**Student Name:** Denise ‘Orah’ Kittrell

**Project Name:** DevNet

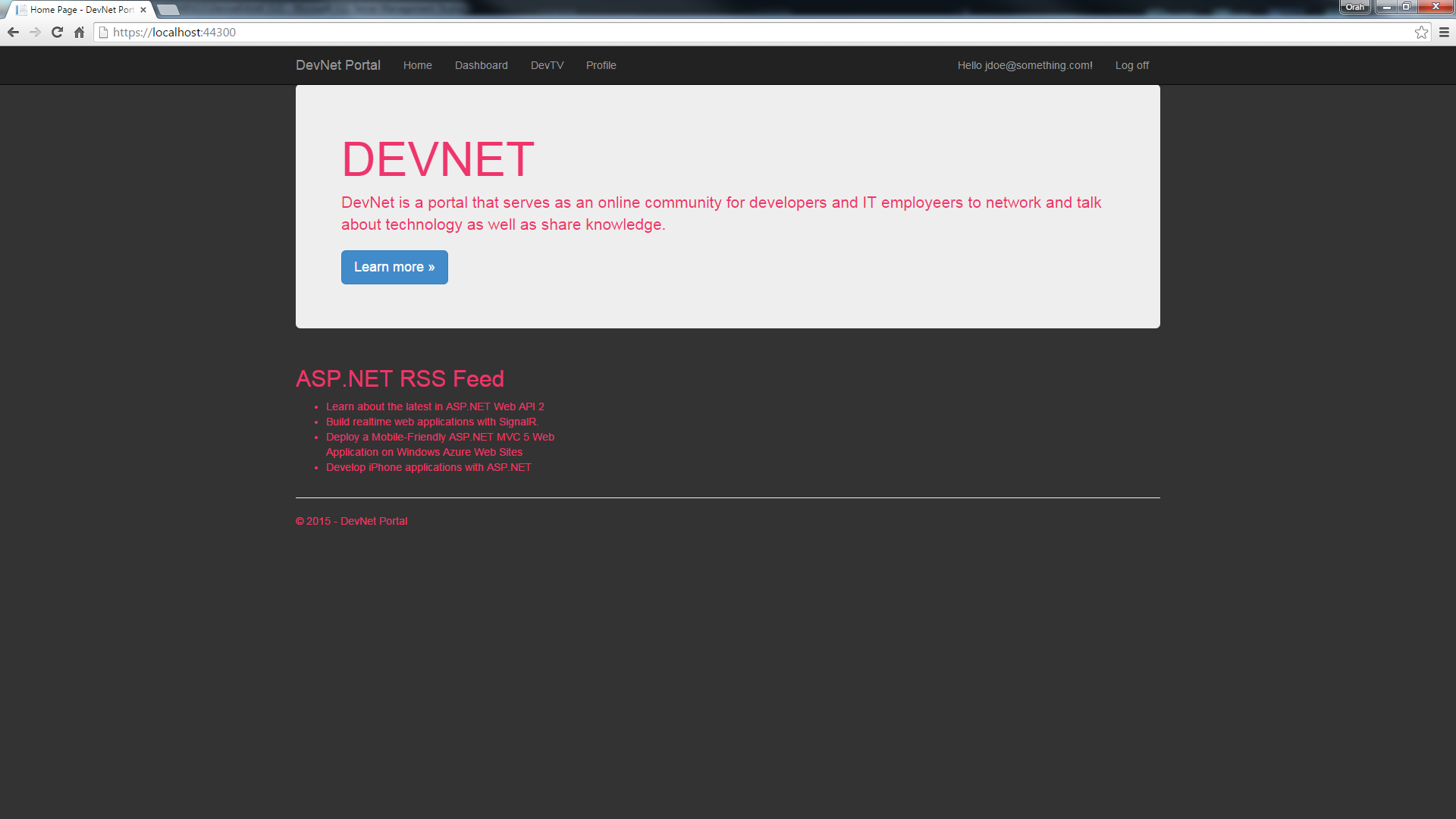
**Date:** 03-23-2015

**CPDM Capstone Final Project Prototype**

The Prototype for DevNet is meant as a visual aid for interested parties, sponsors, business investors, and clients. This helps ensure a mutual understanding of the site’s processes, functionality, and general look and feel before either party invests time and money into creating DevNet. The idea here is an opportunity for potential users of DevNet to provide invaluable feedback to the developer while the state of the project is still very pliable.

I will be providing a brief description of the screens, especially relating them back to the processes outlined in the requirements documentation. All screens are meant to approximate the general look and feel for the site, however the final result will be more refined and carefully crafted. As such these prototype screens are a crude representation of the final result but hopefully provide enough detail to give a general idea of the layout and functionality of the site. The order of the screens will be discussed in three main areas; the home page, screens pertaining to authentication or logging in to the site, and finally the specialized screens accessed via the navigation menu.

