**C:\Users\abc\Desktop\bot_logo2.pngBOTMAN**

**SLACKBOT CREATION FRAMEWORK**

|  |  |
| --- | --- |
| Nachiket Joshi | 011408956 |
| Sagar Mane | 011419135 |
| Abhishek Madan | 011408969 |
| Saurabh Gedam | 011451674 |

**INTRODUCTION:**

Chatbots are on a rise. These are automated computer programs which respond to user queries by employing a suitable ML algorithm to gauge what the user is asking and reply appropriately. Bots have appeal and huge fan following not only among the developer community, but also among the non-developer tech community. People can now exploit the easy integration of their bots with the chat platforms like: ‘facebook’ and ‘slack’ to automate their mundane tasks, thereby saving time.

**PROJECT IDEA:**

In order to integrate a chat bot with a channel, there are a number of configuration steps involved- creating a web app project module, creating a cloud based repository to hold the source code, run the application on cloud so that the bot is active independent of the developer’s machine state and many more. Our idea is to provide a framework, which would assist a novice person with coding a bot, guide him through the process of setting up of the environment and deploy the bot on the cloud on a single button click. Thus by using our framework, which would be a hosted service, the user can focus more on the behavioral part of the bot and less on the configuration part.

**COMPETITIVE LANDSCAPE:**

There are only a few web applications which provides user with such a framework. Moreover these platforms are too strict with the bot behavior coding guidelines. They do not allow developers to include additional, rich packages of the underlying language. Looking at how a bot could be helpful for a person, a more liberal framework can definitely find its place in this competitive market.

**TECHNOLOGIES USED:**

**Front end-**

* HTML
* CSS
* Javascript
* Jquery
* Angular js

**Server side-**

* Node package manager
* Node js

**Platform for storage and deployment-**

* Github – It hold the code for the hosting the node application bot.
* Heroku – It is a cloud service for hosting the bot.

