OPPRTUNITY COLLABORATION  
Website

June 12, 2017

# Overview

## Project Background and Description

This is a website application for writing blogs, chatting with friends, searching jobs, searching events and join in forum etc.

## Project Scope

* Website application for writing blogs, chatting with friends, searching jobs, searching events and join in forum etc.
* Consists interactive interfaces to access different blogs, forum, jobs, events, etc.
* Interfaces will enable to search blogs, events, jobs, friends,etc.

## Software and Hardware Requirements

* Software
  + HTML5, CSS, Bootstrap
  + AngularJS, JavaScript, jQuery, JSON, SockJS, STOMP
  + Java, Spring MVC, Hibernate, WEbSocket
  + Oracle Database
* Hardware
  + Processor - Intel Core i5/i7
  + Installed Memory(RAM) – At least 4.00GB
  + System Type – 64 bit OS, x64 based processor
  + Network - Internet

## High-Level Requirements

This system includes the following:

* Ability to allow both internal and external users to access the application without downloading any software.
* Ability to interface with the existing database application.
* Ability to incorporate automated routing and notifications based on business rules(optional).

## Deliverables

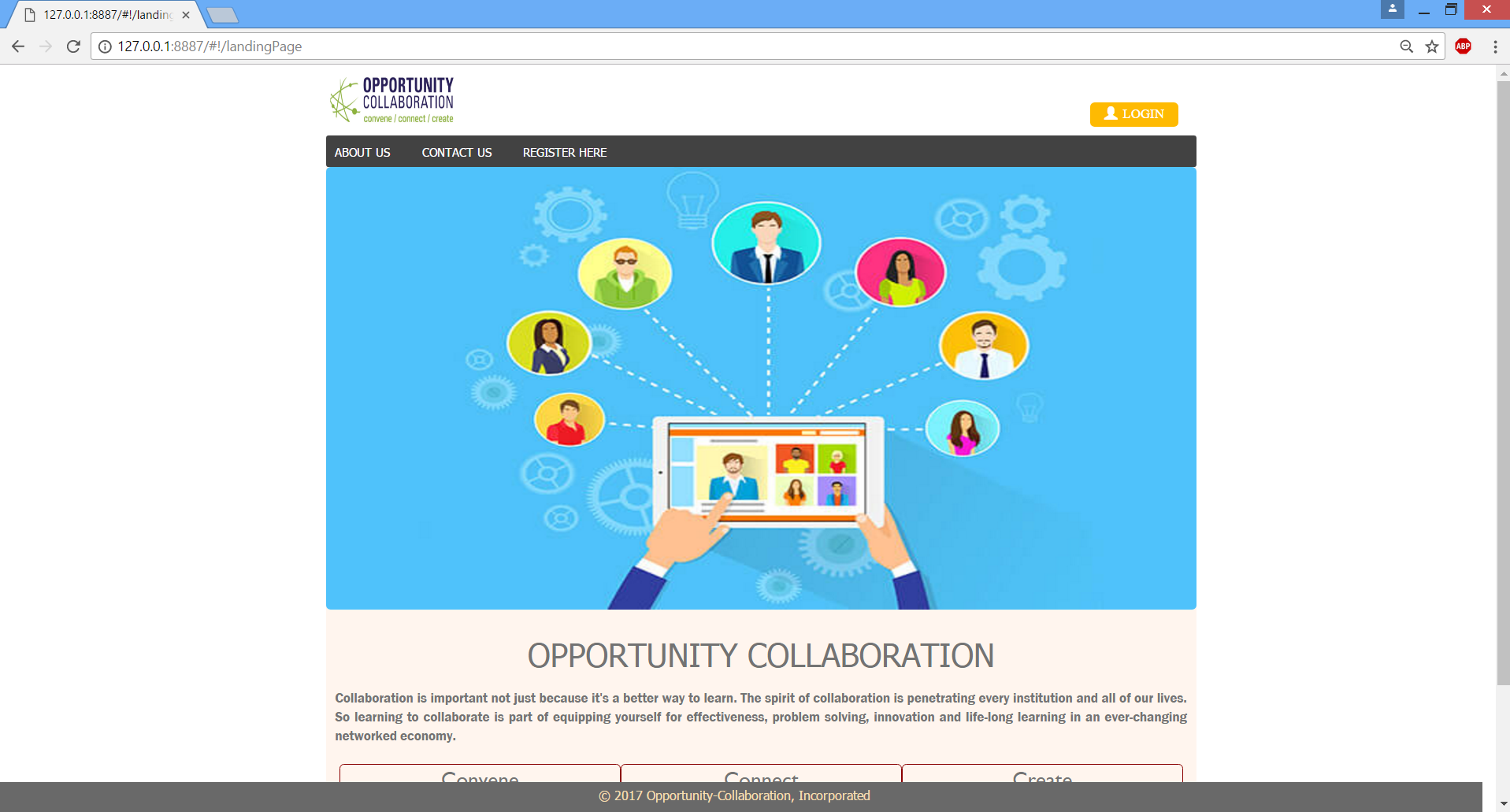
* Admin User will be able to insert, activate, update and remove blogs, jobs, events, forums.
* User will be able to view blogs, friends, jobs, events, forums.
* User will be able to upload profile picture.
* Admin will activate user after registration process by sending auto generated confirmation email.
* User and Admin can chat with online friends.
* User can join events, apply for job, join forum, write to forum, write blogs, etc.

# Approval and Authority to Proceed

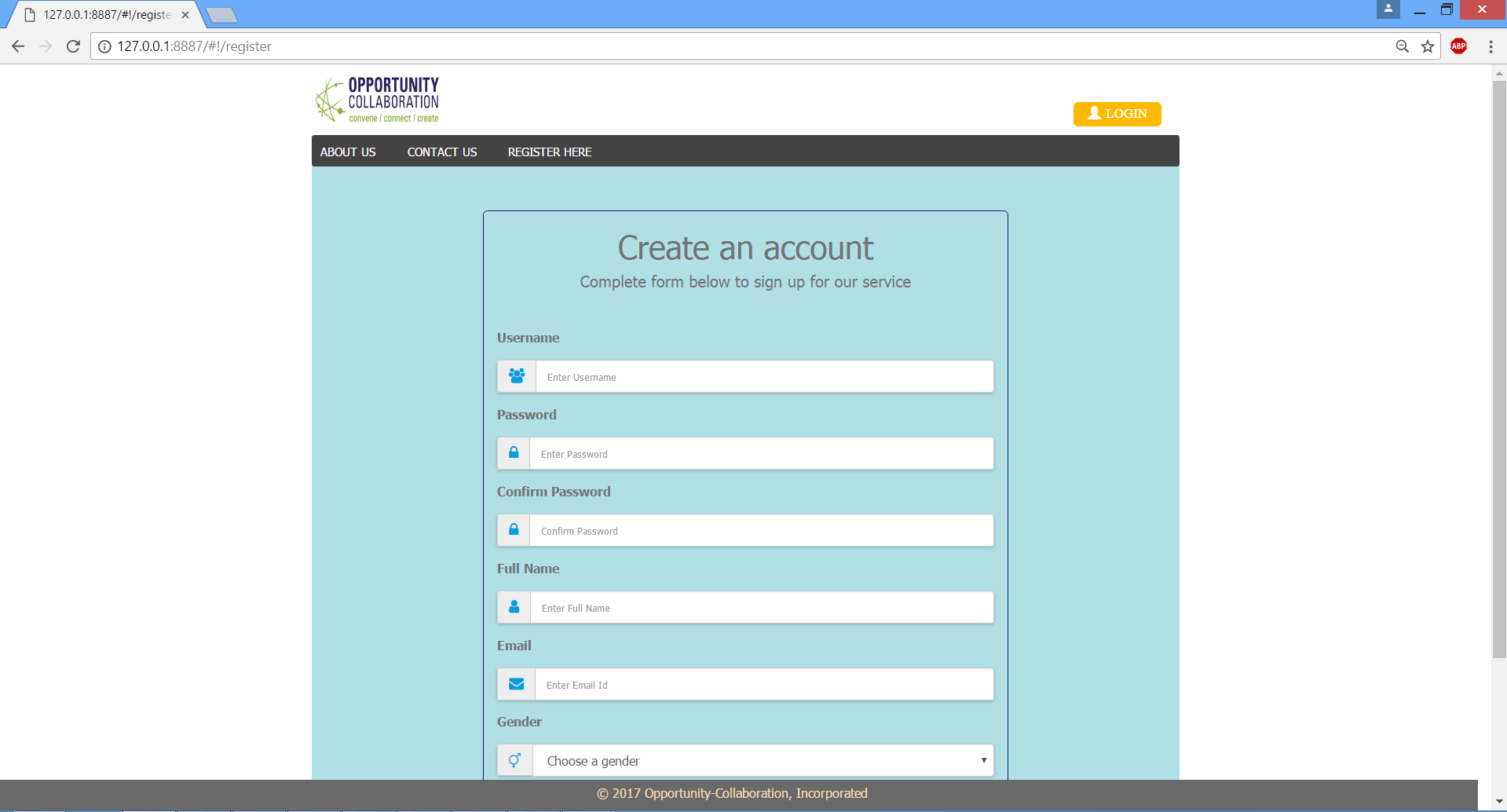
|  |  |  |
| --- | --- | --- |
| Name | Title | Date |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| Approved By |  |  | Date |  | Approved By |  |  | Date |