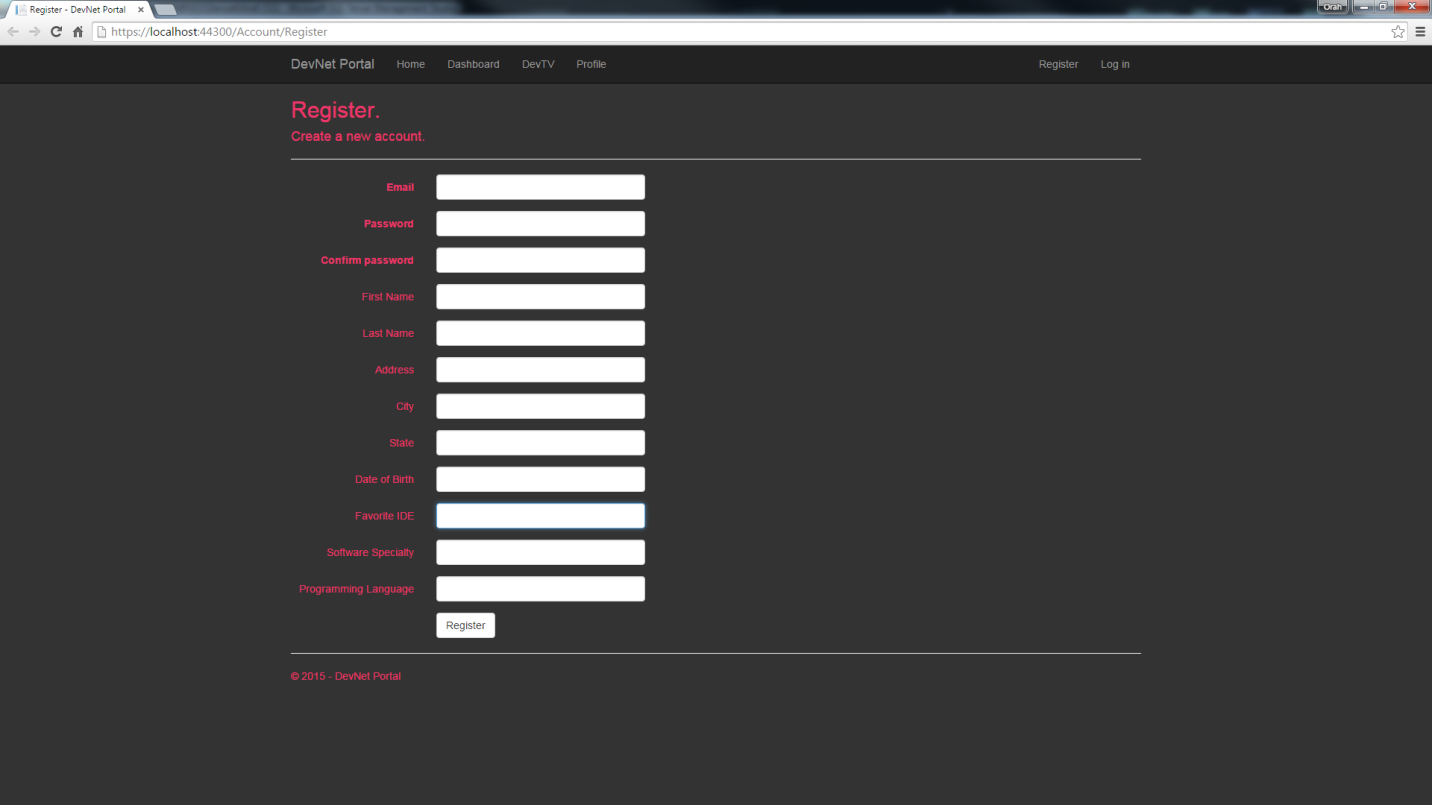
The home page(refer to Figure 1 below) is the first screen that a user sees regardless of whether they are registered with the site or not. This site will have a dark theme with a redish-magenta colored font. The choice of font here will most likely be different for the final site. There will be a top horizontal navigation bar. The navigation links that are left justified will take you to the specialized areas of the site. The navigation links that are right justified pertain to authentication. This navigation bar remains a fixed element on the site regardless of what screen one navigates to. The large white rectangular area is the banner area that prominently displays the DevNet logo. Here we have a rather plain text logo but this will be custom for the design of the final site. The logo is followed by a brief description of the site’s purpose. Optionally, a “learn more” button may be functional here to provide further details. The text beneath the white banner area are informative technical articles or links to technical sites provided by an RSS feed. This feed will only show up on the home page after a user logs into the site. The choice of RSS Feed is determined by the recommender system by using predictive analytics based on the user’s registration information.



