1)Git Repositories:

Frontend code(Source Code Angular can be found here):

https://github.com/varun-murali-coder/vtracker-frontend

Backend code (API Documentation can also be found here):

https://github.com/varun-murali-coder/trackerBackend.git

2.)Web Hosting details(Domain name):

Frontend: http://vcoderlearn.com

Backend : api.vcoderlearn.com

3.) Project Design

a)Login Functionality:

->First the user has to register with VTRACKER tool using the sign-up button

->Once the user becomes a registered user of VTRACKER tool,he is redirected to the login page

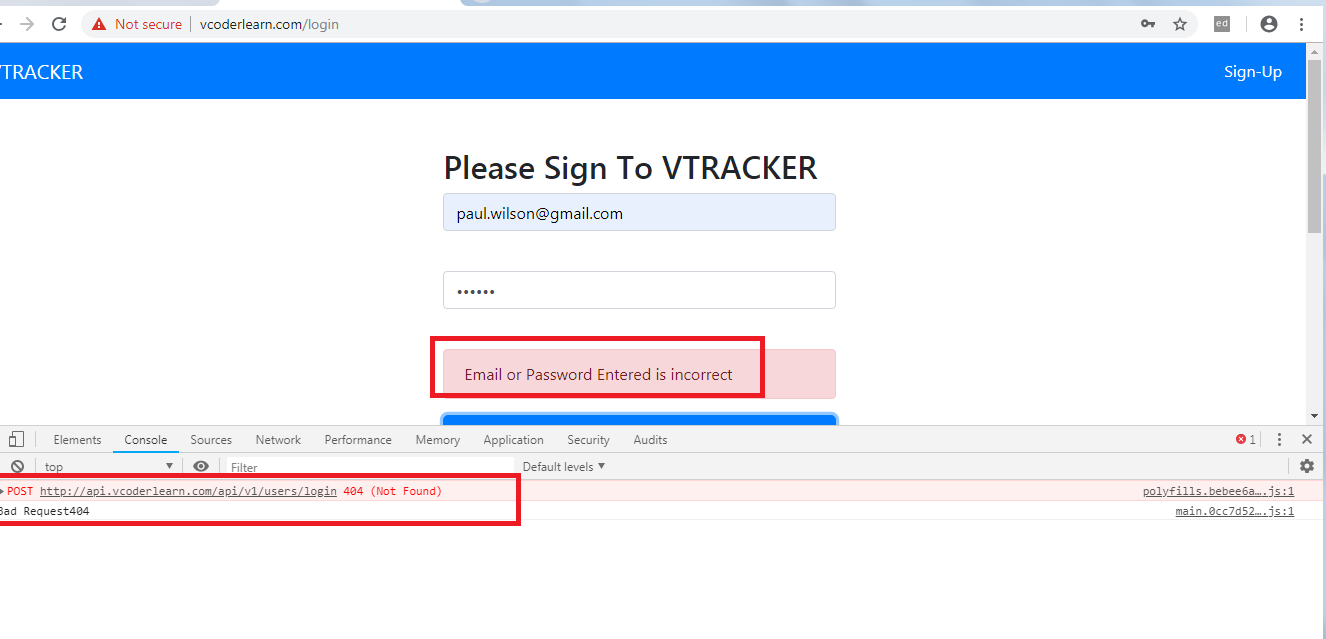
->Validations applied:

.All the fields are made mandatory

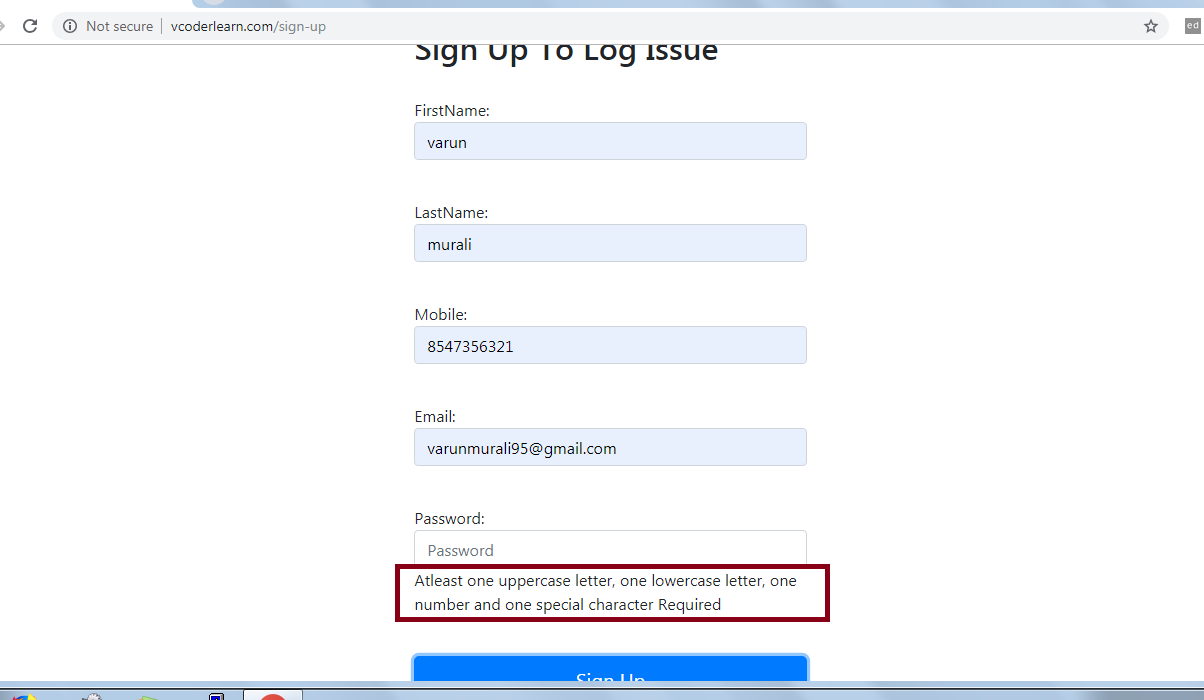
.After registeration if the user enters invalid email/password,appropriate validations are applied

Screenshots:

1.)Server side validation on login Page



2)Client Side validation on client page(After successfull sign up customer will be redirected to login page)



b)Dashboard Functionality:

->After successful login ,the user is redirected to the dashboard page where user has access to all the tickets assigned to him.

->The tickets are displayed in ag-grid, where the grid displaying the tickets provide options like filtering,pagination,sorting etc.

->Once the user clicks the title column he is redirected to the issue description page.

->There is a create button provided at the navbar which allows user to log a ticket.

->Also a search functionality is provided at the navbar via which user can have access to tickets assigned to others as well.

->Ticket will only be assigned if users full name matches the assignee field at the time of ticket description.

Note:Search functionality works only in dashboard page.

Additional feature:

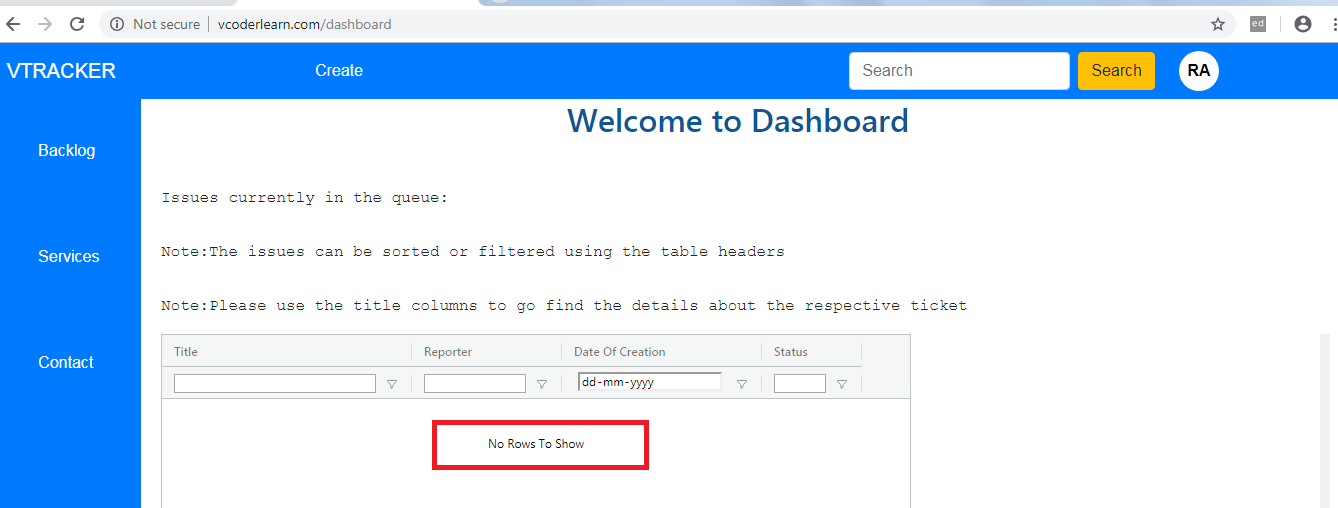
->A Backlog functionality is provided where user can get the information regarding all the tickets not having status:Done

->Circular logout functionality displaying the first two letters of logged in user.