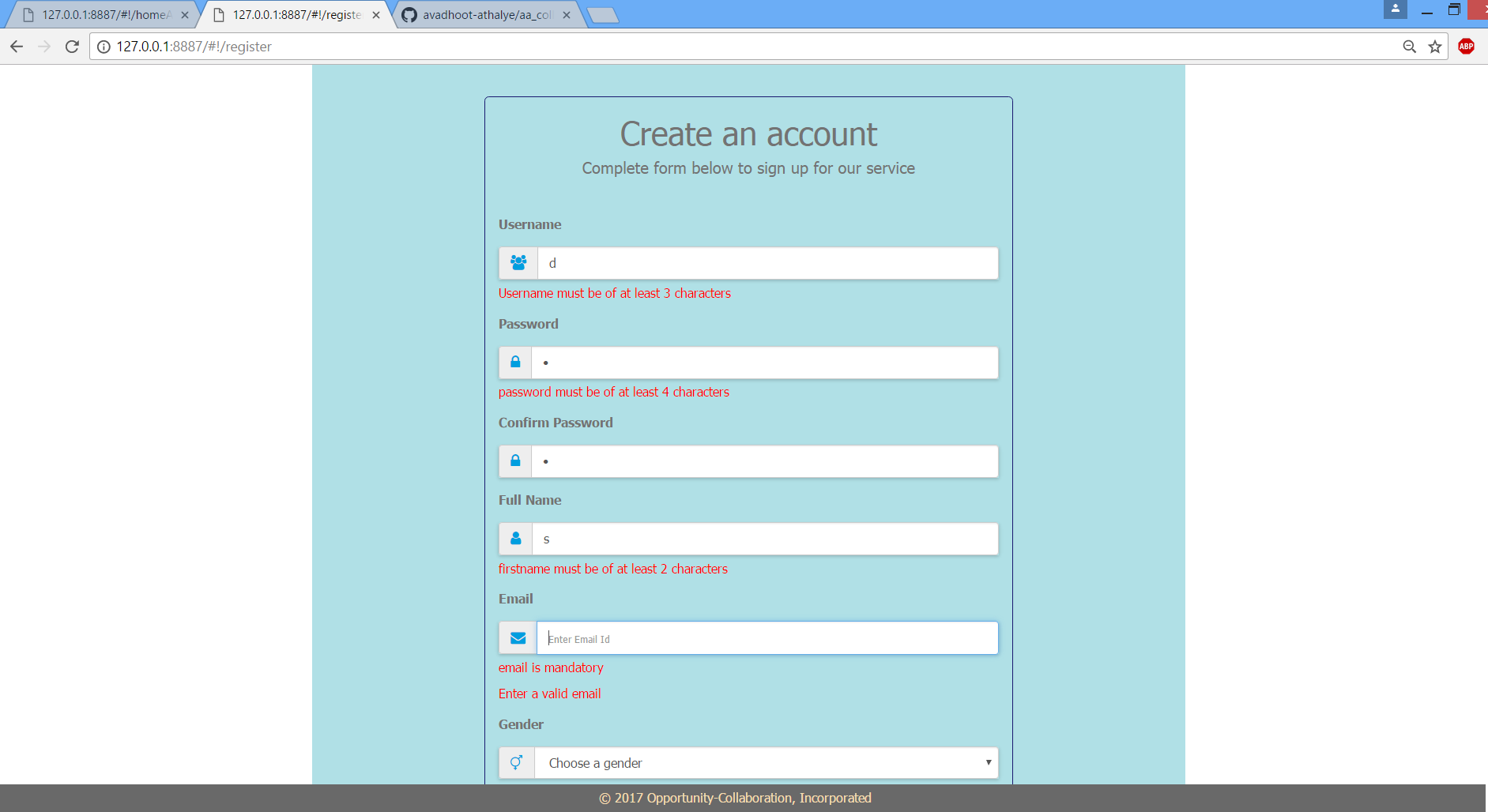
1. Landing Page



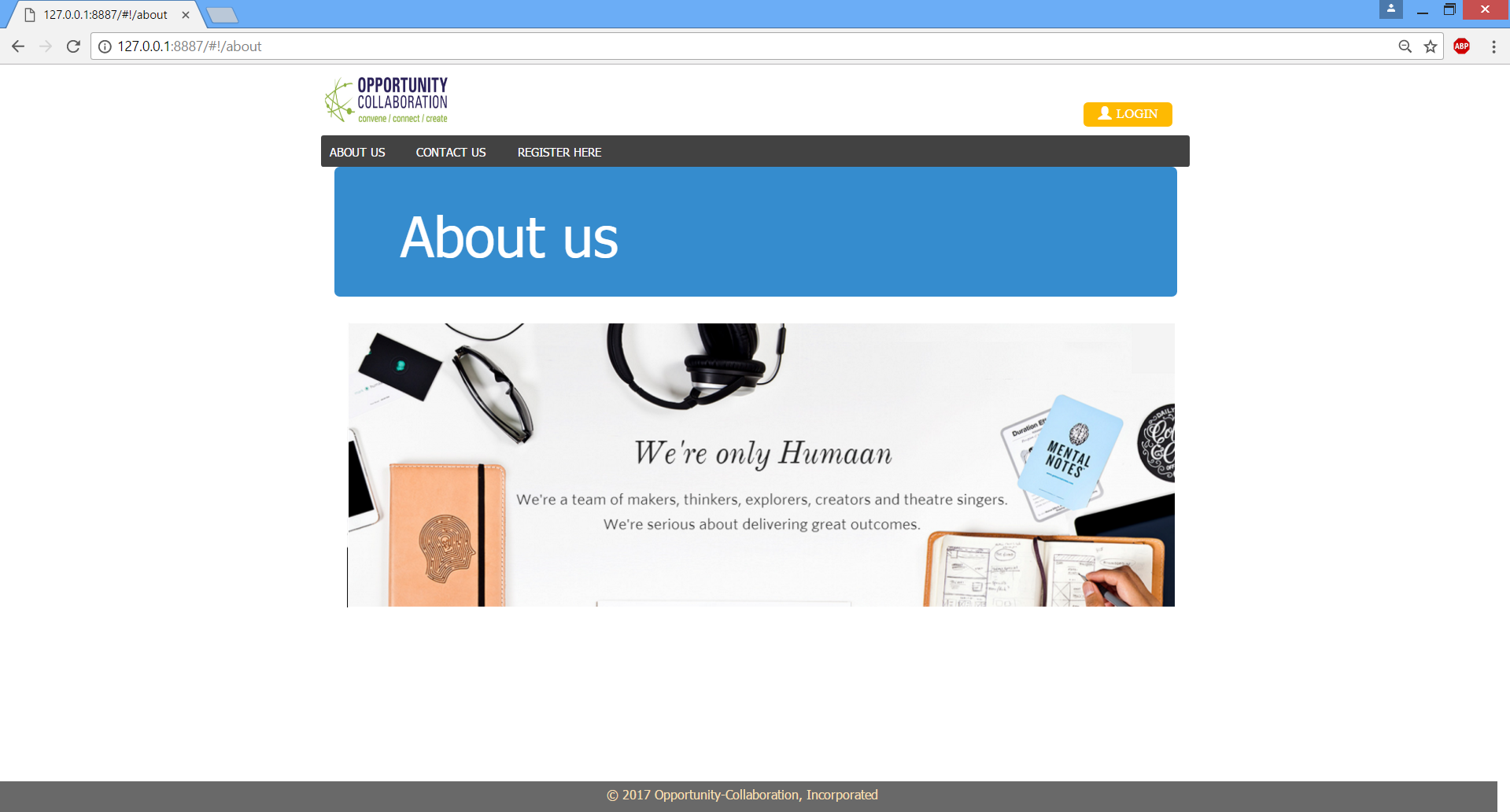
1. Technologies used to design the landing page are HTML, CSS, Bootstrap, JavaScript.
2. User will click to REGISTER HERE menu to perform registration if user is not registered yet.
3. Registration Page



