**Final Report**

*Blog Generator*

*Group Members: Jun Yin, Yue Wang, Lee-yin Wang, Huan Wang*

**1. Introduction to background**

People who love writing may go through 3 phases:

*Phase one*: Newly play with blog and feel really interesting, so start to find a free blog system to blog.

*Phase two*: Figured out there is too many limitation of the free blog system, then buy a domain name and build up their own blog website.

*Phase three*: Feel that there is so much work to do when really own a website, it will be much better if only focus on writing and some mechanism exists there in charge of the whole managerial tasks of the website.

Most of bloggers stay on phase one and phase two, because it is much harder to get to phase three: no such a free mechanism which would always be there at your beck and call.

Two years ago, things changed. Some programmer started to set up their blog on Github. You have absolute administration authority of your own page. In the meantime, you can enjoy the convenience provided by GitHub — you can publish an article as long as you initial a commit with GitHub, whenever and wherever. Besides, it’s totally free. GitHub has no rate limitation. Basically you have a server with no entry limitation and this server has the willing to manage your webpage.

However, not all people who love to blog have a programming background. What we designed is a blog generator that allows you to generate a brand new blog only by clicking on several buttons.

**2. Advantages of our blog system**

* It is more effective. Saves more disks space: only part of the content need to be saved in the sever.
* You will have a blog with a unique URL which you can share with your friends instead of a page on a public blog system with redundancy information and functionalities.
* You only focus on writing and no need to manage your blog, most importantly, it is free to use.
* Your blog looks special because you have different choices with respects to the blog theme.

**3. How the blog works?**

First, all the writing would be stored in a repository. Users uses Jekyll as a simple, blog aware, static site generator(details is discussed below). At the top of the repository, there’s a file contains metadata about the whole file. This data tells Jekyll what layout to give the file, what the page’s title should be, etc. Also, Jekyll gets the list of blog posts by parsing the files in the \_post directory. Each post’s filename contains the publishing date and slug (what shows up in the URL) that the final HTML file should have. Posts are handled in a special way by Jekyll. The date you specify in the filename is used to construct the URL in the generated site.

**What is Markdown**

Markdown is a plain text formatting syntax designed so that it can optionally be converted to HTML using a tool like Jekyll. It is popularly used as format for readme files, or for writing messages in online discussion forums.

In terms of standard, there is no clearly definition for Markdown, apart from the original writeup and implementation by John Gruber, which is considered to be abandonware, leading to fragmentation as different vendors write their own variants of the language to correct flaws or add missing features. In late 2012, a standardization effort was started, spurred in part by a blog post of Jeff Atwood. A community website now aims to "document various tools and resources available to document authors and developers, as well as implementers of the various markdown implementations". A tool (named Babelmark2) is also available to "compare the output of various implementations" to "promote discussion of how and whether certain vague aspects of the markdown spec should be clarified [1].

**Why Markdown?**

This blog applied Markdown to write content and separated the layout frame with the content. Because of the simple writing style of Markdown has provided, users are able to concentrate more on writing task instead of having to focus first on the manipulation of the layout, which is sometimes even more time-consuming than the writing task itself. With Markdown automatically generate such readable formatting syntax, writing process is perfectly simplified. And it could transfer the content so easily into HTML format without any HTML language requirement like tags or labels, aided by Jekyll.

**What is Jekyll**

Jekyll is a simple, blog-aware, static site generator. It takes a template directory containing raw text files in various formats, runs it through Markdown (or Textile) and Liquid converters, and spits out a complete, ready-to-publish static website suitable for serving with your favorite web server. Jekyll also happens to be the engine behind GitHub Pages, which means you can use Jekyll to host your project’s page, blog, or website from GitHub’s servers for free.[2]

**Why Jekyll**

As mentioned, Jekyll is a simple free Blog generation tool, similar to WordPress. However, as Jekyll generates only static pages which no database support (or at least direct database support) is offered. This feature allowed a much more user-friendly interface compared with WordPress, and with third-party services as Disqus, a comment and discussion support plug -in, a lot more supportive functions would be offered. Overall, Jekyll is a parsing engine bundled as a ruby gem used to build static websites from dynamic components such as templates, partials, liquid code, markdown, etc. Jekyll is known as “a simple, blog aware, static site generator”[3].

