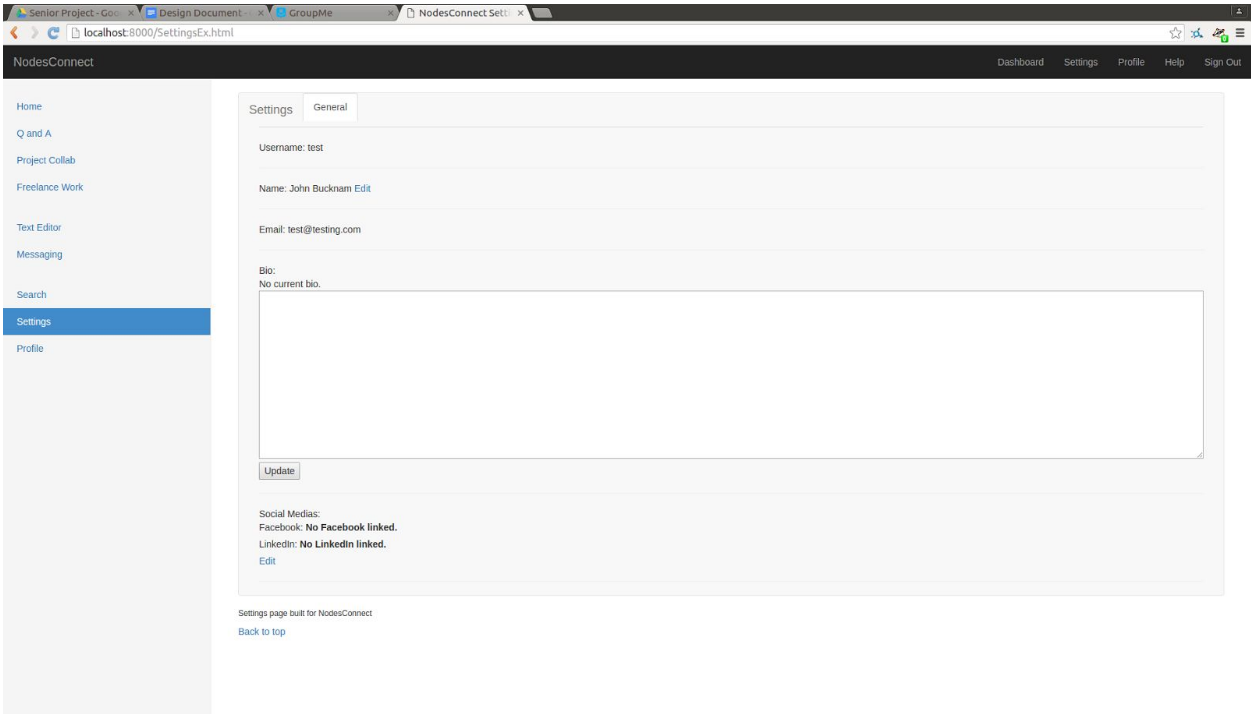
Text Editor

Our current text editor uses a third-party site, CollabEdit, in order to have multiple users collaborate on a similar project. This can be accessed by using the Text Editor tab or by choosing the option in the Q&A post if needing assistance with code.



Search Page

The search page allows our users to search the Nodes Connect Platform for Posts, Groups, or User Profiles. The search works in a way similar to most other search tools, the user types in a string they want to search for and choose where they want to search for that string. The user can specify their search for Q and A Posts by tags or title, Freelance Posts by tags or title, People by username or email, and Groups by group name. Searching for a title or group name will be split on spaces and searched for all the words in the search text. Similarly tag searches will be split on commas and posts containing one of those tags will be displayed. Last, when searching for people the search text will be split on a colon. Text previous to the colon is assumed to be the Name of the user and text after the colon is assumed to be an email. Users that have a matching username or email will be displayed. If a search has found matching data the page will be updated to show the results and each result (except for private groups) will have a hyperlink to the full version of the results a post, profile or group.