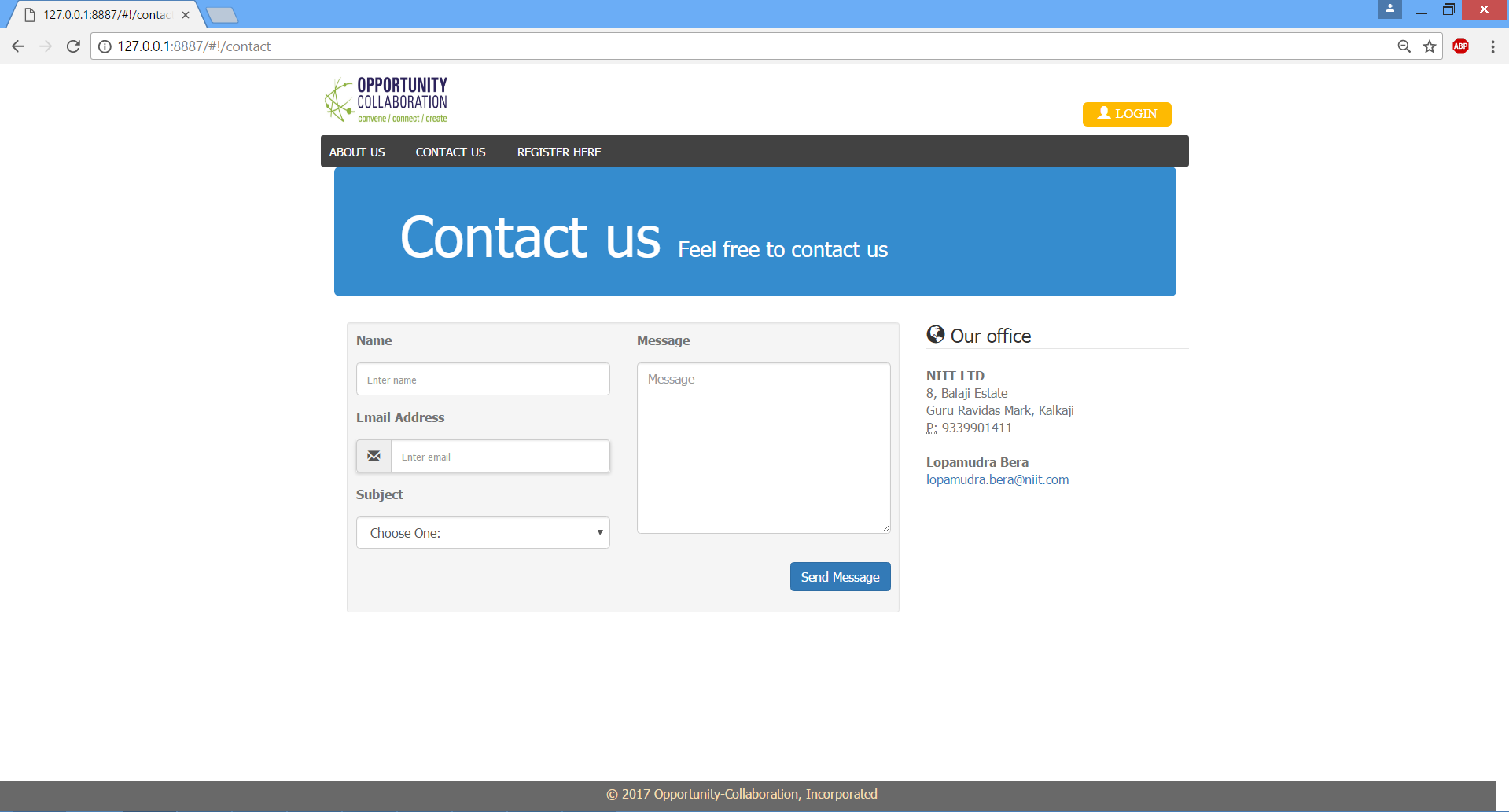
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular RegisterController to Angular RegisterServices which calls backend API method for registration.
3. AngularJS validation is used to validate this page.
4. After successful registration, user will go back to Landing page for Login.
5. Only ADMIN can login immediately after registration.
6. USER required to be approved by ADMIN.
7. Register page with Validation error messages