**ARCHITECTURE:**

5. Bot deployed on Heroku listen’s on Slack Channel and replies.

4. Push bot to cloud and run it to make it live

****

GIT



2. Create a Heroku directory for deploying a bot

1. Register a new bot on a channel and obtain accesses token

3. Upload bot source code Git

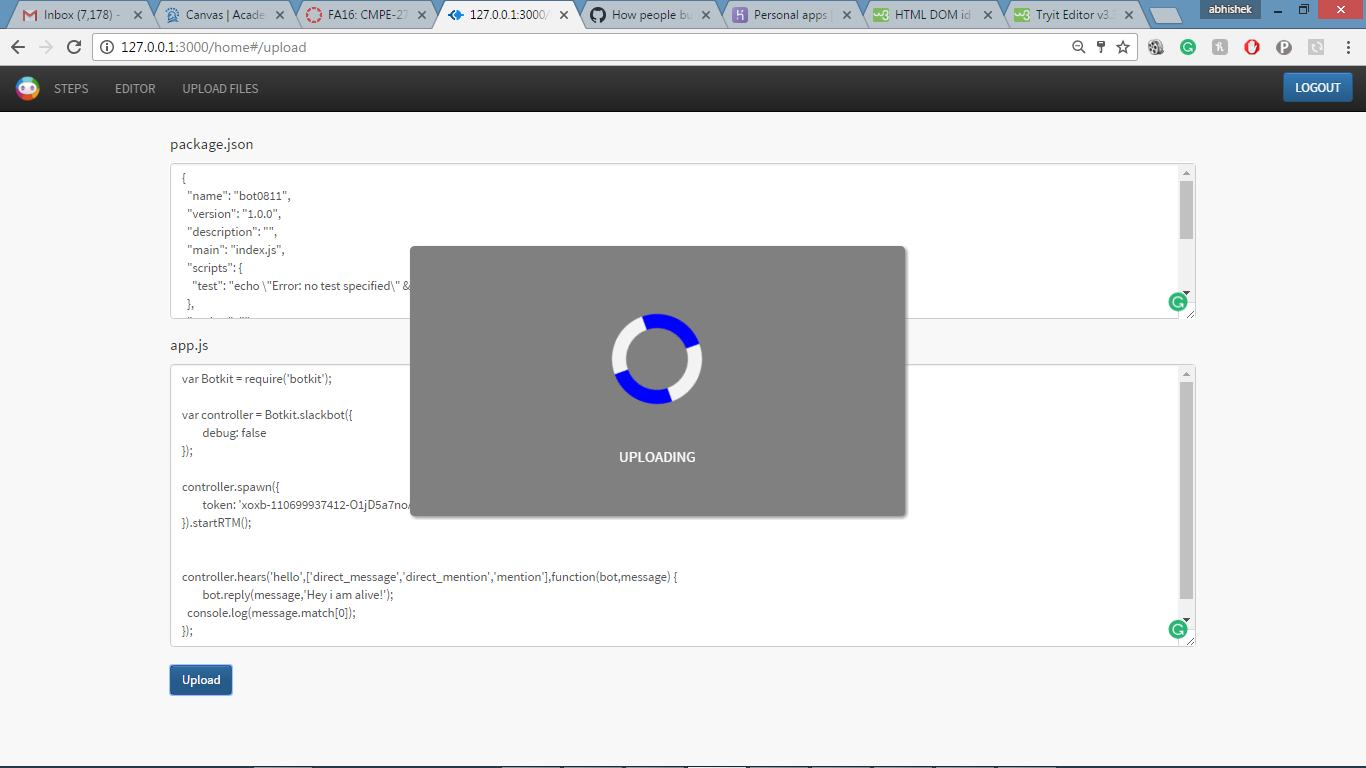
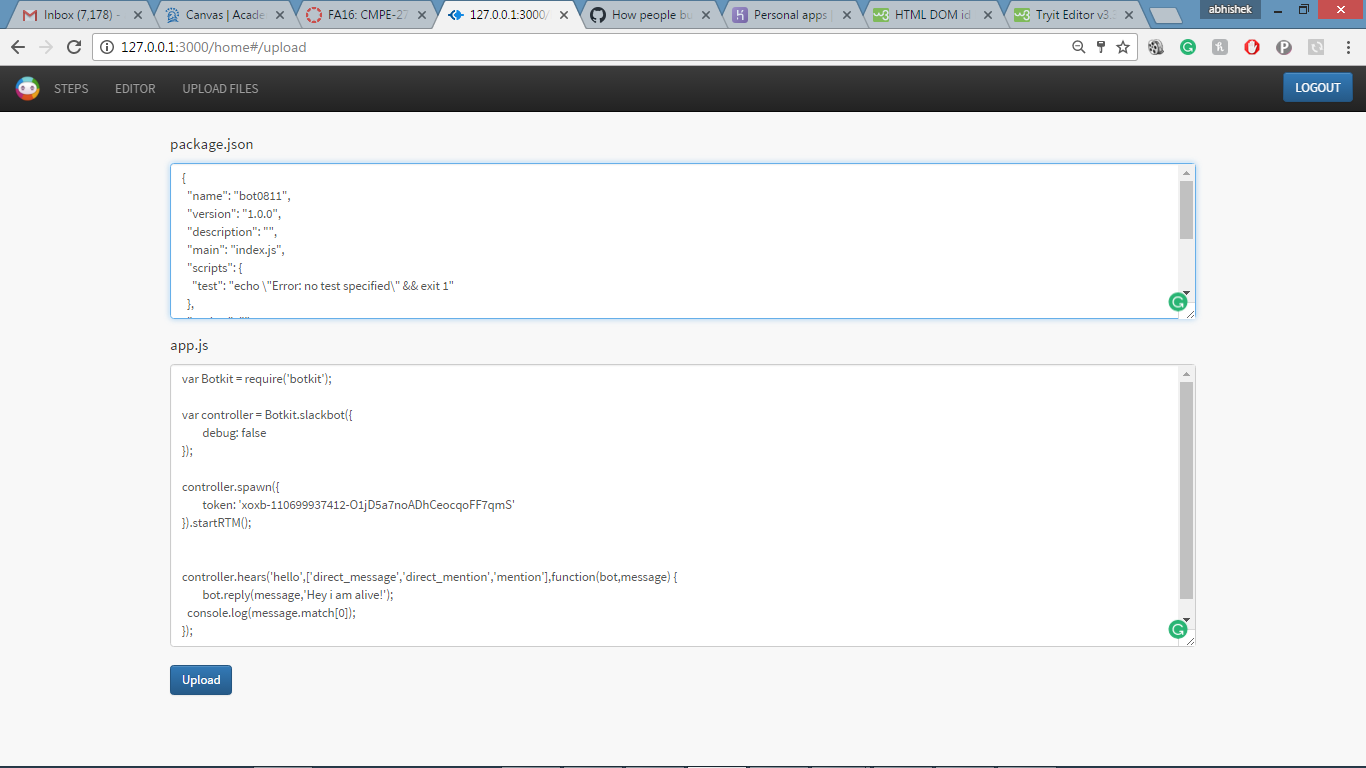