->Create functionality made with the help of modal

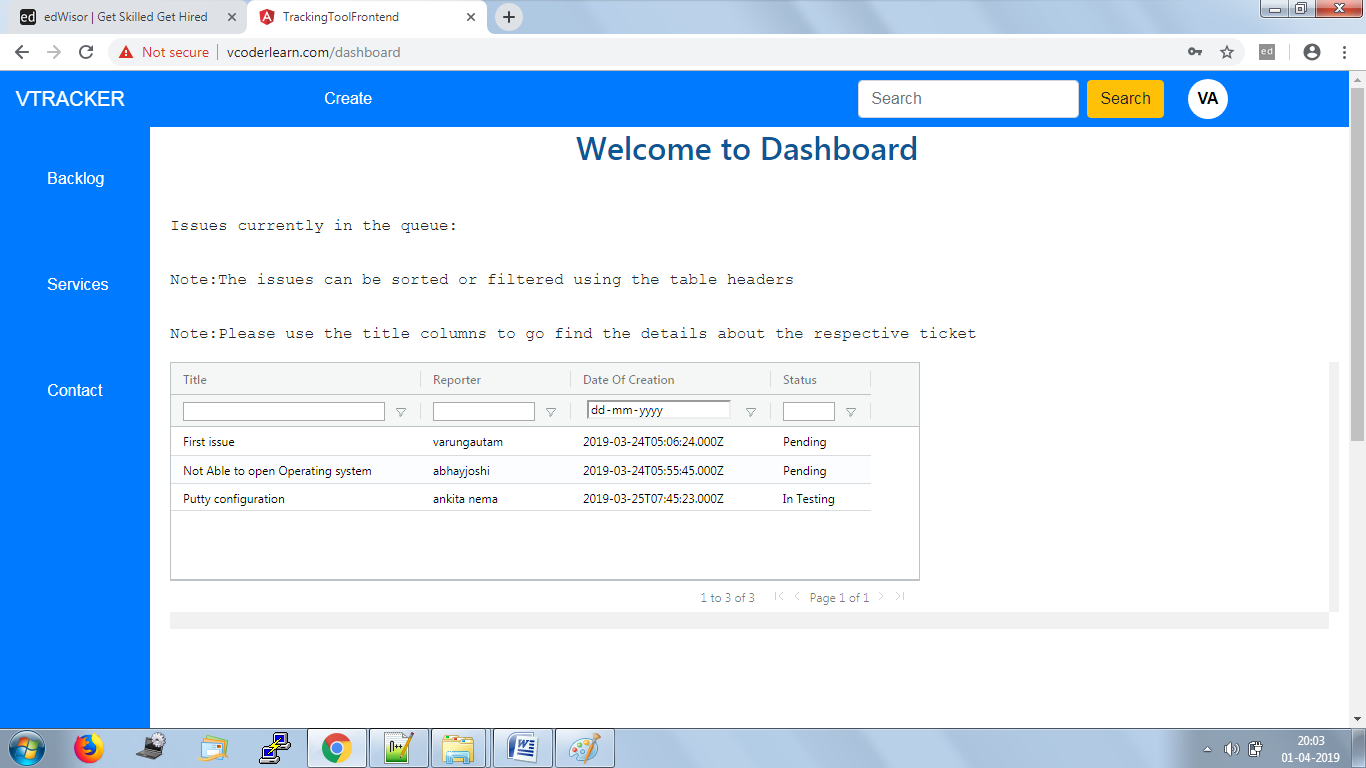
Screenshots:

1)Handling server side task not found case:

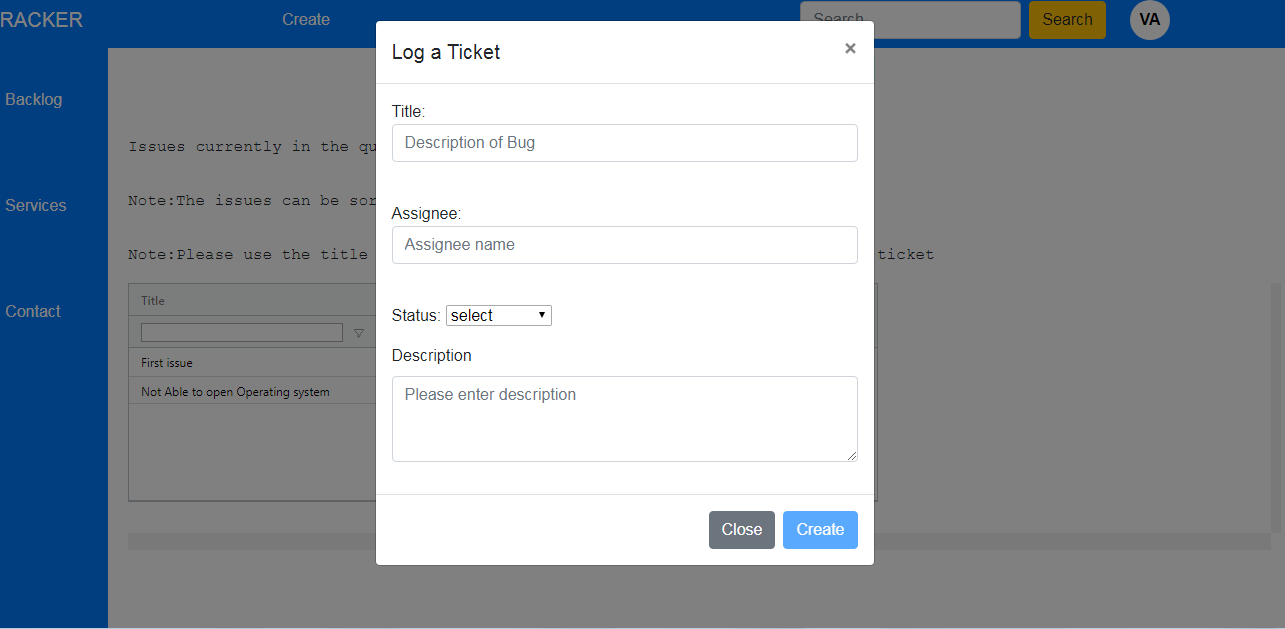


2)Dashboard page

Note:In the tables:Filteration,Pagination,Sorting etc can be performed.



3)Ticket create Functionality:



c)Search Functionality

->Smart keyword search funtionality has been implemented, where search filter are displayed for each keyword entered by user.

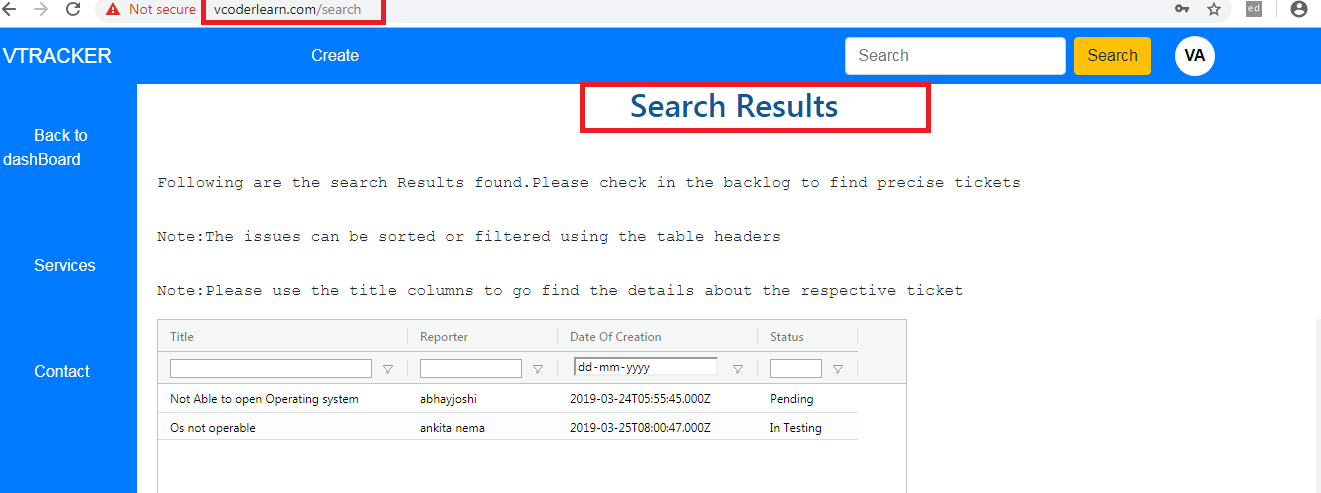
->This page provides the user option to become the watcher for other issues which are not assigned to him.

->Once the user clicks the title column he is redirected to the issue description page.