1. AboutUs Page



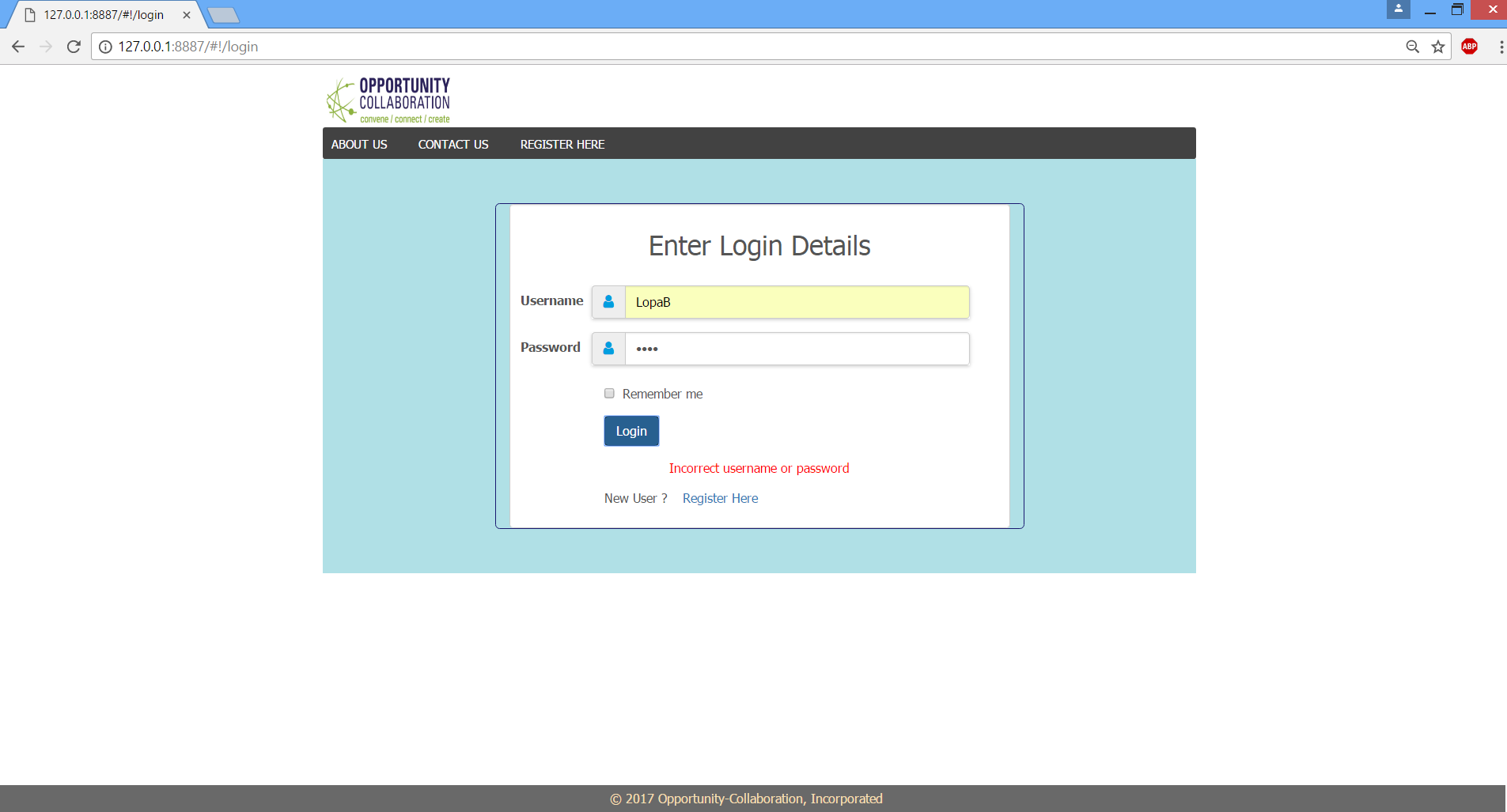
1. ContactUs Page



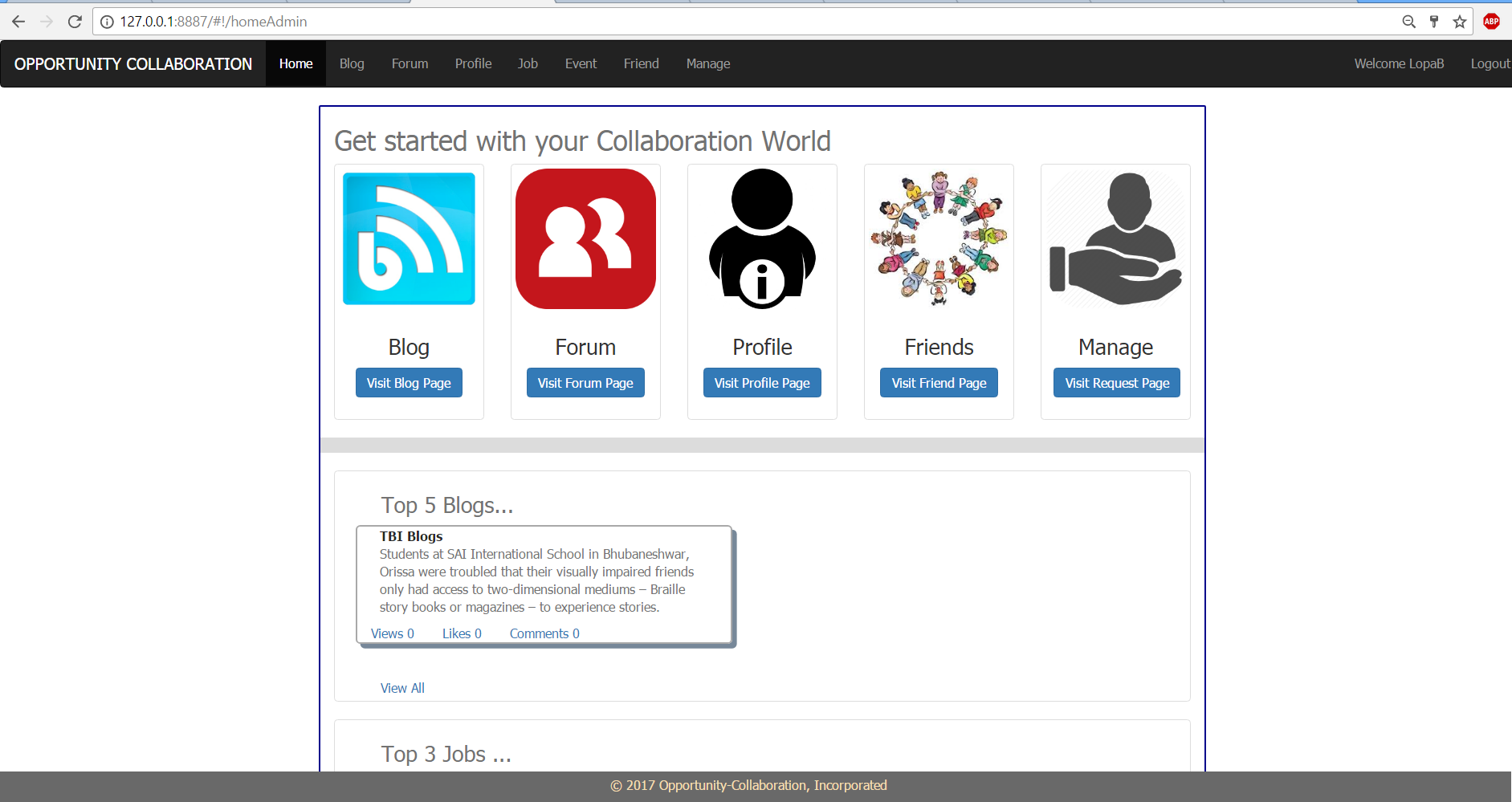
1. Login Page



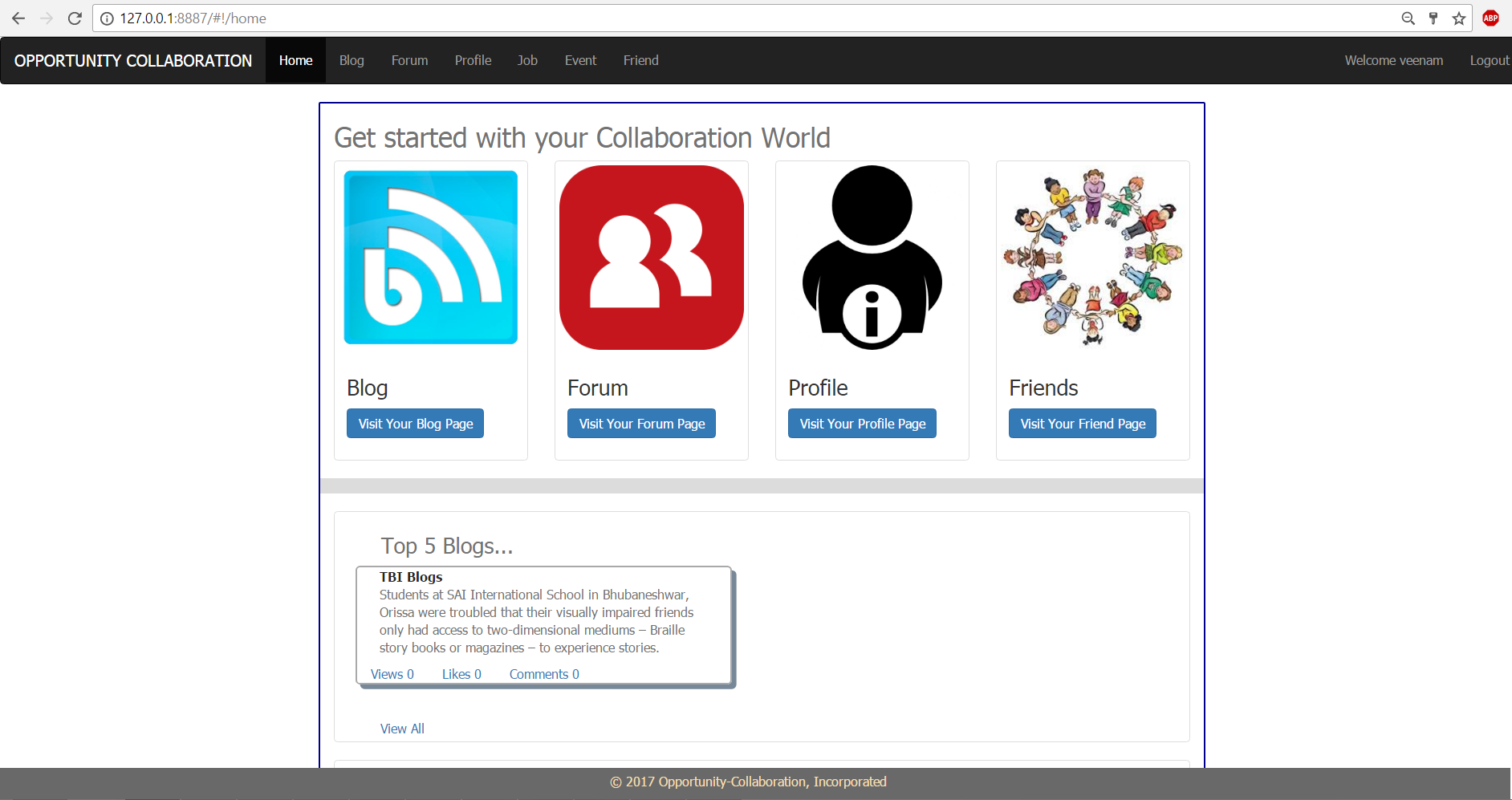
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular RegisterController to Angular RegisterService which calls API method for login.
3. AngularJS validation is used to validate this page.
4. After successful login ADMIN will move to ADMIN HOME page and USER will move to HOME page
5. Login Page with Validation