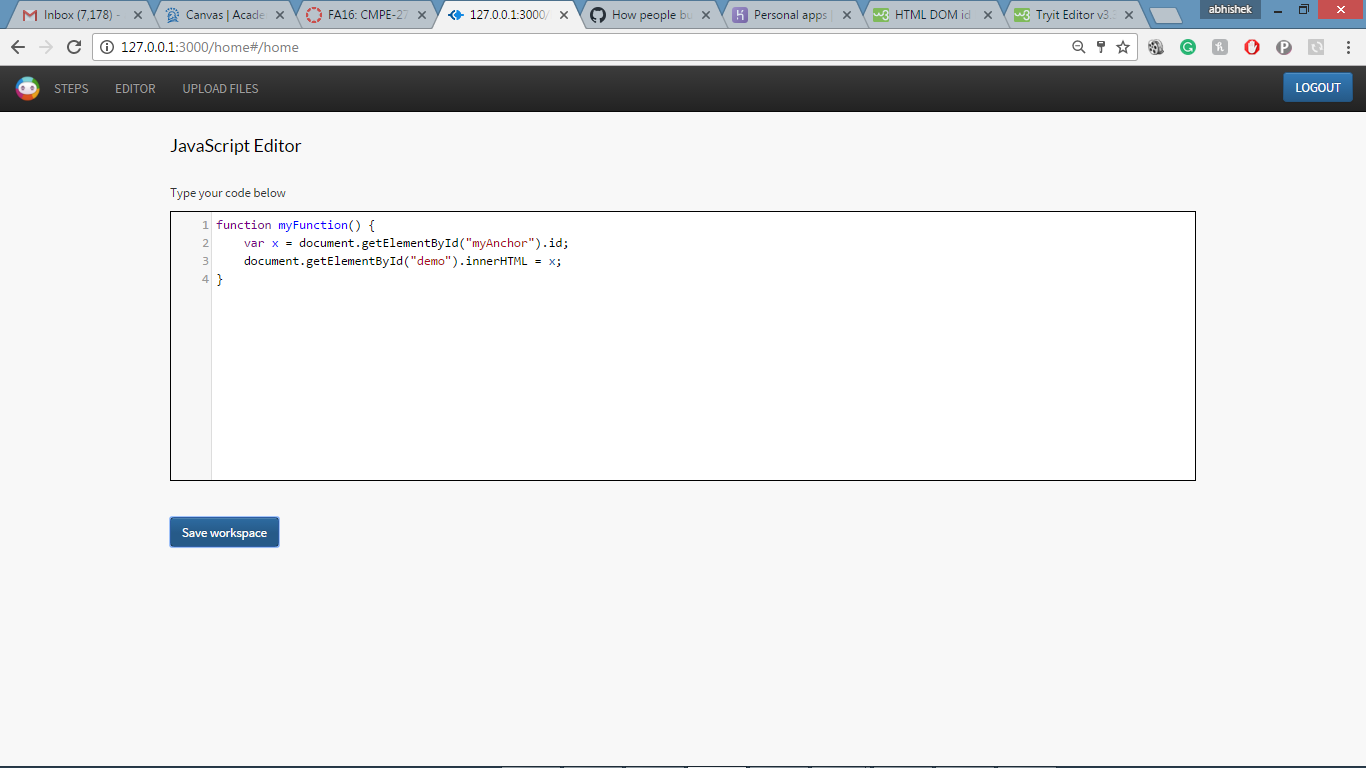
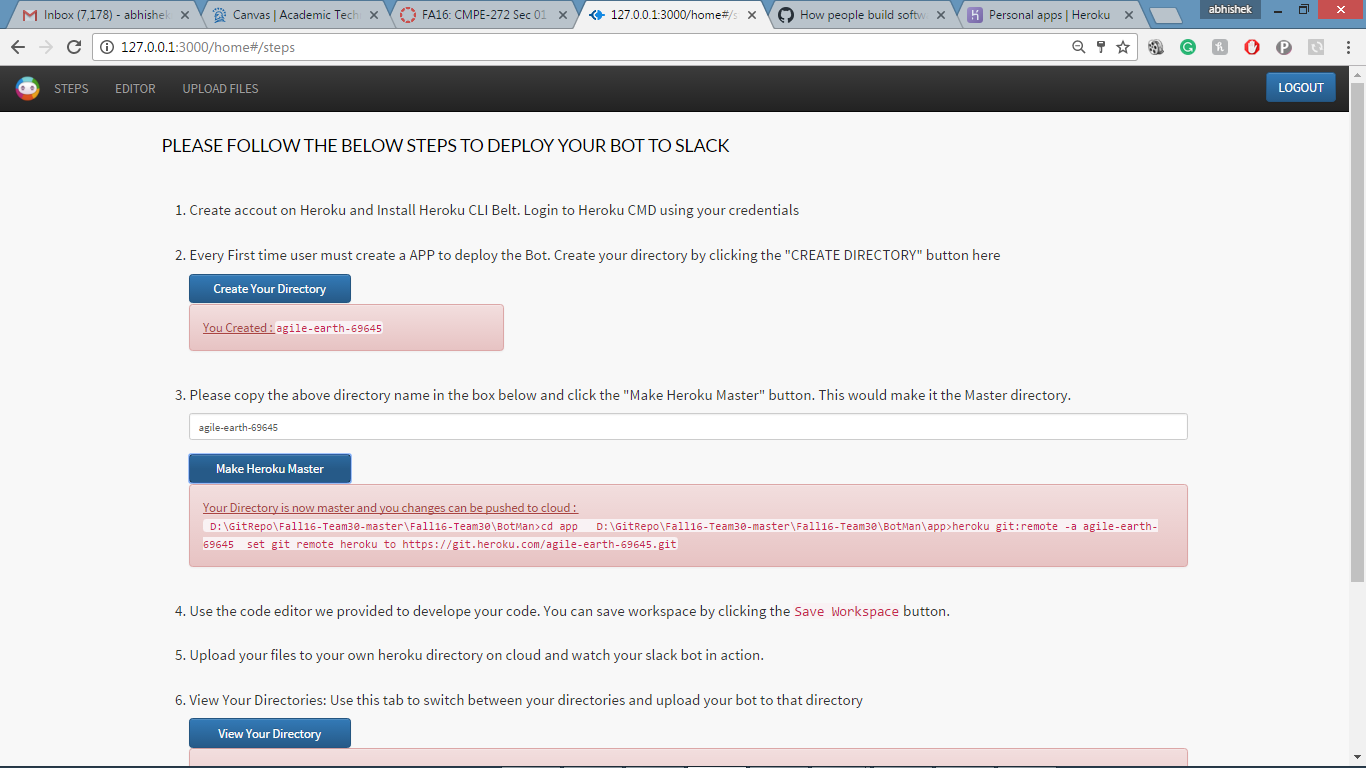