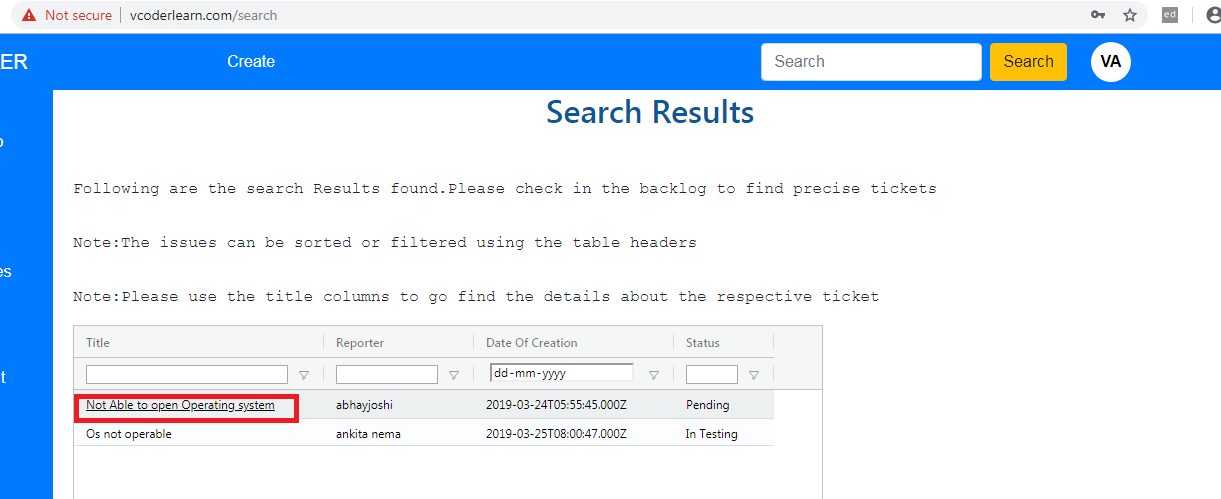
Note:Search functionality is meant only for Dashboard page as mentioned in the problem statement.

Screenshots:

1)Search Page



2)Browsing from search page to issue description page can be done using the title column



d.)Issue Description Page

->Edit and View Functionality kept together.User can make use of save button to commit the changes.

->Add/Edit functionality is available for all the users,via which ticket details can be changed.

->Once the user clicks the save button he will be redirected to the dashboard page getting which in turn will get populated with the edited values.

->Back button is provided for the user to go back to dashboard

->Watch button is provided so that user gets notification related to a particular ticket he is interested in.

->Once the user clicks the watch button he is redirected back to the dashboard so that he can add himself as watcher to other tickets as well

->The watchers list is also getting displayed on the issue description page

->Description section is made using quill-editor with additional feature showing the number of characters that can be entered.

->Any user is also allowed to post any comment using the post a comment button which is persisted even after the user moves to some other page.

->The notification functionality was implemented using the angular service worker,but the trigger function to get the notification logic is not written,since

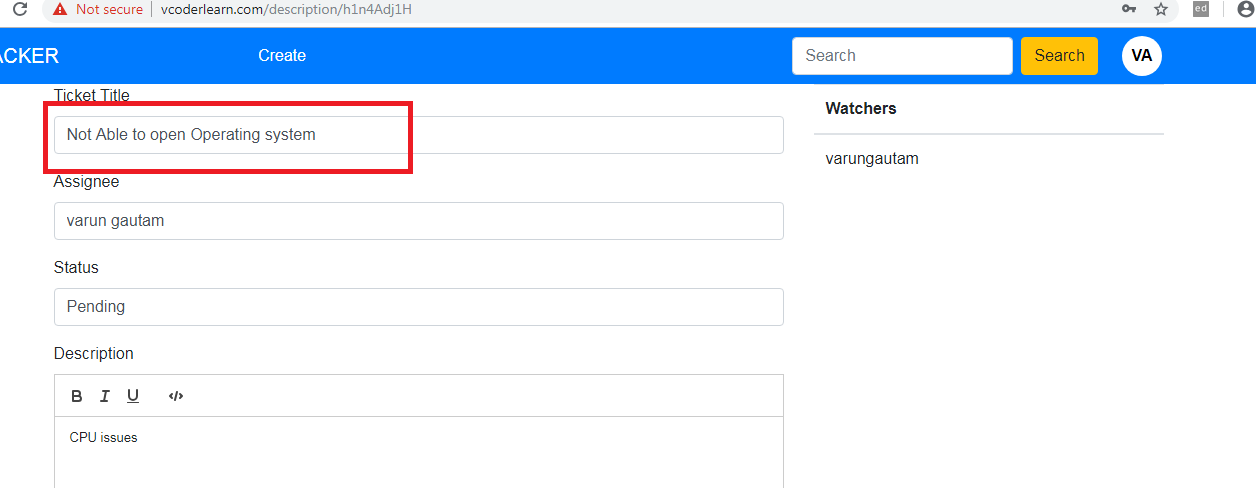
angular service worker is supported under https protocol.The code references can be found in the git repo.

->The subscription logic is written on the click of watch button.