1. Admin Home Page



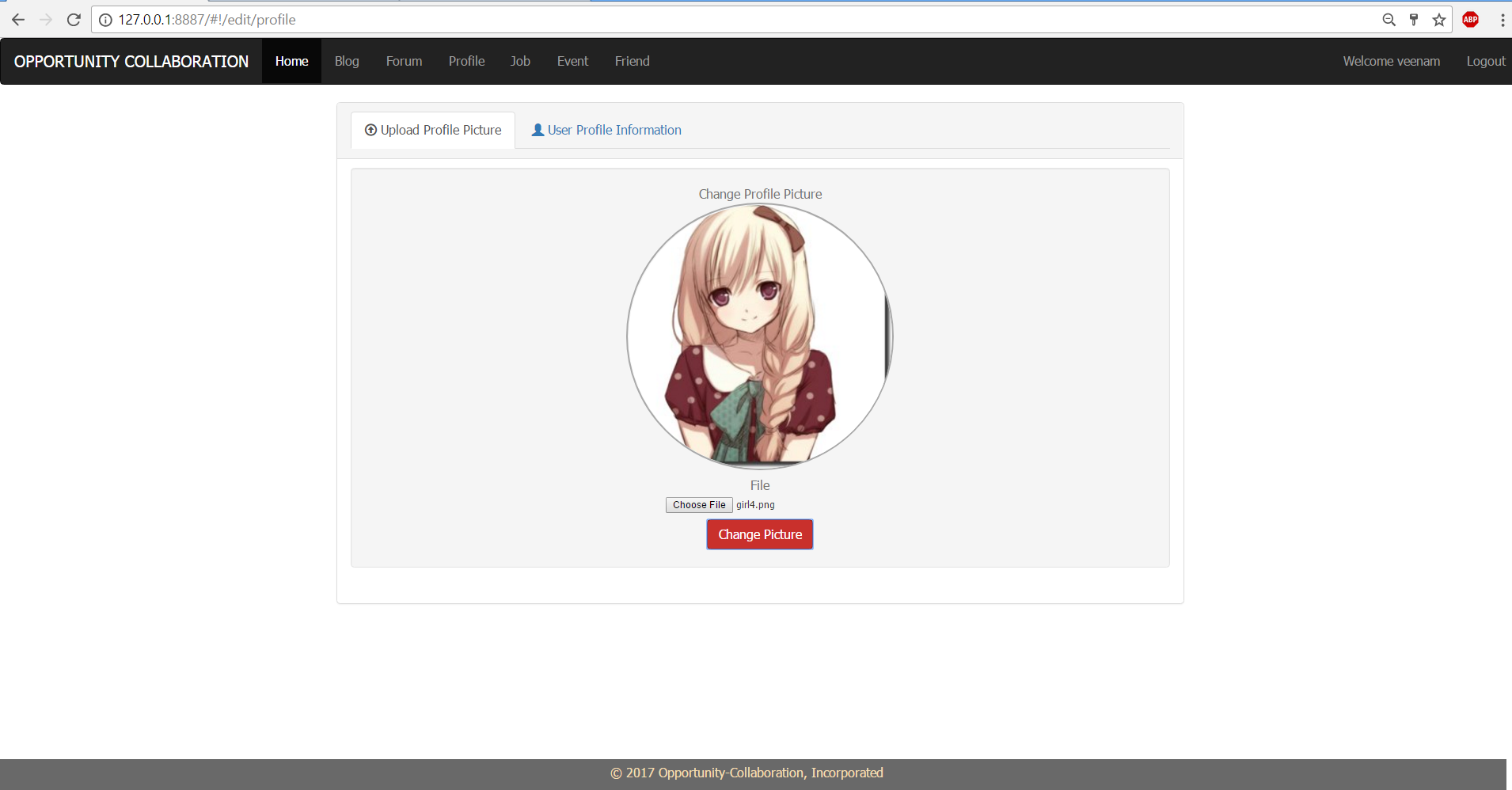
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UserController to Angular UserService which calls API method for showing username and logout process.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(homeAdmin.html).
4. Admin can navigate to any link from this page like blog, forum, profile, friends, managing admin activity like activate user, activate blog, etc.
5. Home page also contains with top 5 blogs, top 3 Forums, top 3 events, top 3 jobs.
6. User Home Page



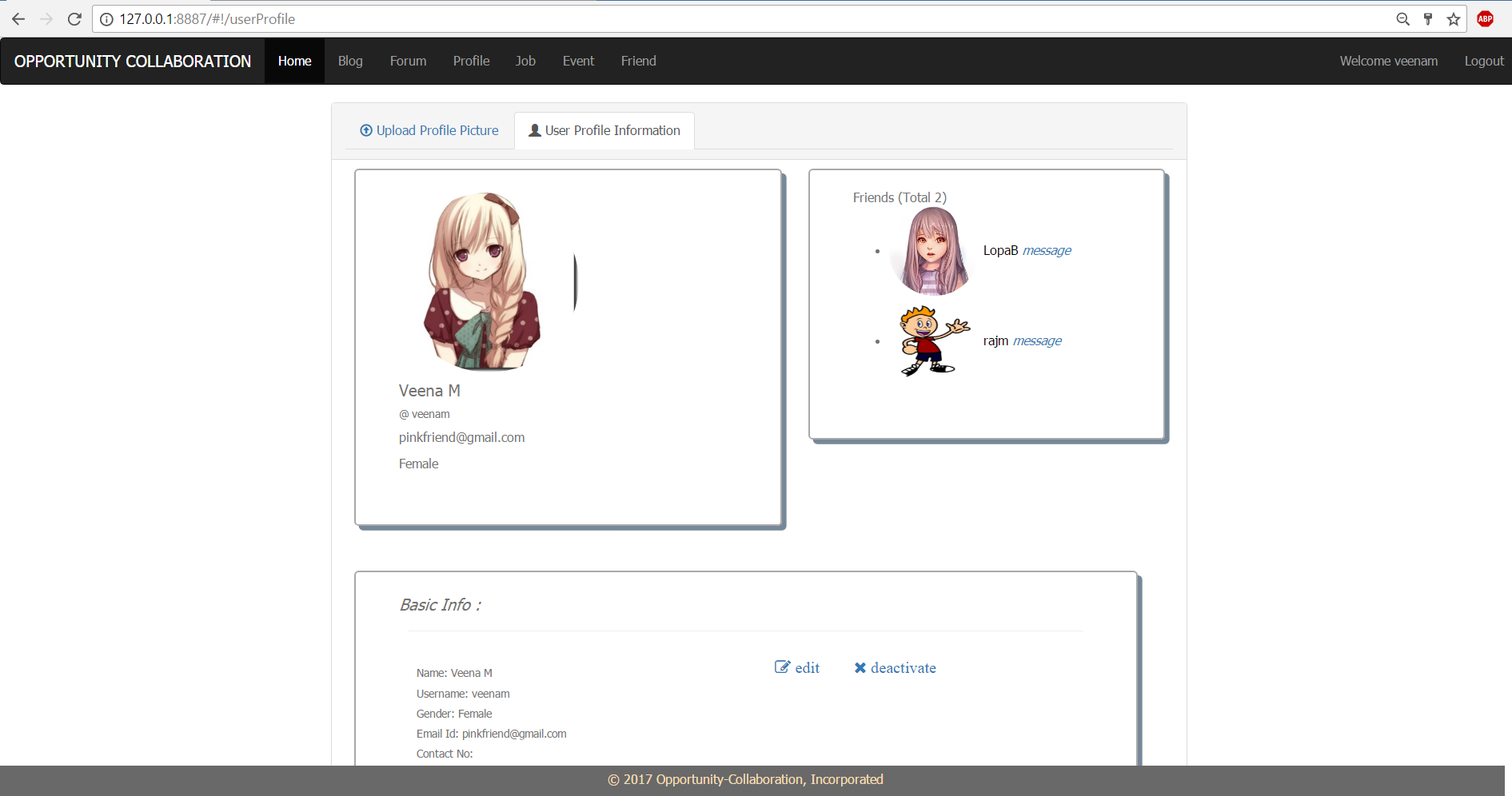
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UserController to Angular UserService which calls API method for showing username and logout process.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(home.html).
4. Admin can navigate to any link from this page like blog, forum, profile, friends, etc.
5. Home page also contains with top 5 blogs, top 3 Forums, top 3 events, top 3 jobs.
6. Upload Profile Picture