**Figure 1: Home Page**

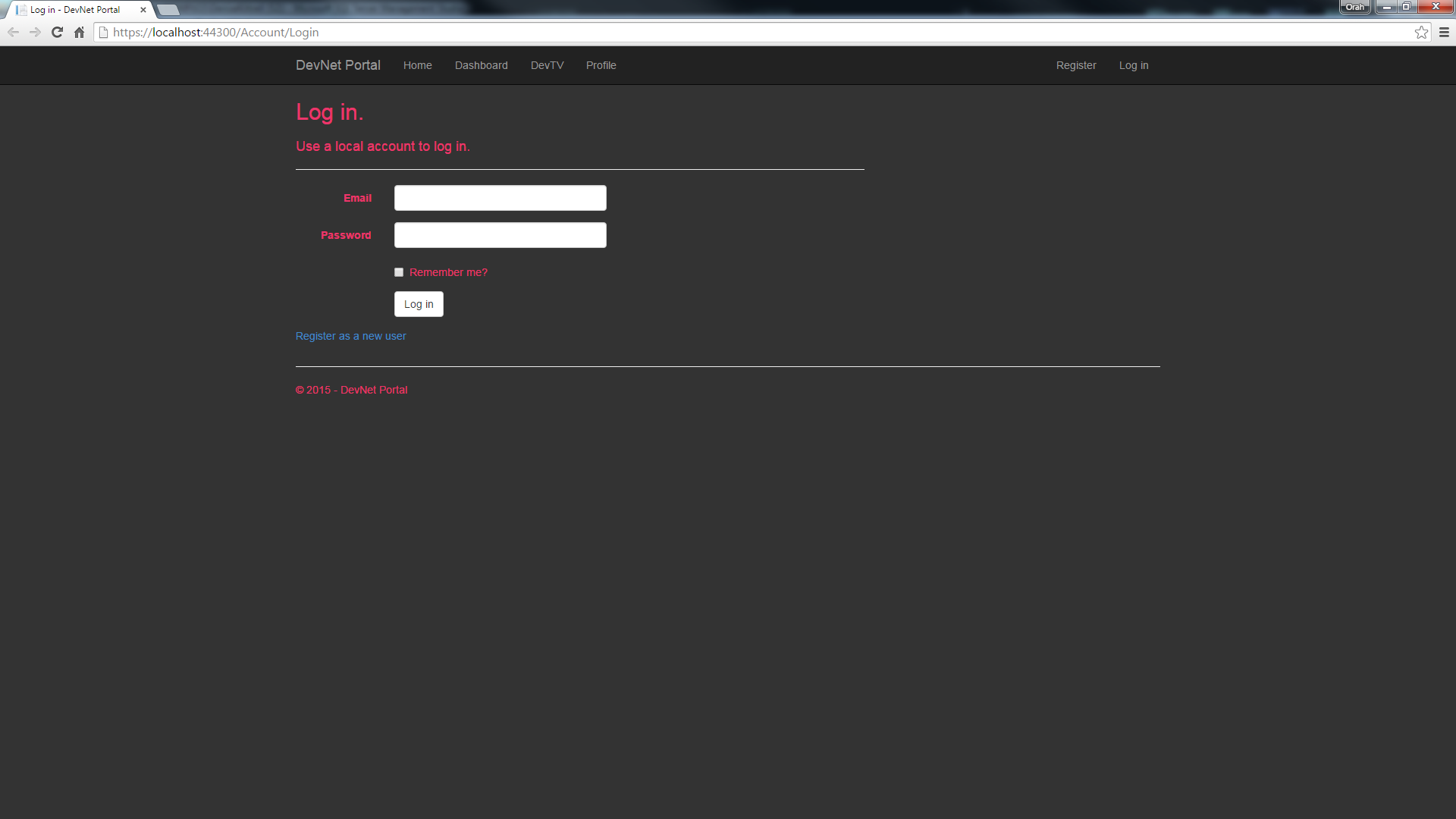
Now we shall move on to the second major area of the site, the authentication or login screens.

Returning to Figure 1 briefly, it shows the appearance of the navigation bar when a registered member is logged in as can be ascertained by the display of the user’s name as well as the “log off” link. On the other hand, if a user hasn’t yet logged into the site in its place there will be a “register” and “login” link displayed. A new user must click on the register link before they can get full access to the site. The only areas a non-member is authorized to access are the home page(of course), the dashboard, the login screen, and of course the registration screen. Figure 2 below shows the registration screen.



**Figure 2: Registration Screen**

This form must be completely filled out for a successful registration. Relevant validation will ensure the integrity of the user entered data. Many of these will be drop downs providing options for the user to make a selection. Namely, the State, Favorite IDE, Software Specialty, and Programming Language will be presented as options. The information gathered in the above fields will be used to determine the most likely RSS feed for the user. Once a successful registration is made after clicking the “Register” button, a web service is called. This web service call is made asynchronously and some of the form parameters are passed into the web service call. The service then returns an output parameter that is used to determine which RSS feed to recommend. This all occurs immediately after the registration on the server side. If all goes well, the user will be redirected to the home page with a prompt asking them if they accept the recommended RSS feed. If they accept, they will now see this RSS feed on their home screen every time they login to the site. An additional option would be allowing the user to manually select an RSS feed if they don’t like the one provided by the recommender system. Additionally, it could be an option to change this in the account management screen but these are extra features and may be out of scope for the time constraints of this project. Below shows the login screen for returning members to gain access to the site.