**4. Blog generator design process**

**User operation procedure design:**

Our Blog generator is a web application which provides a more simple way to generate a blog. Users generate a desirable blog with a unique URL by simply clicking on several buttons. After users setted up the environment according to the “blog generator install manual”, they may open the index.asp file and it may redirect you to the start page of the web application.

After you registered and logged in, you may specify your blog’s name and start to write your first blog. You can save your blog after you finished your first blog. Once all the procedures went through, you may go to the “configuration” page to change your blog name or choose a theme you like and even fill out a list of at most three friend links that you want to be presented on the main page of your blog. After all has been set, go back to the homepage and click on the “generate blog” button. A brand new blog is generated and you’ll receive a URL which is the link to your new blog(Specific demo can be find in Appendix).

**Back-end design**

Blog information store and retrieval:

* All users’ information are stored in MSSQL with unique userID, which is the best way to retrieve data.

Database design:

* MSSQL contains a database named as “bloggenerator”, which has two tables – [bloginfo] and [userinfo].
* For [bloginfo] is like below:

[blogid] [int] IDENTITY(1,1)

[uid] [int], foreign key connect to [userinfo] table

[blogtitle] [varchar](50), Article title

[bloglabel] [varchar](50), Article label

[briefintro] [varchar](50), Article introduction

[blogcontent] [text], Article content

[createdate] [varchar](20), Article create date

* For [userinfo] is like below:

[uid] [int] IDENTITY(1,1), foreign key connect to [bloginfo] talbe

[uemail] [varchar](50), User email

[upassword] [varchar](20), User password

[username] [varchar](50), User name

[style] [tinyint], Blog theme(style)

[blogname] [varchar](50), Blog name

[linkname1] [varchar](50), Name of the first friend link

[linkurl1] [varchar](50), Url of the first friend link

[linkname2] [varchar](50), Name of the second friend link

[linkurl2] [varchar](50), Url of the second friend link

[linkname3] [varchar](50), Name of the third friend link

[linkurl3] [varchar](50), Url of the third friend link

Blog generation:

* When user clicks “blog generate”, all information about this user’s blog(such as blog name, style, All articles) will be retrieved, according to which new blog page documents(html documents and markdown documents) will be created in a given folder. Lastly, calling Jekyll command to parse this folder so that a new blog website is going to be generated in a second.

**5. Timeline**

|  |  |  |
| --- | --- | --- |
| Phase | Research Activity | Target Dates |
| Phase I: | Came up with the idea, and approved by the members  Kick off meeting to discuss the project  Discuss the report proposal  Generate the proposal report and prepare for presentation | Sep 28  Oct 10  Oct 17  Oct 20 |
|  |  |  |
| Phase II: | Design the front-end.  Design the back end and the database.  Testing and debugging | Nov 9  Nov 25  Nov 27 |
|  |  |  |
| Phase III: | Prepare for the demo  Finish final report | Dec 1  Dec  8 |

**6. Responsibility**

Jun Yin:

The proposer of the idea of this project. Mainly contributed to algorithm design, background design and development of website, the setup of Jekyll and connection between Jekyll and the website. Also, taking responsibility as a team leader.

Yue Wang:

Mainly contributed to home page, log-in and register parts and the page of editing users’ information making the website more user-friendly. Assisted to design the operation procedure and database. Partly contributed to the proposal and PPT for the project proposal presentation.

Lee-yin Wang:

Mainly contributed to BlogName and BlogContent page. The major work in BlogContent page is to use HTML Editor controller, and to insert and update all data into database. Adjust the index.aspx, embellish the data list of all blog part. Set up the database and the table attribute. Integrate all the codes of this platform, testing and debugging it. Help team members to solve problem with C#.