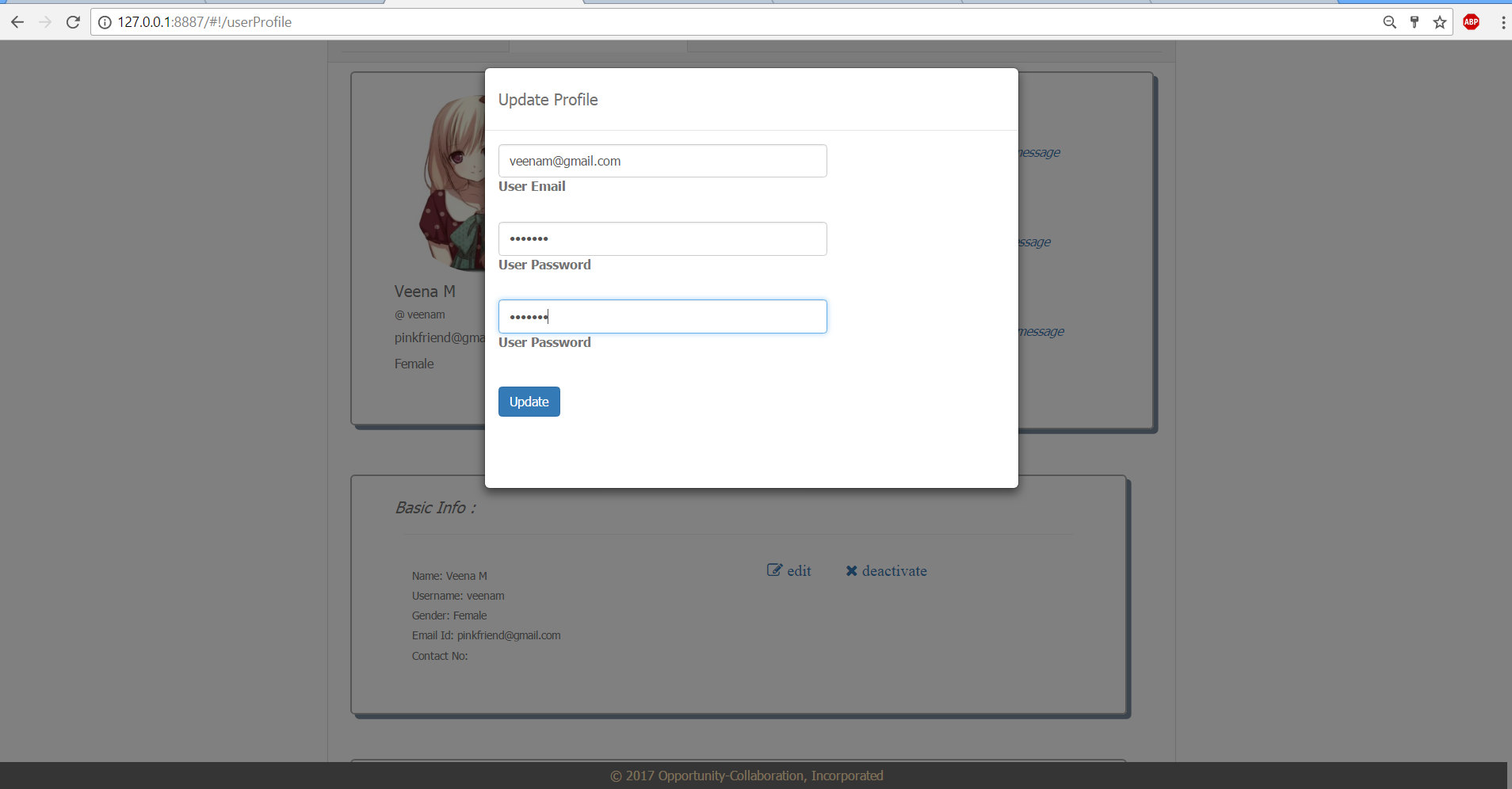
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UploadController to Angular UploadService which calls API method for upload file.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(editProfile.html).
4. This page will allow to upload profile picture.
5. Dependencies used in backend project like commons-fileupload, commons-io.
6. After uploading the picture