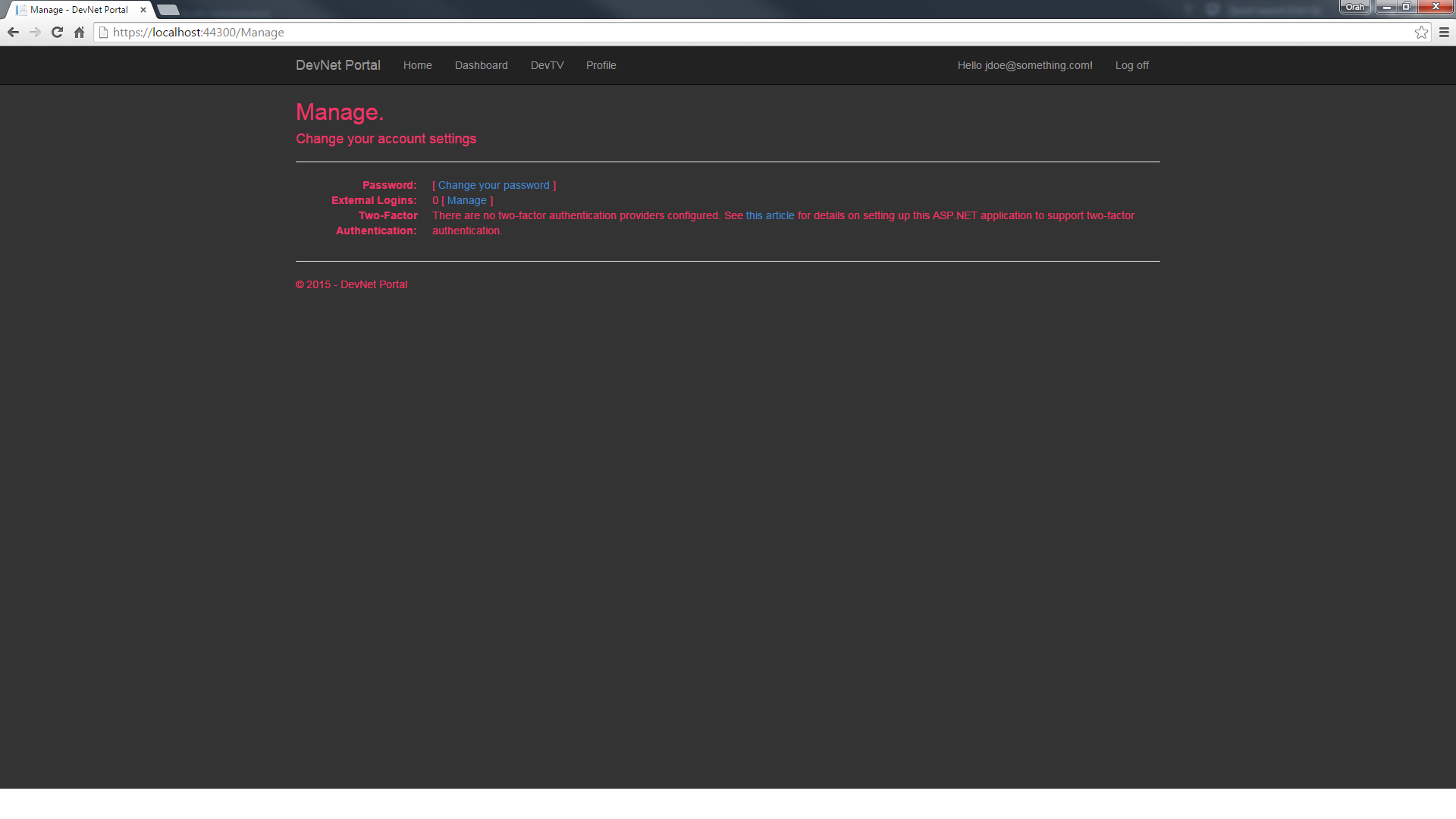


**Figure 3: Login Screen**

Upon examination of the login screen above, we can see that there is an option to remember the user’s login information for their convenience, additionally there is a link to the registration page incase new users came here first. Now, if this workaholic developer has time to get to it, there may be social networking authentication using OAuth2 sign in. Well that’s what coffee is for. What this means is the user could use their Google, Facebook, LinkedIn sign-ons instead. This is merely a means of making the user’s experience more convenient. This is why in Figure 3 above it reads, “local account”. The local account here is the once tracked by DevNet.