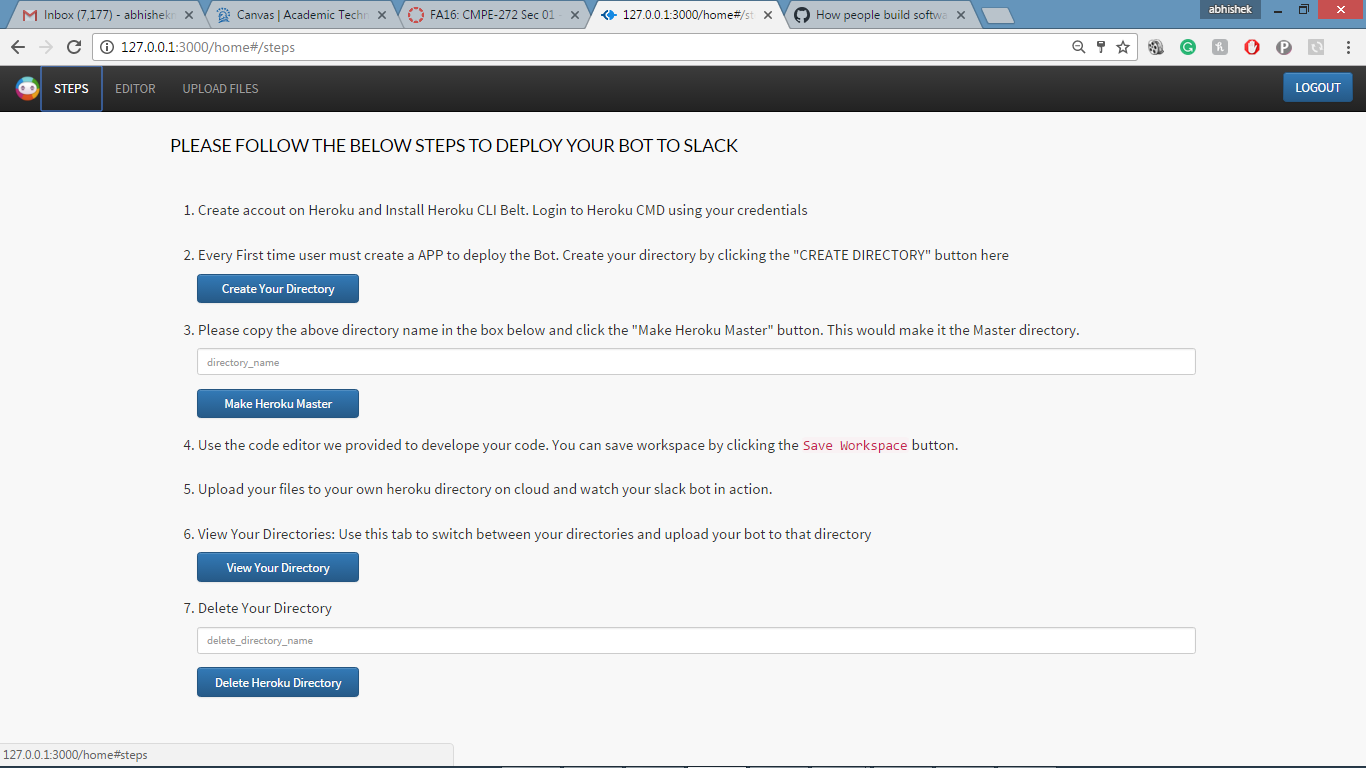
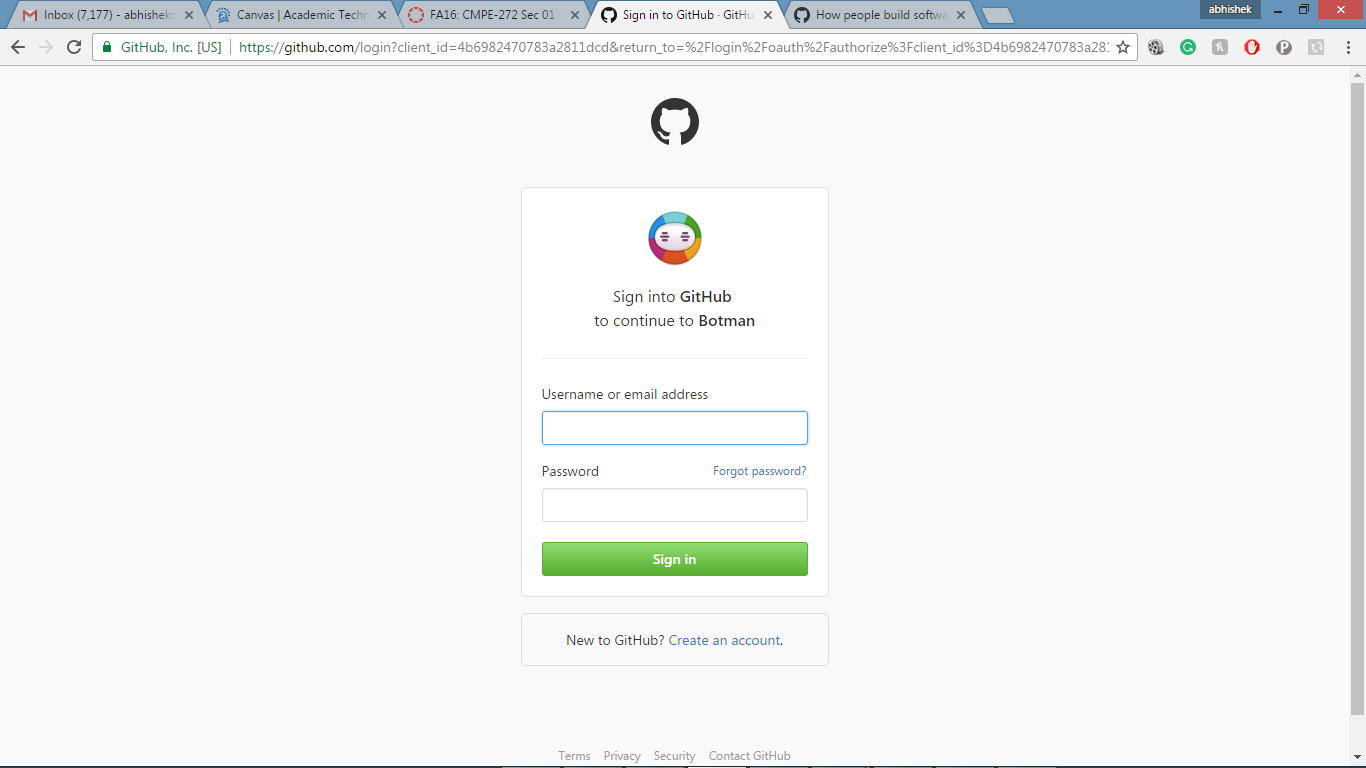
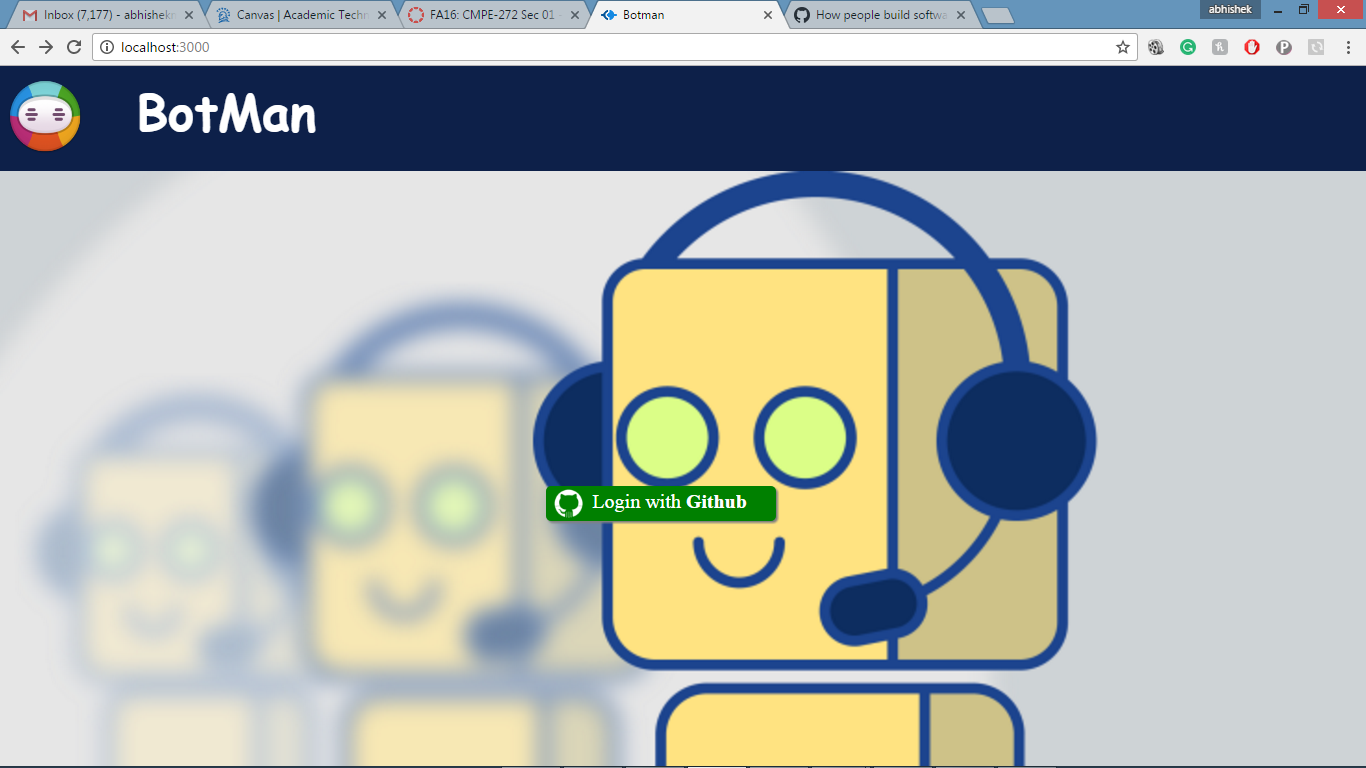
**BOTMAN**