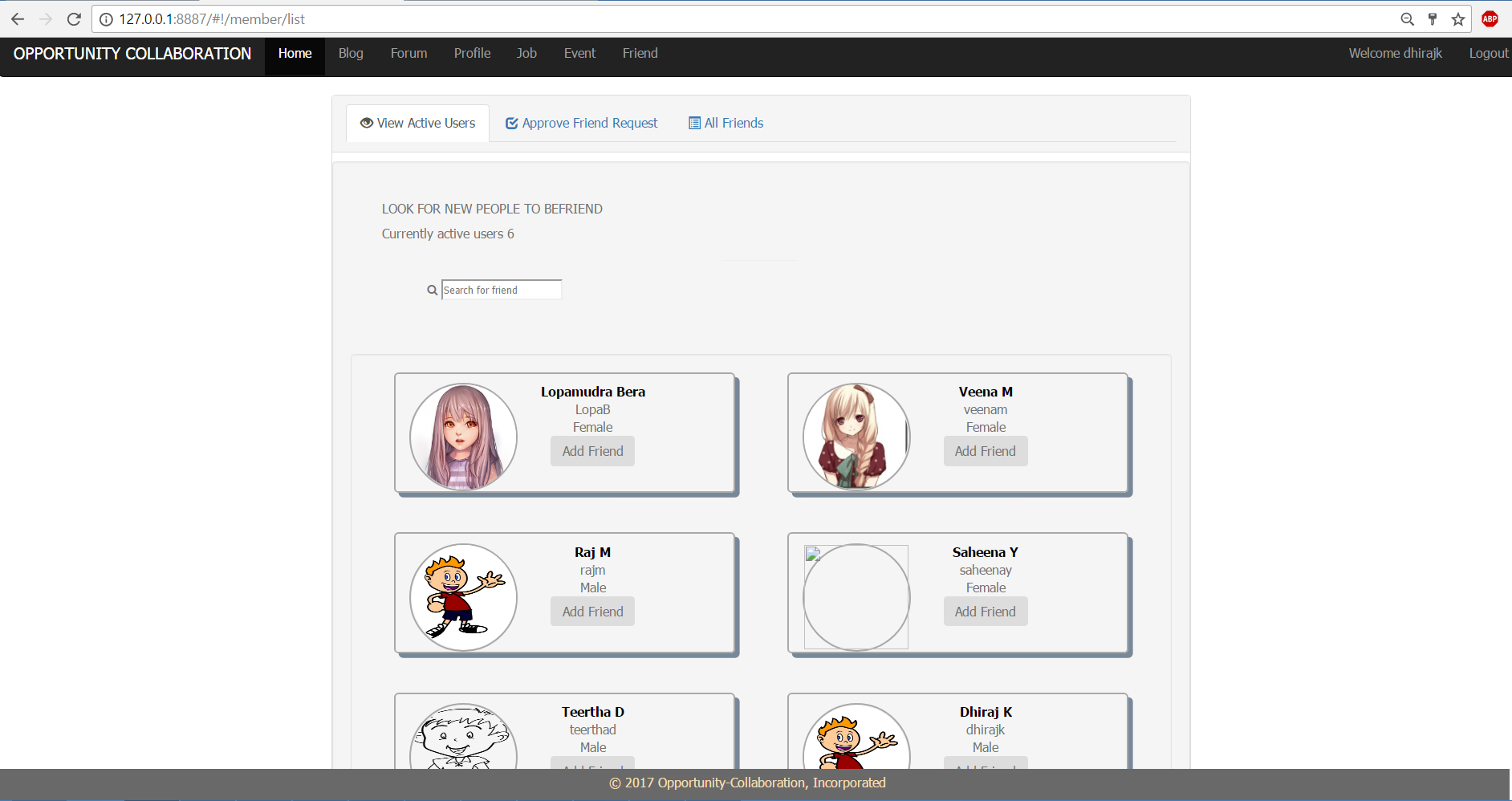
1. View User Profile Page



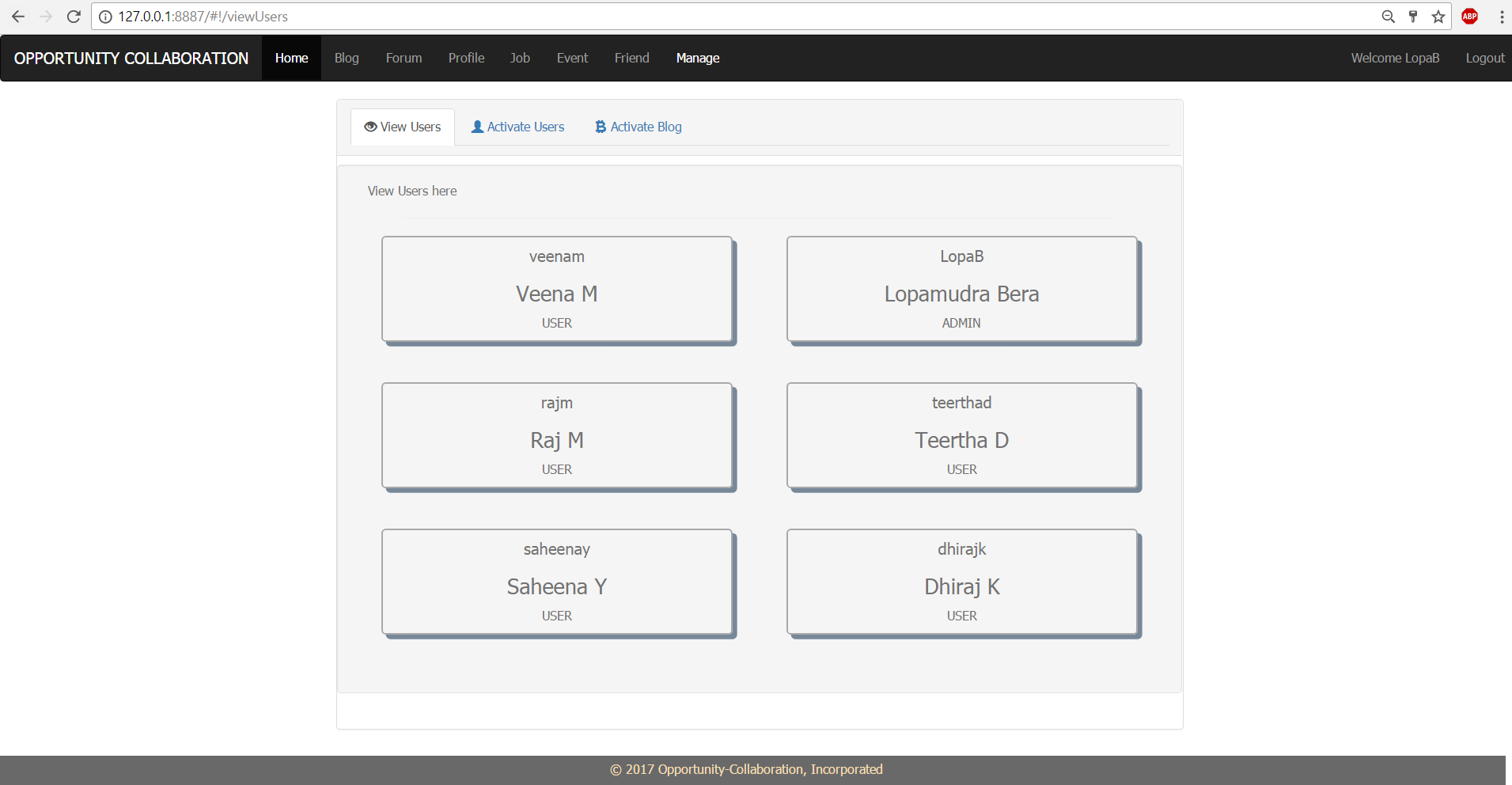
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UserController to Angular UserService which calls API method for view user detail.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(userProfile.html).
4. This page will display user’s all detail like profile information, blog posted, job applied, event joined. Also user can make himself/herself deactivate.
5. User Profile Update



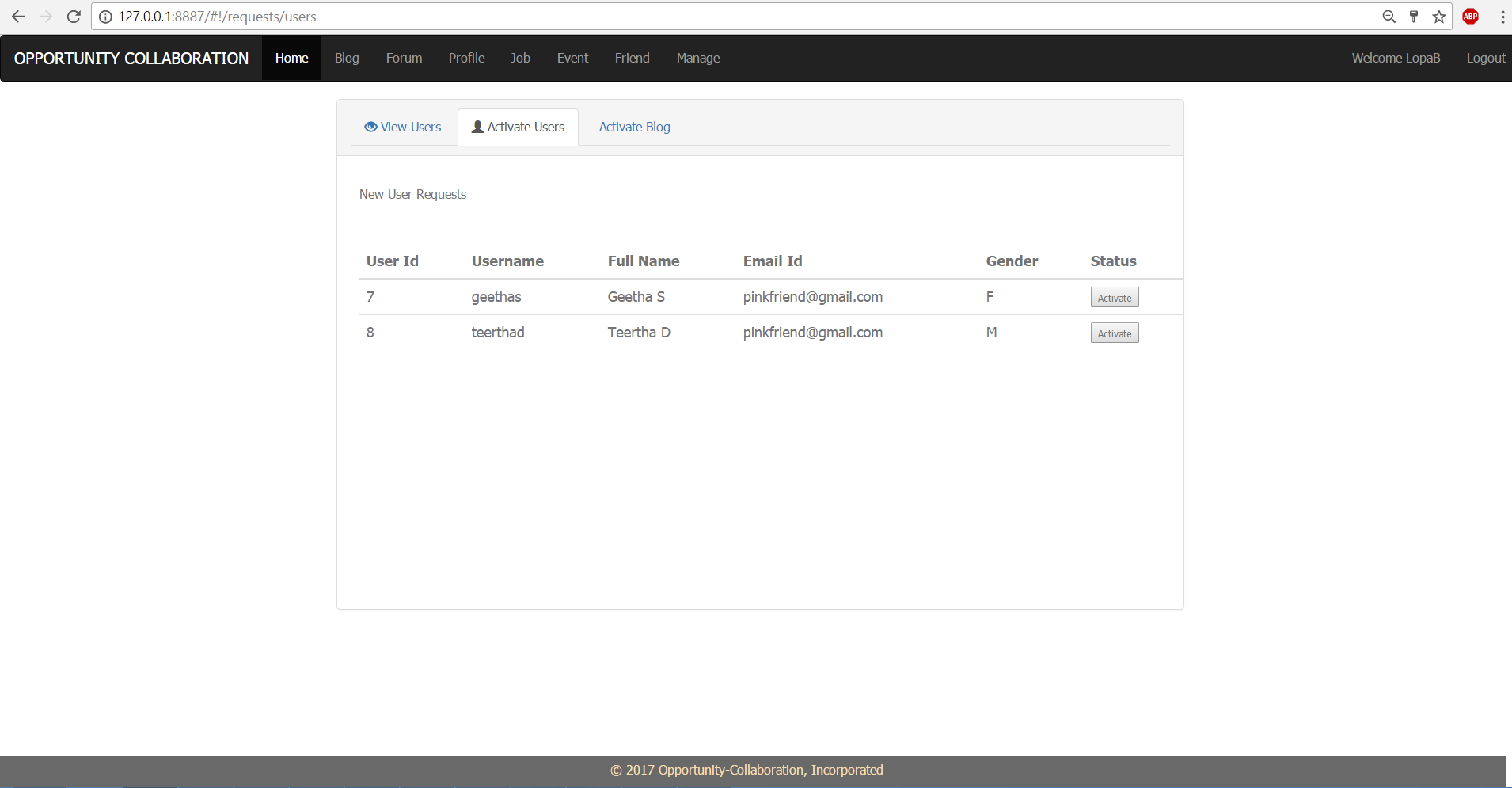
1. Modal is used to design this update form which is bind with Angular UserController to Angular UserService which calls API method for update user detail.
2. All activate users but not friend page