Username: test

Name: John Bucknam Edit

Email: test@testing.com

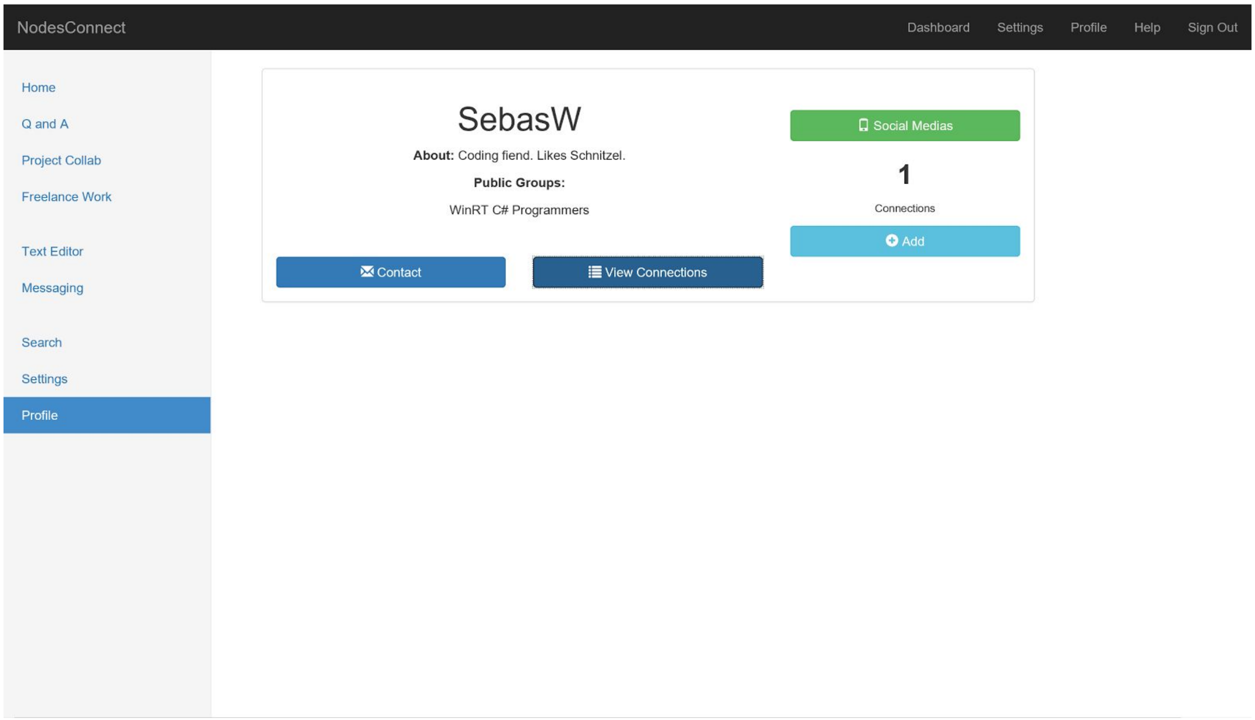
Bio:
No current bio.

Social Medias:
Facebook: **No Facebook linked.**
LinkedIn: **No LinkedIn linked.**
Edit

Settings page built for NodesConnect

Profile Settings

These settings are given to each user, allowing them to edit their First and Last name, update their bio, as well as add social media connections.



SebasW

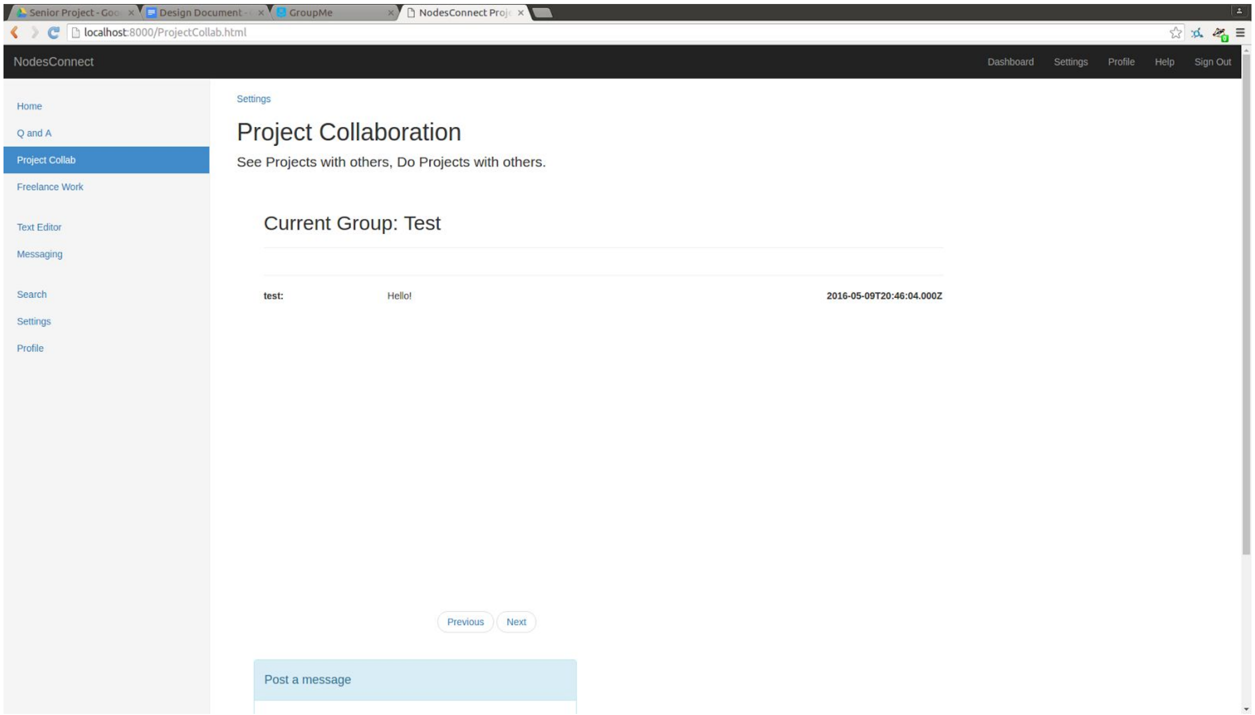
About: Coding fiend. Likes Schnitzel.