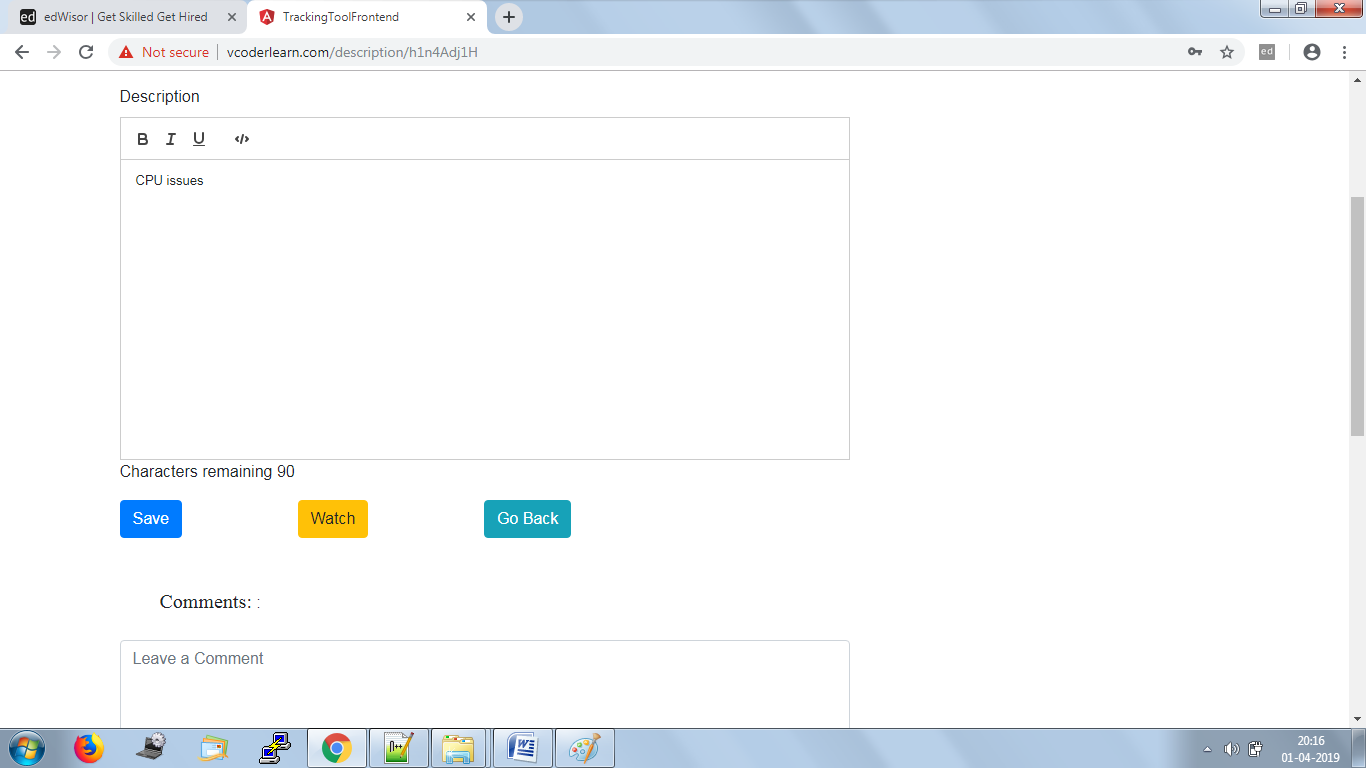
->File upload using multer has also been left out due to functionality breaks whle deployment.

Screenshots:

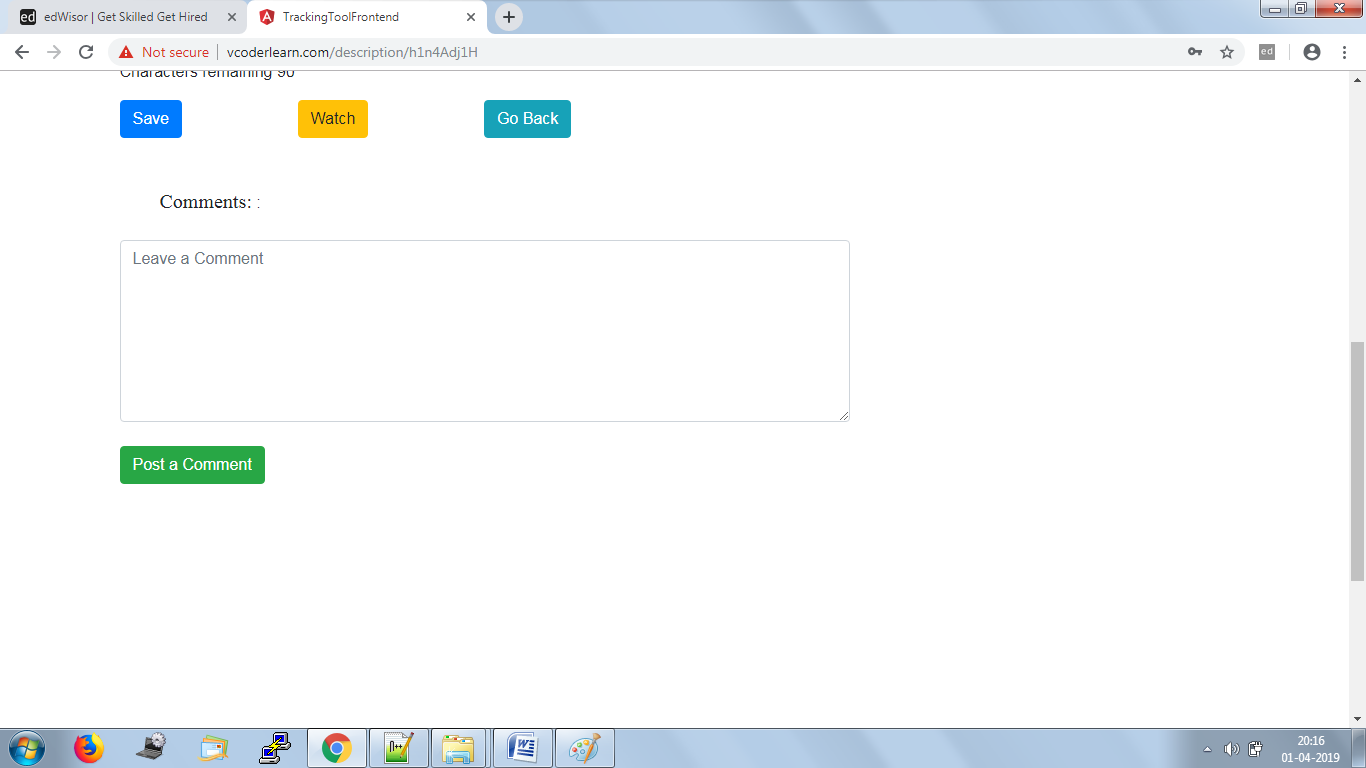
1)Edit as well as view functionality:



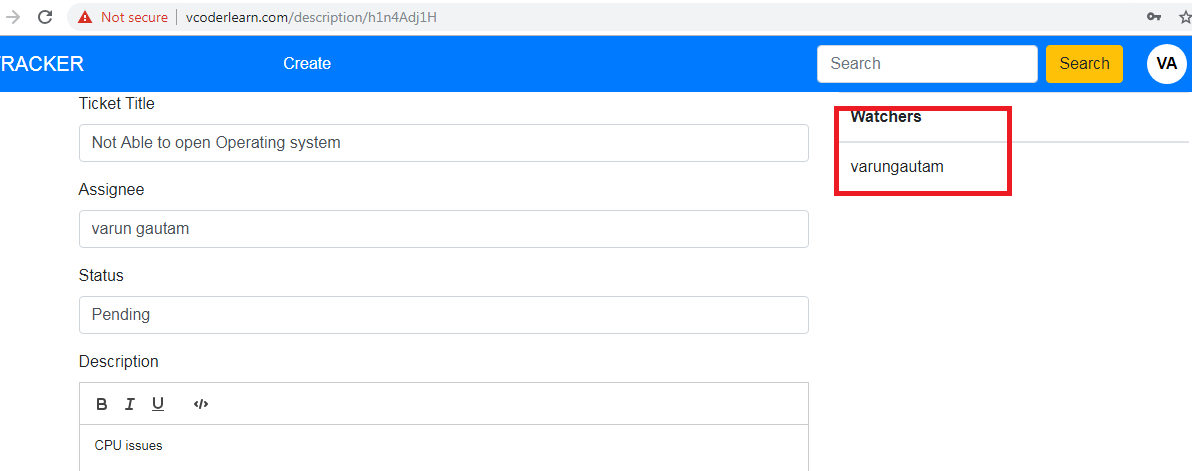
2)Description section :



