*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

User Manual

LegalWise

**Team Members Product Owner**: Jaime Borras

Amanda Chiu **Instructor**: Masoud Sadjadi

Valeria Lopez



*Copyright and trademark notices, restrictions on copying or distributing the documentation, information for contacting the issuing organization (reader’s comments), warranties, contractual obligations or disclaimers, and general warnings and cautions.*

# User Guide

## Introduction

The purpose of the LegalWise web application is to answer question on the legal domain. The application has a user interface that allows the user to interact with question and answer system running in the background. The question and answer system is based on the Solr server and counts with frameworks such as Lucene to improve word matching and indexing. The server also stores the legal files from where the questions are answer. The idea of the LegalWise application was to allow legal professionals to find related cases or answer questions regarding a specific domain an easy task.

## Hardware and Software Requirements

In order to run the LegalWise application we need the following requirements:

1. VM running a Linux distribution
2. Java 1.7 or higher
3. Maven 3.0 or higher
4. Node.js version v04 or higher

The application is divided in a front end and a back end. The front end application has the user interface which runs on Node.js. Currently the front end is hosted on Bluemix and the building process takes place in DevOps on the IBM Cloud. The back end application is the question and answer system and was built on Java. The application is currently hosted at a FIU VM with the requirements above.

## Getting Started

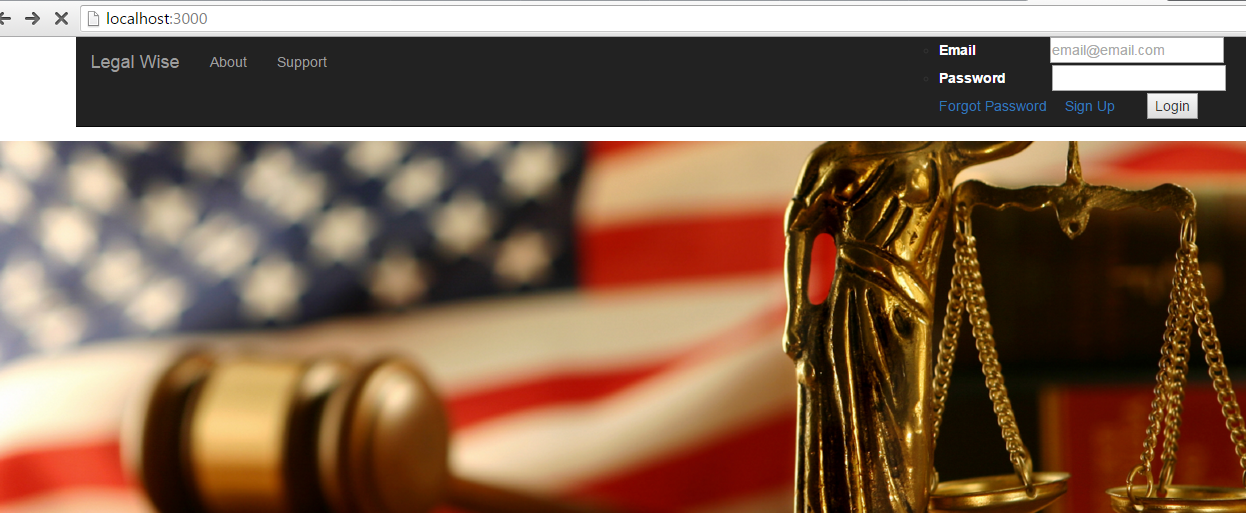
**How to run the application**

The front end application is currently running on the Bluemix Cloud at <http://legalwise.mybluemix.net/>

To run the application locally, Node.js must be installed on the local system. After you install the Node.js runtime environment locally, enter the following commands to start the application:



Then, by opening a browser window and navigating to http://localhost:3000, you can see the following screen:

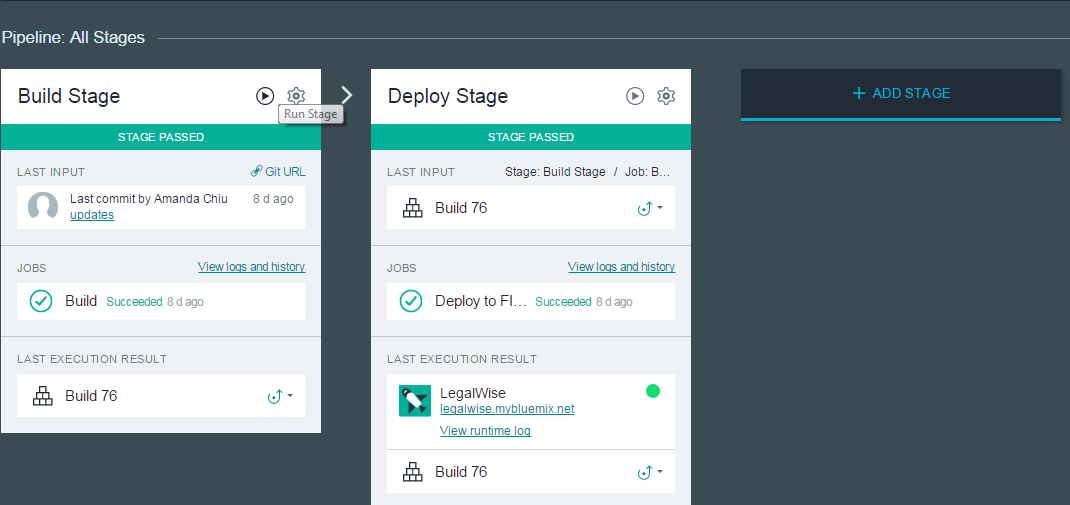


After you start the Node.js application locally, you can make changes to the application code locally. You can then push your changes to DevOps and built the application on Bluemix.