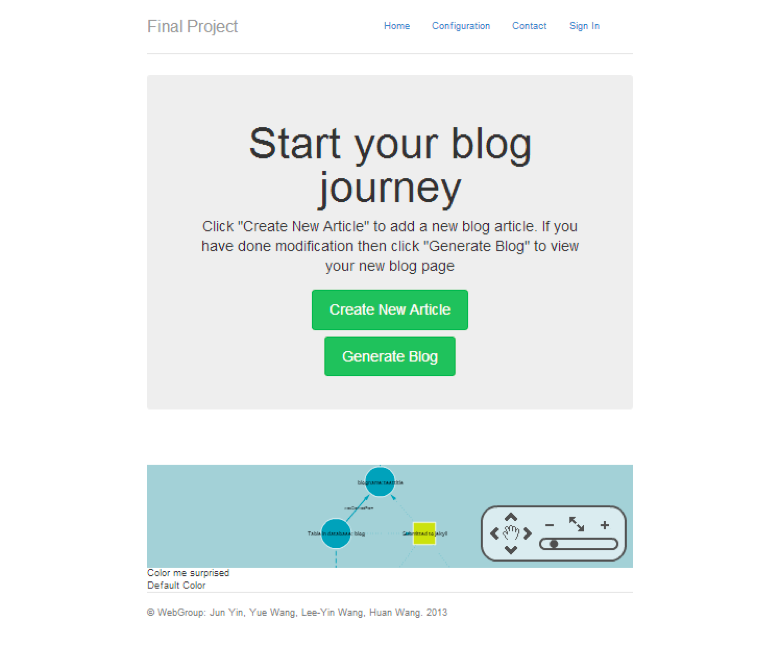
Huan Wang:

Mainly focused on front-end and user interface design. Partly designed the user operation procedure. Deployed HTML and CSS, also implemented Bootstrap. Finish documentation part.

**Appendix**

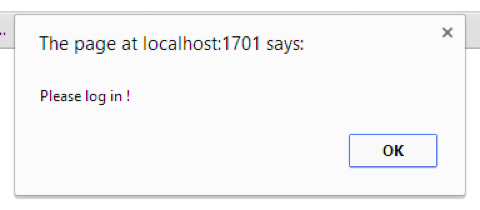
**Demo of our blog generator:**

Mr. Smith would like to use *Blog Generator* to create his own blog. Once Mr. Smith opens this web site, the homepage(Graph 1) comes in.

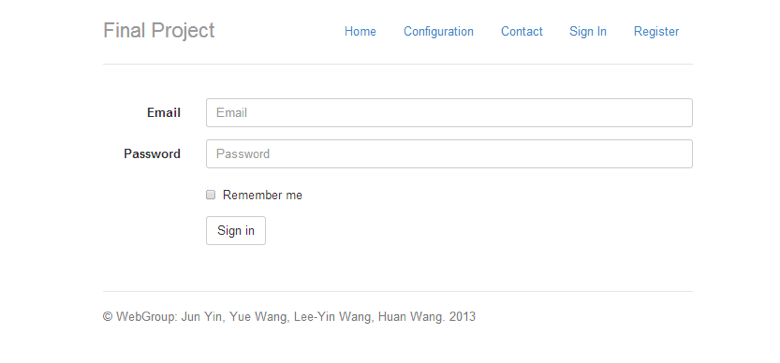


Graph 1. Homepage of *Blog Generator*

Mr. Smith is allowed to use the rest of the functions in the web site only if he has signed in. If Mr. Smith does not sign in and tries to create new articles or generate his blog, an alert will comes into the window(Graph 2). And then he will be directed to the *Sign In* page.



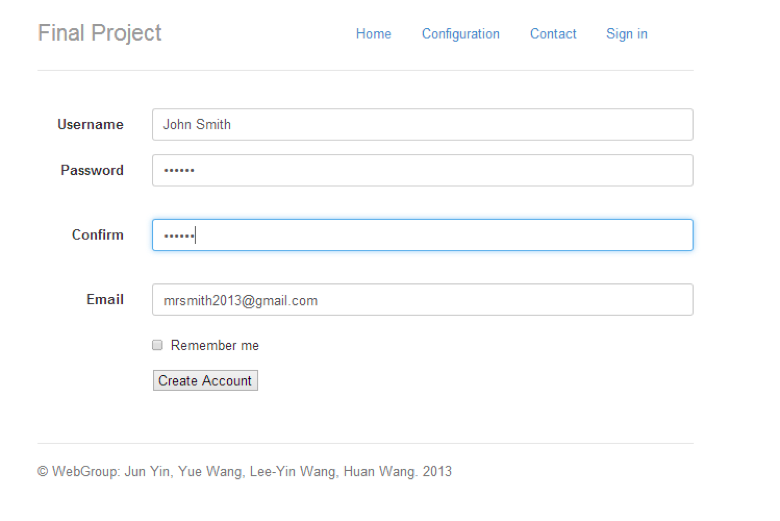
Graph 2. Login reminder pop up window



Graph 3. Sign in page

If Mr. Smith has account for our *Blog Generator*, he can *sign in*(Graph 3) with his account email and password.