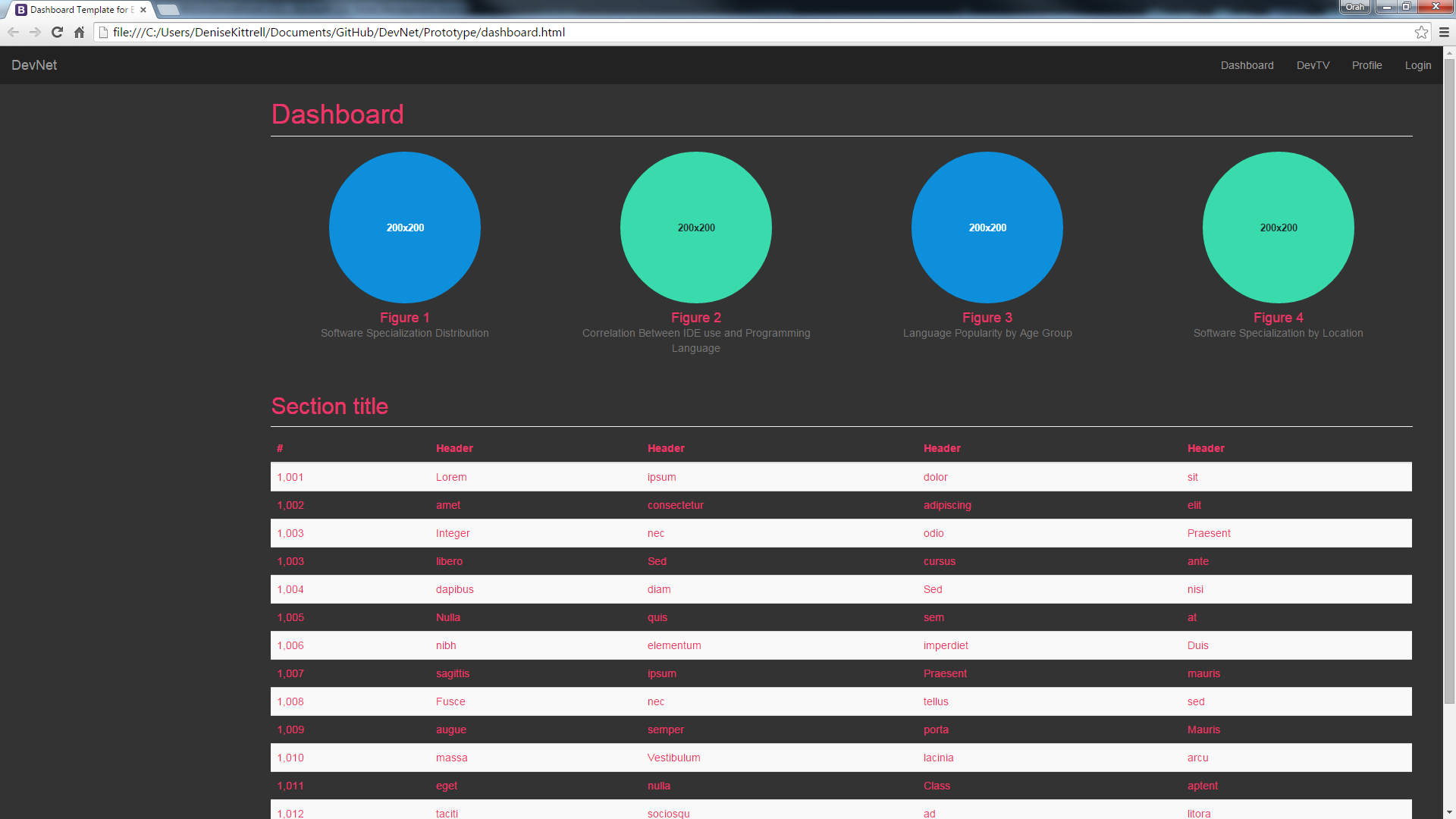
When a user is logged in their name is displayed and it will also be a clickable link.

This link takes the user to a management screen as shown in figure 4 below.

**Figure 4: Management Screen**

Here the user can change their password as well as manage their external(social networking) logins if applicable. This is also the place where the user could change what RSS feed is displayed on their home page upon login. This feature along with the external login feature are optional for the scope of this project but would be nice to have if time permits. However, the ability for the user to change their password will be operational for this project. The screens for the “change password screen”, “manage external login screen”, and “manage RSS feed screen” are not shown in this document for the sake of brevity especially since the latter two screens may not be implemented anyway.