**USER INTERFACE FLOW:**

C:\Users\saura\AppData\Local\Microsoft\Windows\INetCacheContent.Word\latest  (2).png

**APPLICATION SCREENSHOTS:**



**USECASES:**

Our framework could be used to create intelligent bots that could be useful in the following scenarios-

An office work group manager has to maintain list of the things that need to be bought for the meeting. This can be done by adding an inventory management bot to the slack channel. This bot could act as a manager and record the requests for stationary products. While discussing on the channel, if any employee thinks of something to be bought then instead of writing down in a separate notepad and then saving it somewhere, he can just request it over the channel and the bot could record them. This will reduce the overhead of physically recording things or creating a portal where the employee has to raise a request for the same.

In a project team while discussing the things sometimes someone uses words which are unknown to the others, in that case a project team can have a slack bot created through BotMan which will search those words in a dictionary and return the results to the team member. This way the communication between the team members will be efficient and productive.

An additional use case is where an organization like: Dominos can create its bot for order creation. Firms can integrate this bot on to their channel and just by a simple chat could be able to place an order.

**CHALLENGES:**

We encountered various challenges-

1. To embed a plugin in our web application that would help the user to type syntactically correct code. To overcome this, we used ‘*JSLint*’ and ‘*codemirror*’ plugins.
2. To find a node server which could run the bot code and keep it alive. Heroku(Paas) was an answer to this. Just by including a ‘*Procfile*’, we could configure set up the environment and configure the start point for the application on cloud.

**CONCLUSION:**

While bots have long lived in the quieter corners of the Internet, apps like Slack (and WhatsApp, Kik, and WeChat) are pushing them into the mainstream. A framework like Botman would give an impetus to the growth and would find its foothold in the market, assisting people to create customized bots for their personal and commercial use.

**FUTURE SCOPE FOR THE PROJECT:**

Currently our framework only allows a user to code his bot in Javascript programming language. In the next version of the framework, we would like to provide support for addition languages like: python and swift. This would help us popularize our application amongst developers who are comfortable with these languages.

**REFERENCES:**

*Project code:* [*https://github.com/SJSU272Lab/Fall16-Team30/tree/master/BotMan*](https://github.com/SJSU272Lab/Fall16-Team30/tree/master/BotMan)

1. <https://medium.com/@surmenok/chatbot-architecture-496f5bf820ed#.s5d7mvv87>
2. <https://www.sitepoint.com/custom-slackbot-with-node/>
3. <https://devcenter.heroku.com/articles>
4. <https://www.stackoverflow.com/>