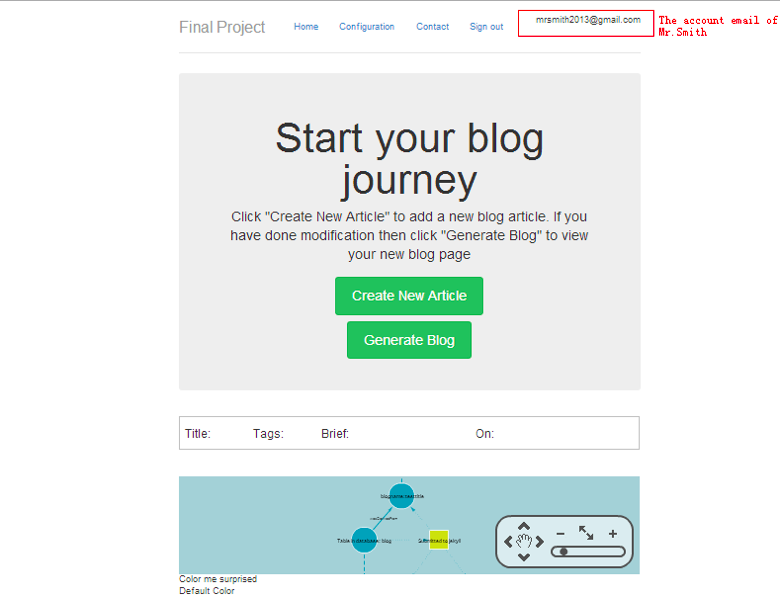
However, if Mr. Smith does not have any account for this web site, he has to go to register page (Graph 4) to open an account by his email and unique user name.



Graph 4. Registration page

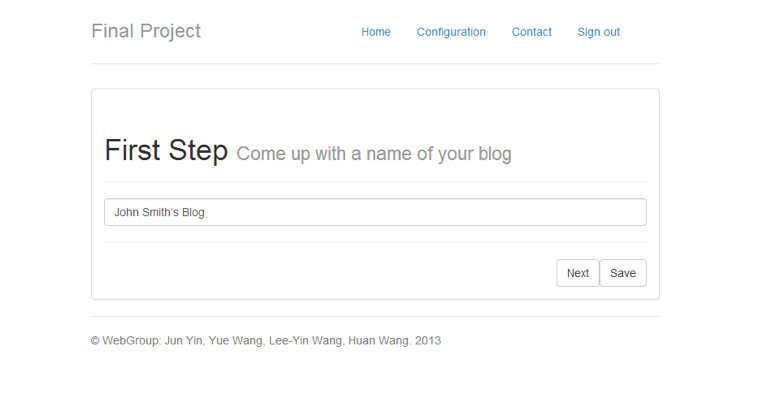
After he completes the registration steps and gets an account successfully, Mr. Smith will be directed to the homepage, and then he can start to generate his own blog.

If he signs out and then comes back to the website, Mr. Smith has go to sign in again before using other functions. After signing in, Mr. Smith is directed to the homepage and his email address is displayed on the right of the navigation bar(Graph 5), which means he signs in successfully.