Public Groups:

WinRT C# Programmers

Profile page

The profile page provides an overview of a user. Information like a biography and current public groups the user is member of is displayed. Further functionality is provided by the various buttons placed on this page - the “Contact”, “View Connections”, “Add”, and “Social Medias” button. The contact button will trigger an e-mail to the user, and “View Connections” will present the user with a modal popup dialog listing the user’s connections. Next to each connection name in this popup, controls are displayed for the functionality of viewing that connection’s profile as well as removing him/her from the user’s connections list. The “Add” button will present the user with a prompt to type the username of a user they’d like to add to their own connection list. Above the “Add” button, the user will be able to see his/her number of connections. Finally, the “Social Medias” button, when clicked, will show the user’s social media links in a modal popup.



Project Collaboration

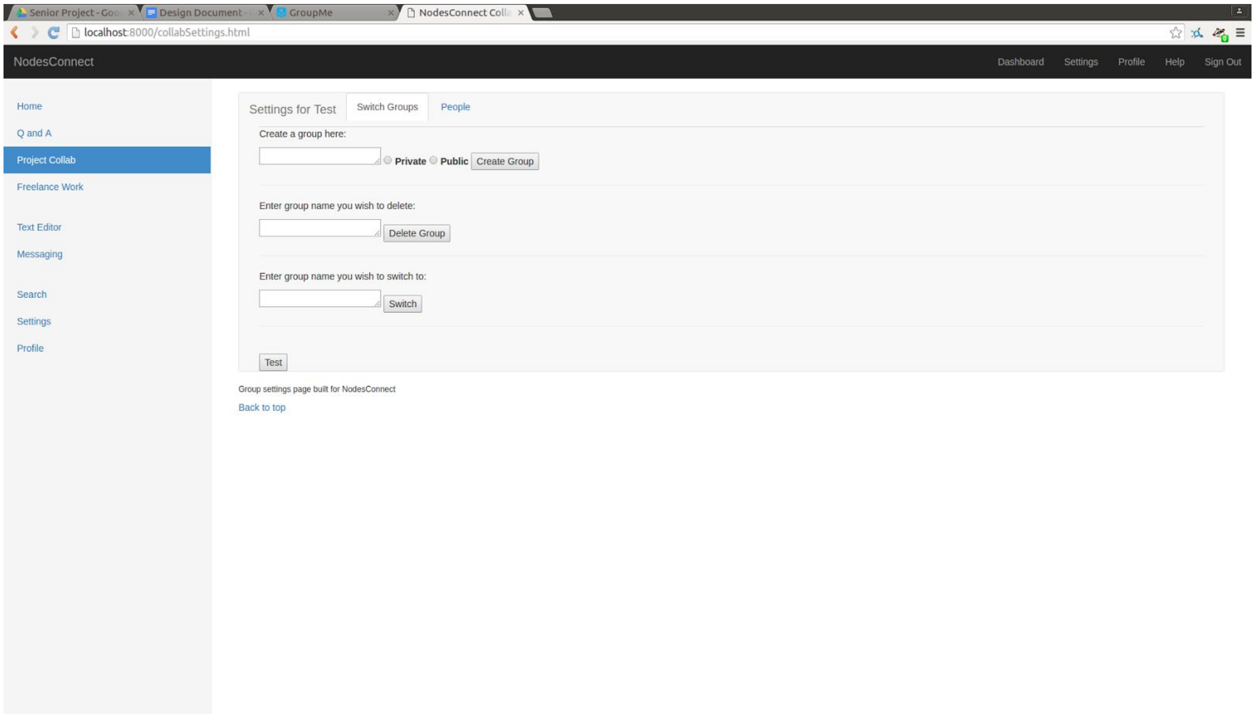
See Projects with others, Do Projects with others.

Current Group: Test

test: Hello! 2016-05-09T20:46:04.000Z

Project Collab

Our group page, Project Collab, is able to send messages that can be stored within our database and is only visible to other group members. We are able to post new messages on the bottom, as well as view previous messages that can be viewed in multiple pages. We are also able to see who posted it, what message they posted, and when they posted it.



Create a group here:

Enter group name you wish to delete:

Enter group name you wish to switch to:

Group settings page built for NodesConnect

Project Collab Settings

Within our Project Collab page, we are able to configure our group for different permissions as well as add or remove members. In this first tab, we are able to create a public or private group, delete a group we have administrative access to, as well as switch to a group that we are part of.

In the People tab, we are able to add new members as either an Admin or regular Member. Admins can delete other members and delete the group while members can only view and add posts as well remove themselves from the group.