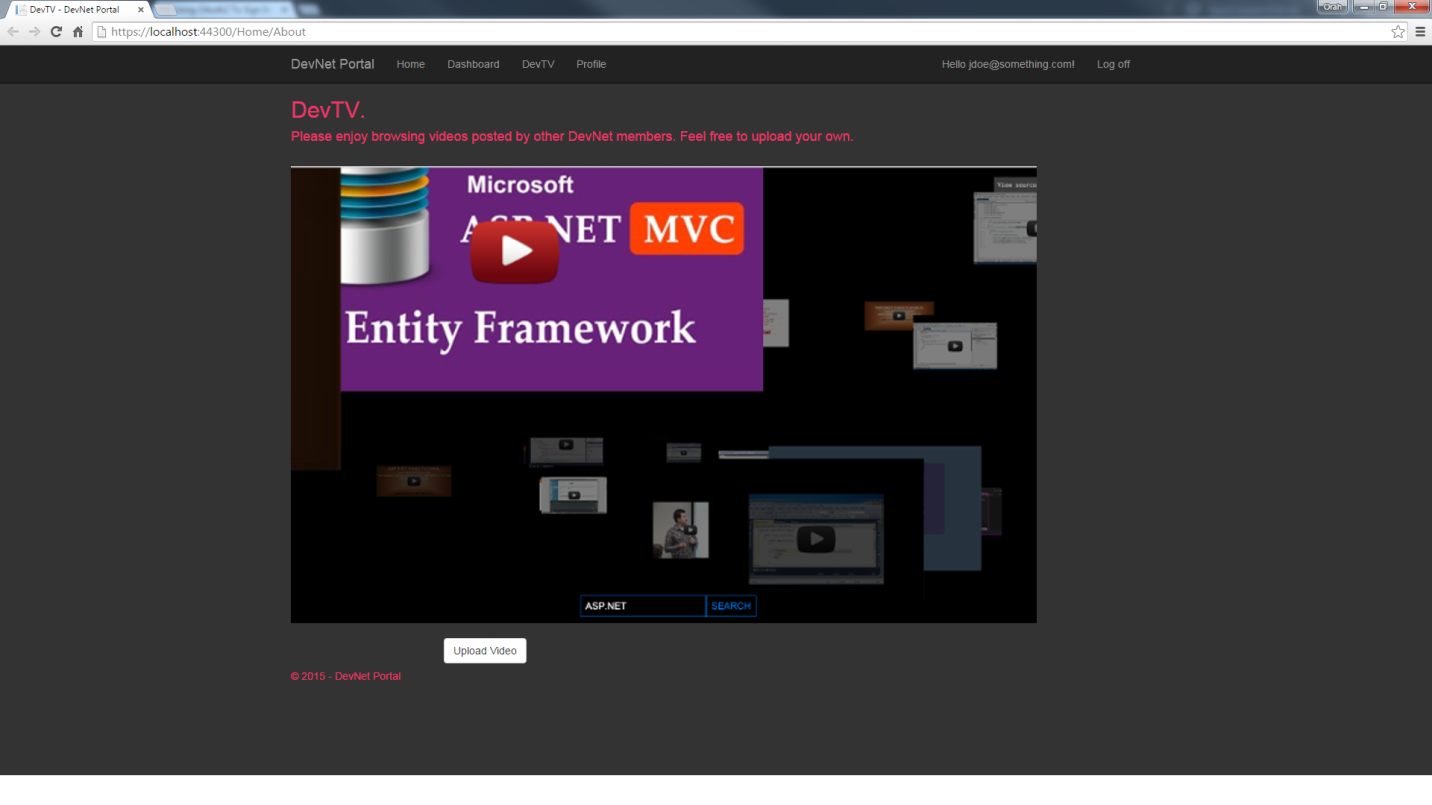
At this point I will discuss the third area of DevNet, the specialization screens. The first specialization screen is the dashboard screen. The purpose of the dashboard is to display statistics calculated from a model applying machine learning on the target data set, whether it is a linear regression algorithm or a clustering algorithm. In Figure 5 below, we can see a rather crude dashboard filled with dummy data and lame colored circles. Hopefully this figure conveys the intent and concept of the dashboard despite its lack of detail.