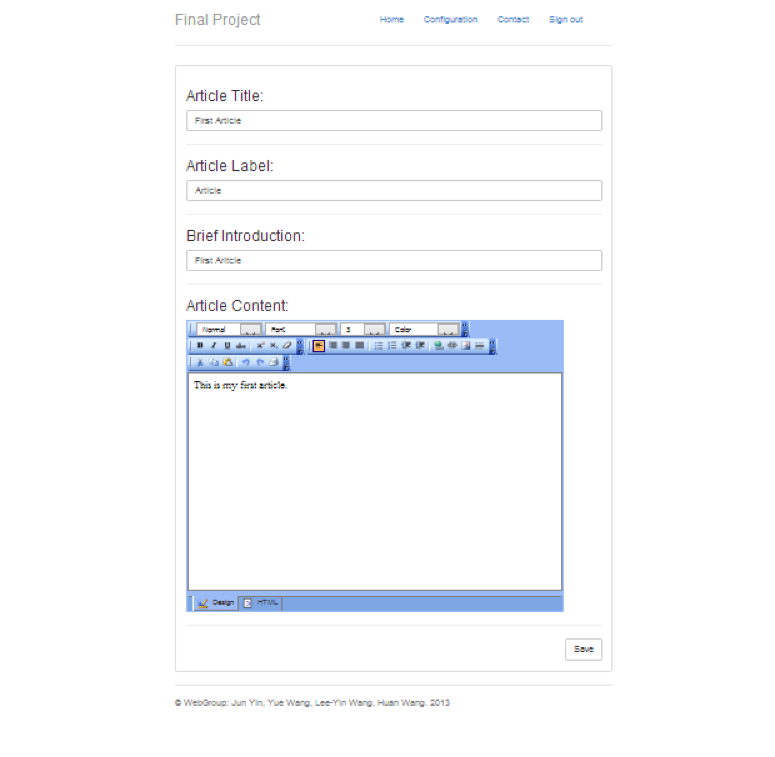
Graph 5. After-sign-in Homepage

Till now, Mr. Smith has no blog yet at this website. He can start to create his own blog. If it is the first time to create blog, he should enter his blog name(Graph 6).



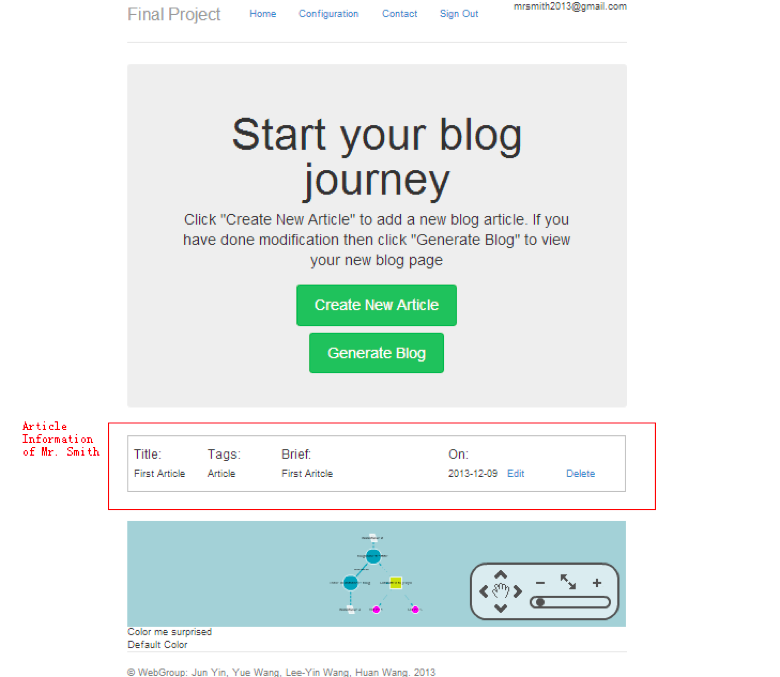
Graph 6. Blog Name Page

He can save his blog name by click *Save*, or he can immediately create a new article by clicking *Next*, and the blog name will be saved in the database and *Blog Content page* (Graph 7) will be displayed.