In order to know if the application build properly in Bluemix, you can navigate to the DevOps services make sure the application is running. To access the DevOps services login with your IBM account into Bluemix and click on the LegalWise application. Then click on the top right corner the GIT URL.

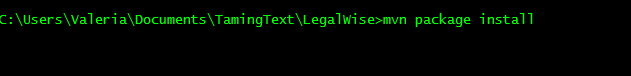


Once you are on GIT you will see on the top right corner a BUILT & DEPLOY button that will take you to the DevOps Services. The screen show below shows the main features of the DevOps Services website. On the Built Stage you make sure the application was built with no errors. On the Deploy Stage the application is deploy on the IBM Cloud. The database credentials can be obtained by installing the cloud foundry on your system. For more information please search for cloud foundry on Bluemix.



The question and answer system is currently hosted in a VM at FIU. Download the source code or clone it using GIT. The code is currently at <https://github.com/FIU-SCIS-Senior-Projects/LegalWise>

To run the back end application make sure to have the software requirements install in your system. On the command line set up your JAVA\_HOME environment variable. Build the application using maven.

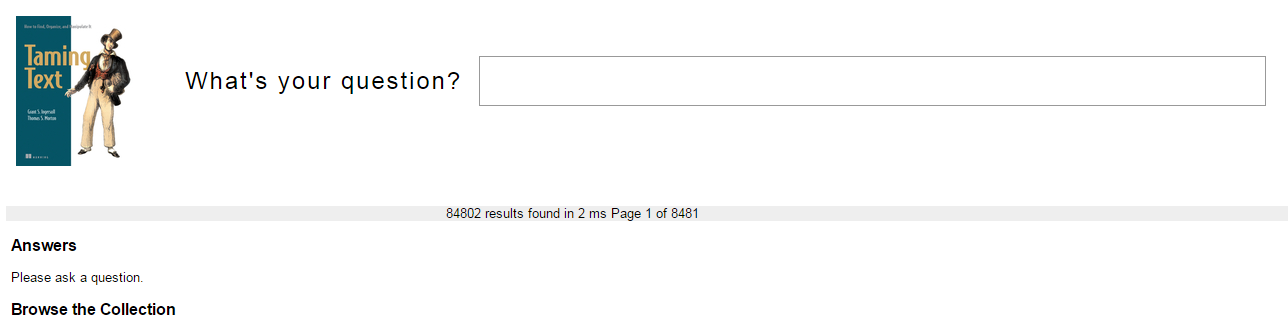


If the application build successfully you will be able to run the application locally by running the following commands:

1. Go to the LegalWise folder
2. cd bin
3. ./start solr solr-qa



The application will be running on localhost:8983/solr/answer/. To make changes to the source code use Eclipse as IDE and push the changes to GIT. The application should look something like this.



## How to Index the QA system with the Wikipedia

The following steps are described on the Taming Text book with more detail. The book is included as part of the documentation for this project.

QA systems that are built on top of search engines (most are), as you might imagine, require content in the search engine to act as the source for discovering answers, since the system doesn’t have some intrinsic knowledge of all questionsand answers. This requirement brings with it the complication that a QA system can only be as good as the content that it uses as its source. For instance, if you fed the engine documents from Europe written in the era before Christopher Columbus (surely they are all digitized, right?) and asked the system “What shape is theEarth?” it likely would answer with the equivalent of flat. For our system, we’ll use adump of the English Wikipedia taken on October 11, 2010 (the first 100K docs are cached at http://maven.tamingtext.com/freebase-wex-2011-01-18-articles-first100k.tsv.gz; 411 MB zipped.) Note that this file is large, but this is necessary, as we wish to demonstrate with real data. After it’s downloaded, unpack it using gunzip or a similar tool. If that file is too big or your want to try a smaller version first, you can download <http://maven.tamingtext.com/freebase-wex-2011-01-18-articles-first10k.tsv>, which consists of the first 10,000 articles of the larger file. This file isn’t compressed, so there’s no need to unpack it.After you have the data, you can index it into your system by running the following steps:

* Type cd QASystem
* Run indexWikipedia.sh --wikiFile <PATH TO WIKI FILE> (\*NIX) or index-

Wikipedia.cmd --wikiFile <PATH TO WIKI FILE> (Windows). This will take some time to complete. Use the --help option to see all available indexing options.