**Figure 5: Dashboard Screen**

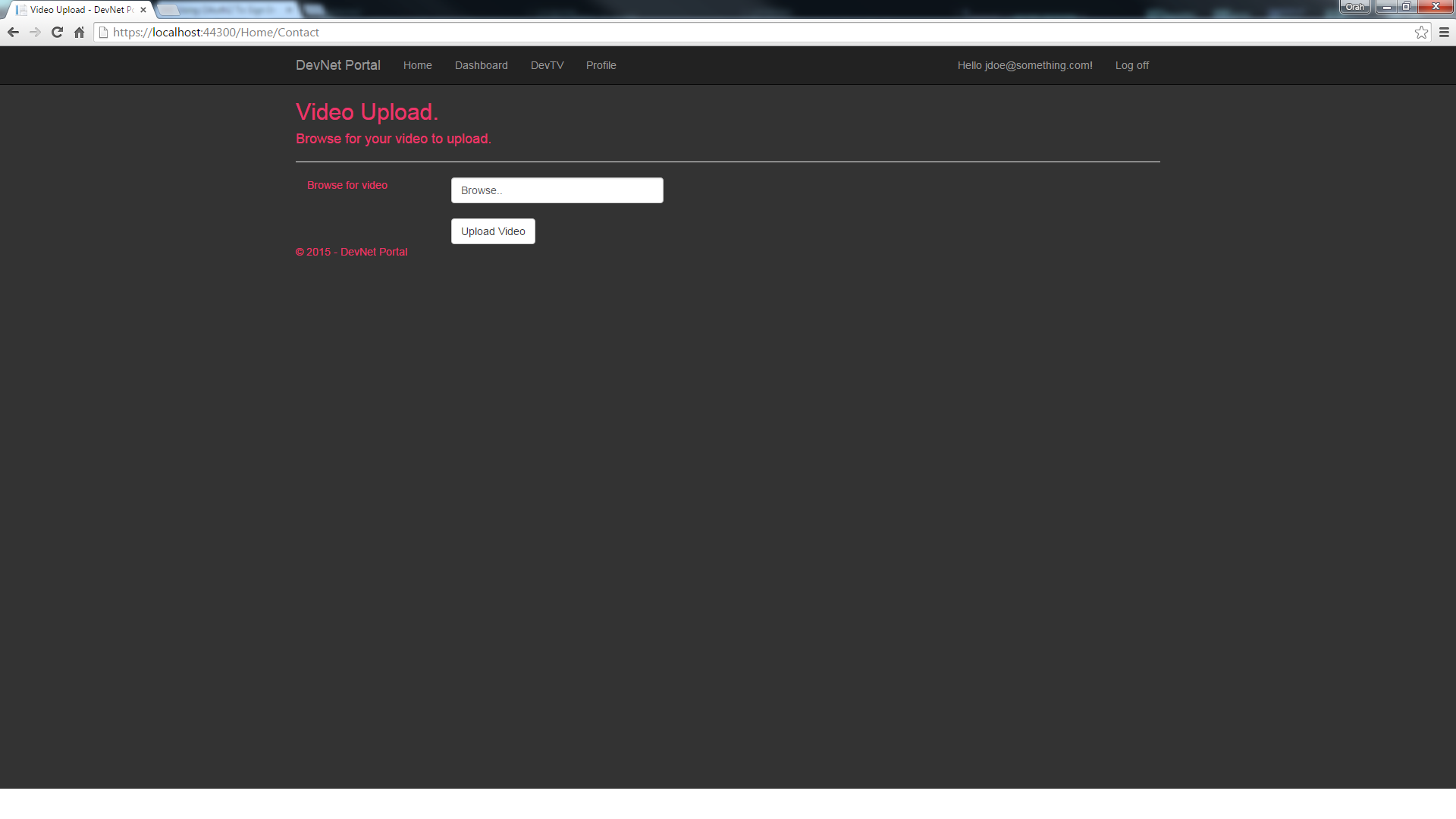
The data set will usually be historical user data gleamed from the site over time. Since this site is new, I have to artificially construct this data at the outset to have a starting point. This is a bit contrived and reverse engineered but for the scope of this project it is the best that can be done. It is possible to test the model’s predictive accuracy. The model can be manually retrained using Azure Machine Learning, which is a cloud based service offered by Microsoft Azure. This DevNet site itself will be hosted in the cloud as an Azure site as well as its database. The data returned from the web service(s) will be graphically displayed using d3.js, a JavaScript library for displaying data. Below these graph(s) will be a flat table of the raw data used in the machine learning model. The main goal of the dashboard will be to provide information that will hopefully be both useful to developer and technology employer alike. The idea is they can make more informed career and or hiring decisions. This is why this screen is accessible to even non-members since they may not want to partake in the developer community, yet they seek information regarding developers, especially locally.

Now machine learning is very interesting topic on its own but now for something really cool. The next specialty screen is the DevTV screen. DevTV is well, TV for developers. Basically, it is a gallery of videos uploaded by members. I have no idea if the content will stay on germane to the site. Knowing how strange and bizarre developers are there will be lots of reddit worthy content and I am sure nyan cat will make an appearance☺. In any event, there will be no pre-screening of uploaded videos. Anything goes, but the site administrator/moderator can always delete the member if things get out of hand! The content will hopefully be related to programming and consist of tutorials or demos. These videos will be hosted in the cloud using Azure’s Media Services. Here comes the cool part, the videos will be floating through a three dimensional environment! It gets even better, the user can navigate in the environment to get to the videos and the play the selected one! As can be seen from Figure 6 below, there will most likely be a search filter to limit the videos.

**Figure 6: DevTV Screen**

Notice the YouTube video player, this is just filler for now. Again, I am starting this site from scratch so I will most likely fill up the gallery with programming related YouTube videos initially.