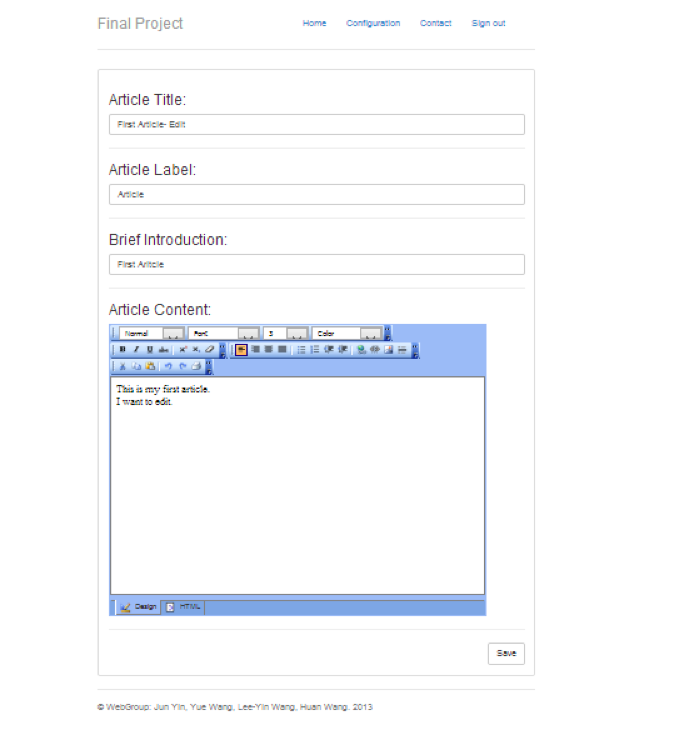
Graph 7. Blog Content Page

Mr. Smith can enter his article information and content and then click *Save.* And then his article will be saved. He also can customize the style of his article content by select Font size, Font style, and etc. After he saved his blog, he will be directed back to *Home* page. His article information will be displayed in the *Home* page (Graph 8).

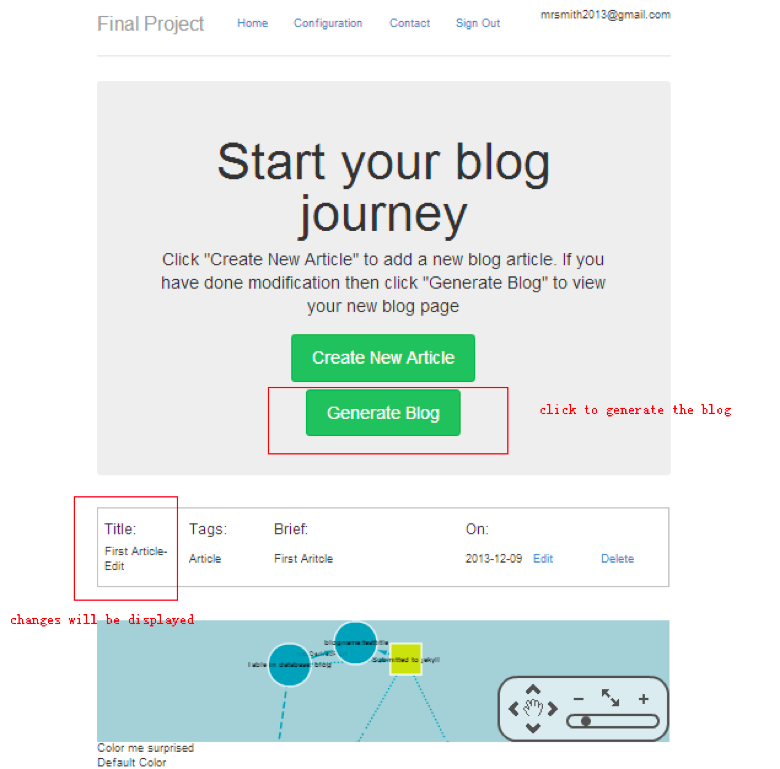


Graph 8. Article information shows on the Home page

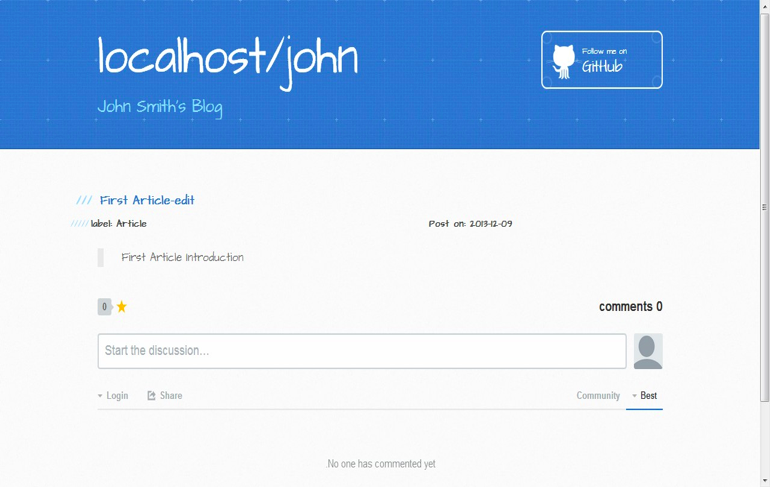
If he would like to change his article information or content, Mr. Smith can click *Edit* link to enter the Edit Page(Graph 9). Also, he can click *Delete* to delete a certain article.

Graph 9. Edit Page

After He modified the article, he can save it again, and all the changes will be saved.