3)Comment section

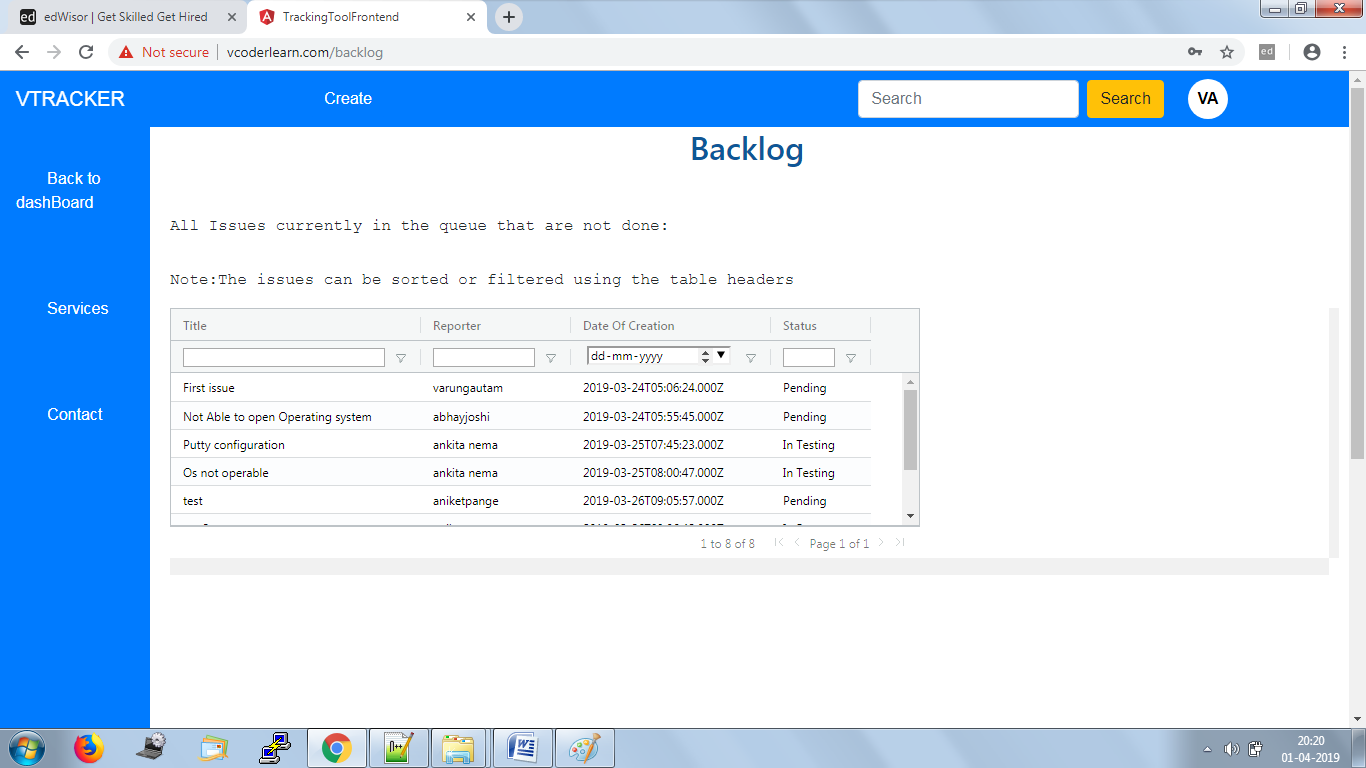


4)Watchers list:You become a part of the issue once you click the watch button

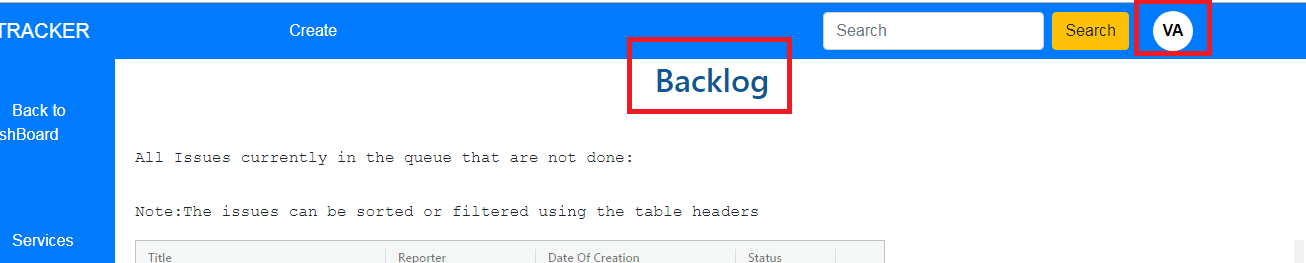


Additional Functionalities:

1)Backlog page with view only access:User can the list of tickets yet to be resolved from here



2)Two letter logout button:



Test Users:

1)varung@gmail.com/Varun@1234

2)abhayjoshi@gmail.com/Abhay@1234

3)rateesh@gmail.com/Rateesh@1234