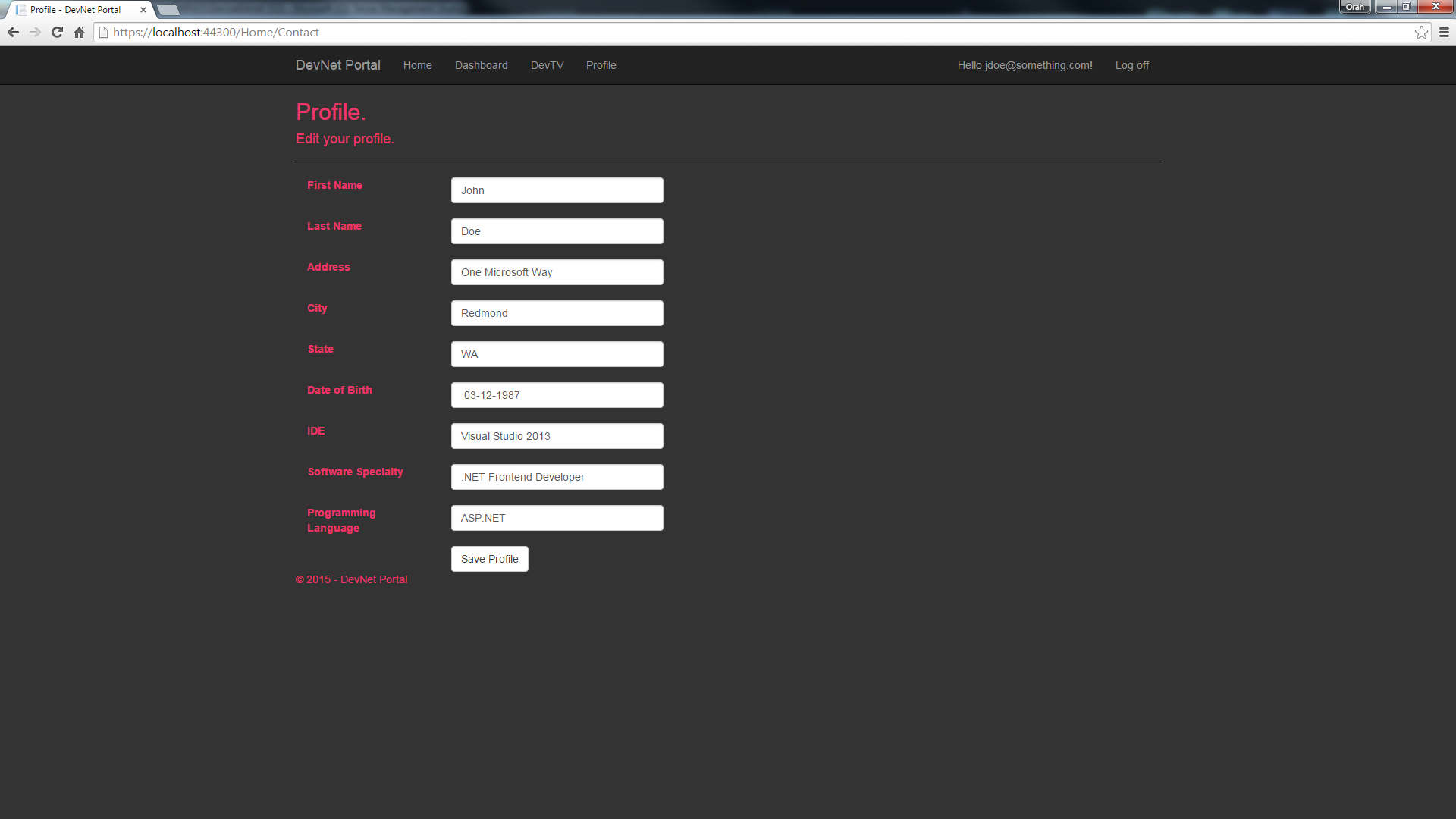
A member of DevNet may if they so choose upload their own video to the gallery for other members to view. This leads us to the video upload screen as shown in Figure 7 below.

**Figure 7: Video Upload Screen**

Again, this screen is pretty simple looking but don’t be deceived, there is a lot going on behind the scenes as background processes. The user may browse their local PC for a video to upload. By clicking the “Upload Video” button they have exposed their video to being ridiculed by other developers on DevNet. Well, this won’t be YouTube so there won’t be an area for trolls to leave comments but still. Just be certain you really want the video to be public. This screen tour is already getting too long so I won’t reiterate the video upload and streaming process here. This was already mentioned in the requirements document.

Finally, we will discuss the Profile Screen. Here a DevNet member can edit their profile.

It will look a lot like the registration screen, not too exciting but here it is.



**Figure 8: Profile Screen**

This could get a lot more exciting and quite possibly rival Facebook! Yeah I keep telling myself that anyway. Wouldn’t it be nice to see what other developers look like!?!? I might even include a dating area in the future as another feature or is that a bug? Well maybe it is a good thing I only have so much time to implement features for DevNet ;o.

Well that concludes the tour of the DevNet prototype. Now it is time to get busy! I will soon understand what it is to be a code monkey. Let’s see 40+ hours of coding at work, at least ten hours collaborating on a video game project with another developer(well I was asked and couldn’t refuse), and of course DevNet! Yes I really do dream in code.