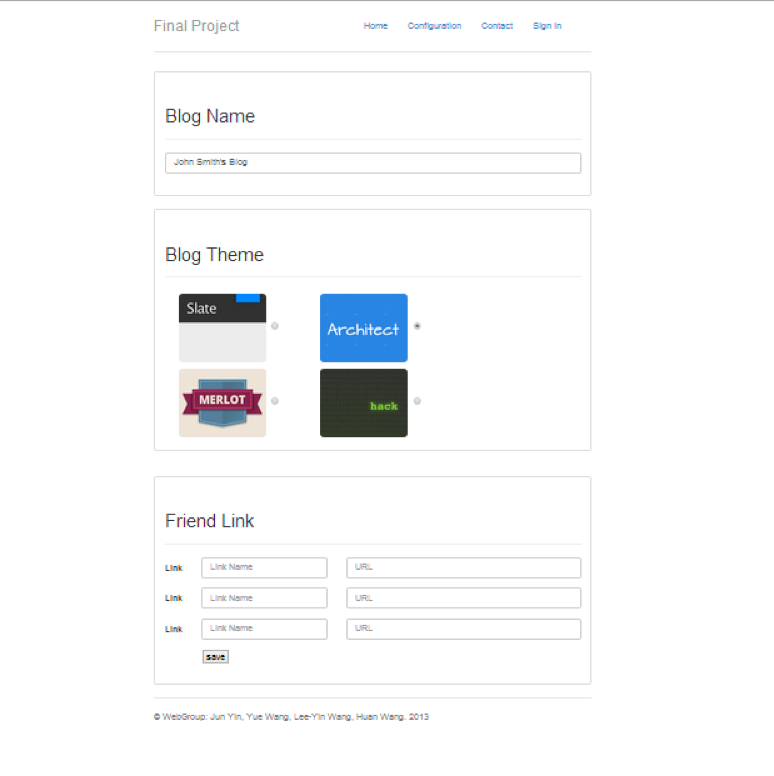
Finally, Mr. Smith can click *Generate Blog* button (Graph 10) to generate his own blog (Graph 11).

Graph 10. Generate Blog

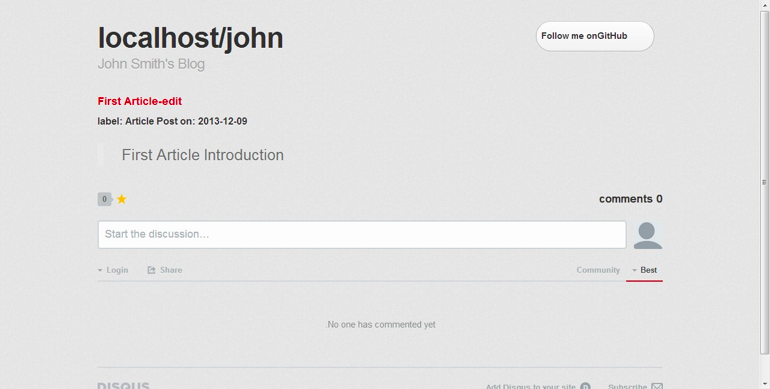


Graph 11. Blog generated by generator

If Mr. Smith would like to change his blog theme style or his blog name. He can click *Configuration* link to enter the *Configuration* page(Graph 12). Blog Generator provides four kinds of style for users to select as their blog interface theme. Besides, Mr. Smith can add his friend link information and the link will be added in the blog interface. For example, he can add his Facebook link within his blog interface(Graph 13).

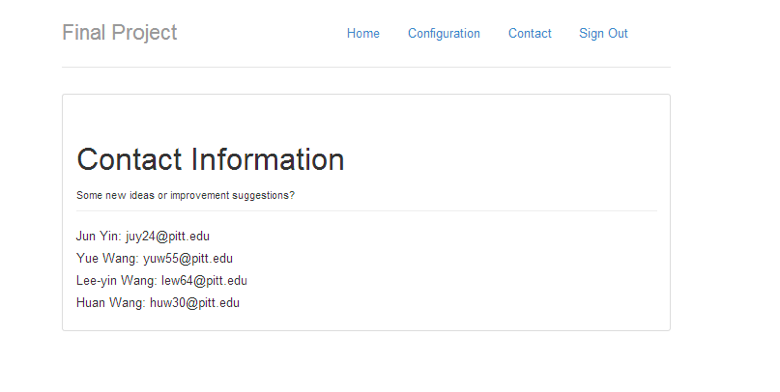


Graph 12. Configuration Page



Graph 13. Changed blog style

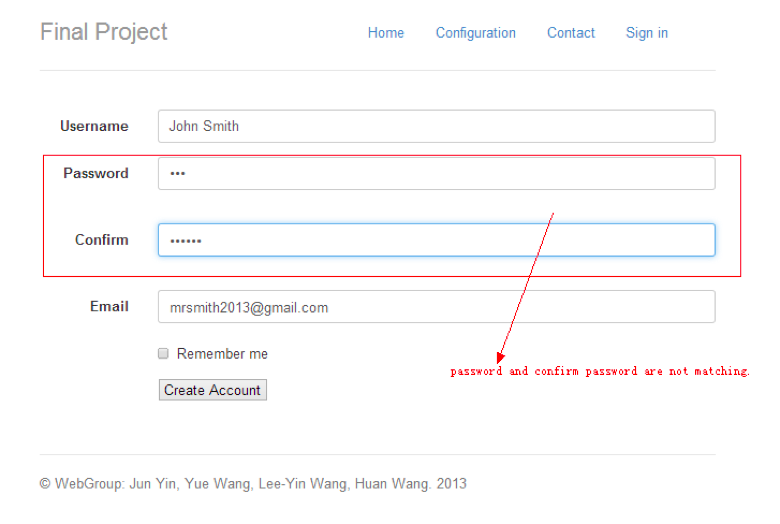
Mr. Smith can contact us(Graph 14) if he has any questions about *Blog Generator*.



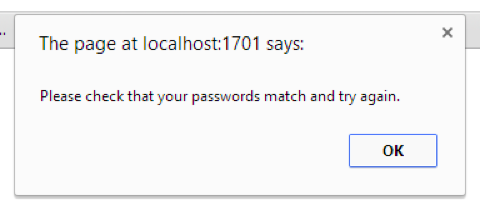
Graph 14. Contact page

## Validation

Password and confirm password must be matching when registering. Otherwise, an alert will be come out(Graph 15, 16).

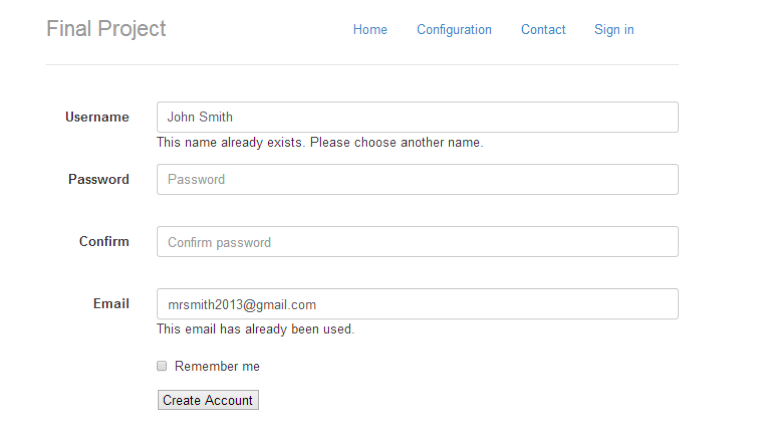


Graph 15. Password and confirm password are not matching

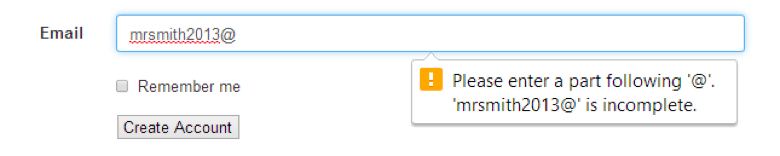


Graph 16. Check-password alert

If Mr. Smith has already applied *John Smith* for the user name and *mrsmith2013@gmail.com* for the account email, *John Smith* and *mrsmith2013@gmail.com* cannot be used again for registration purpose within this website(Graph 17).



Graph 17. Errors in registration steps



Graph 18. Email Validation

**References:**

1. Wikipedia markdown page: <https://en.wikipedia.org/wiki/Markdown>.

2. Jekyll: [http://jekyllrb.com/docs/home](http://jekyllrb.com/docs/home/)/.

3. Blogging like a hacker: <http://tom.preston-werner.com/2008/11/17/blogging-like-a-hacker.html>.