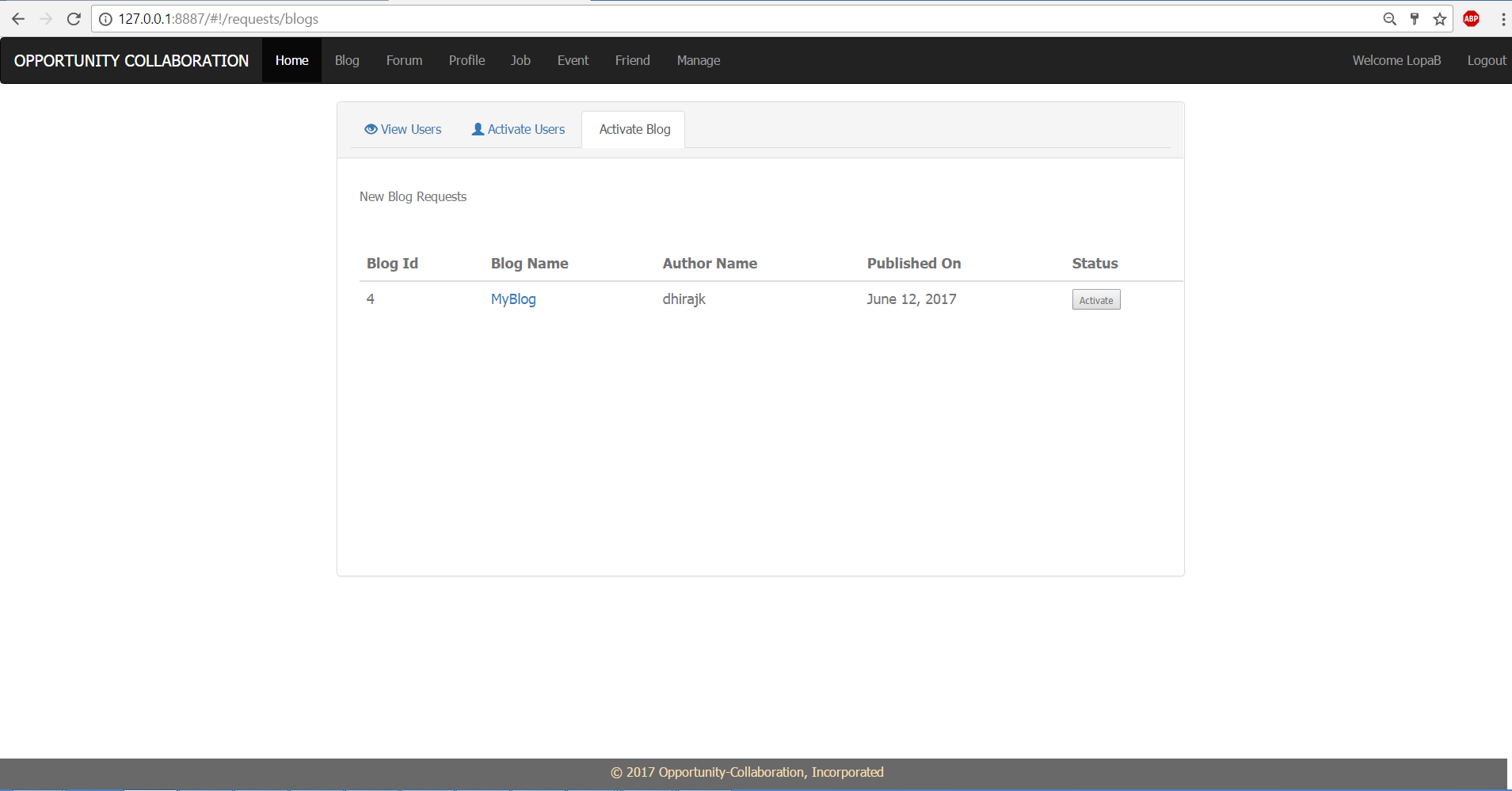
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular FriendController to Angular FriendService which calls API method for view approved users who can be friend.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(memberList.html).
4. User can add friend any user from this page.
5. View All Approved Users(ADMIN)



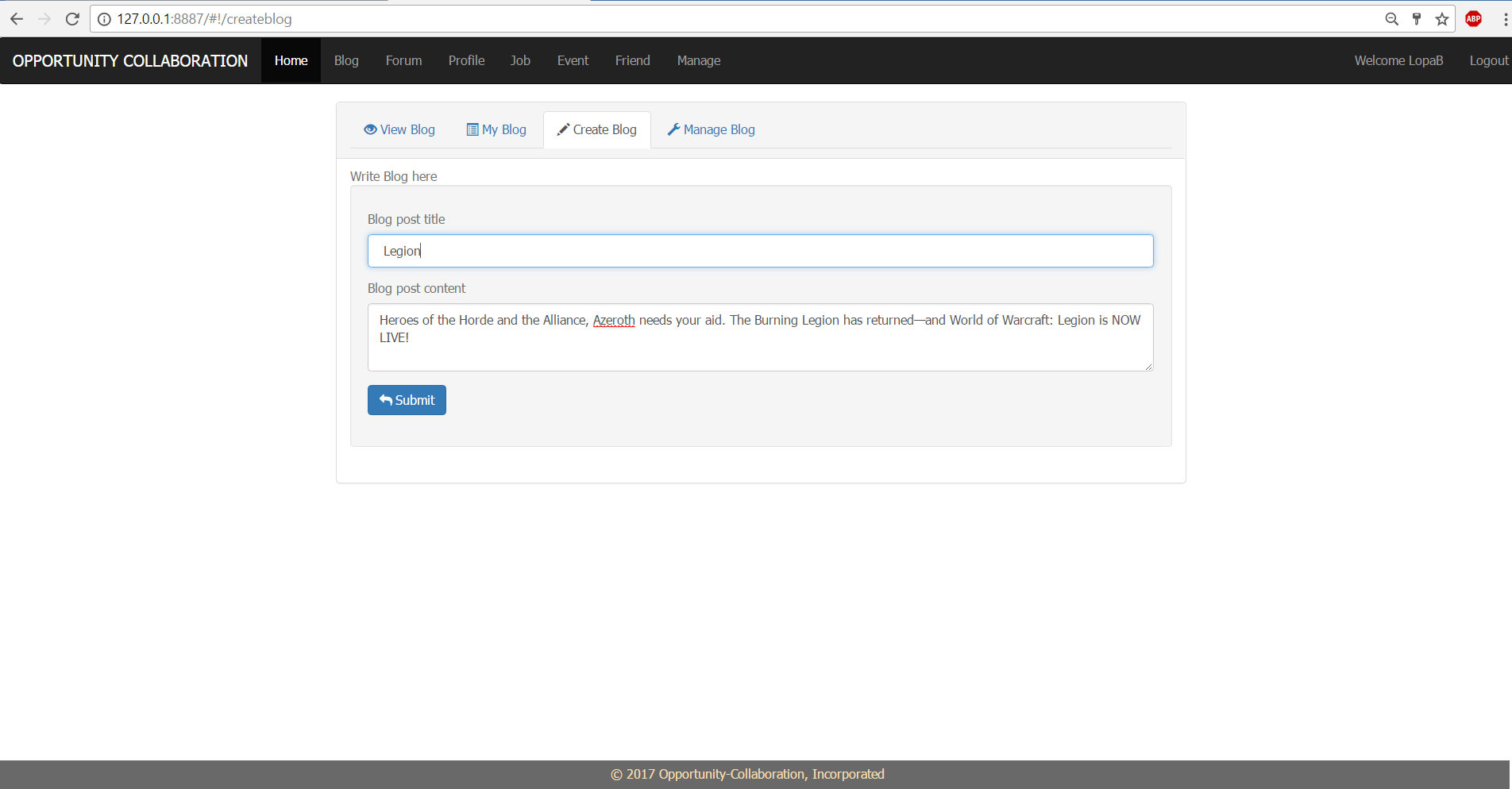
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UserController to Angular UserService which calls API method for view approved users by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(viewUsers.html).
4. User can add friend any user from this page.
5. User Request Approval Page(ADMIN)



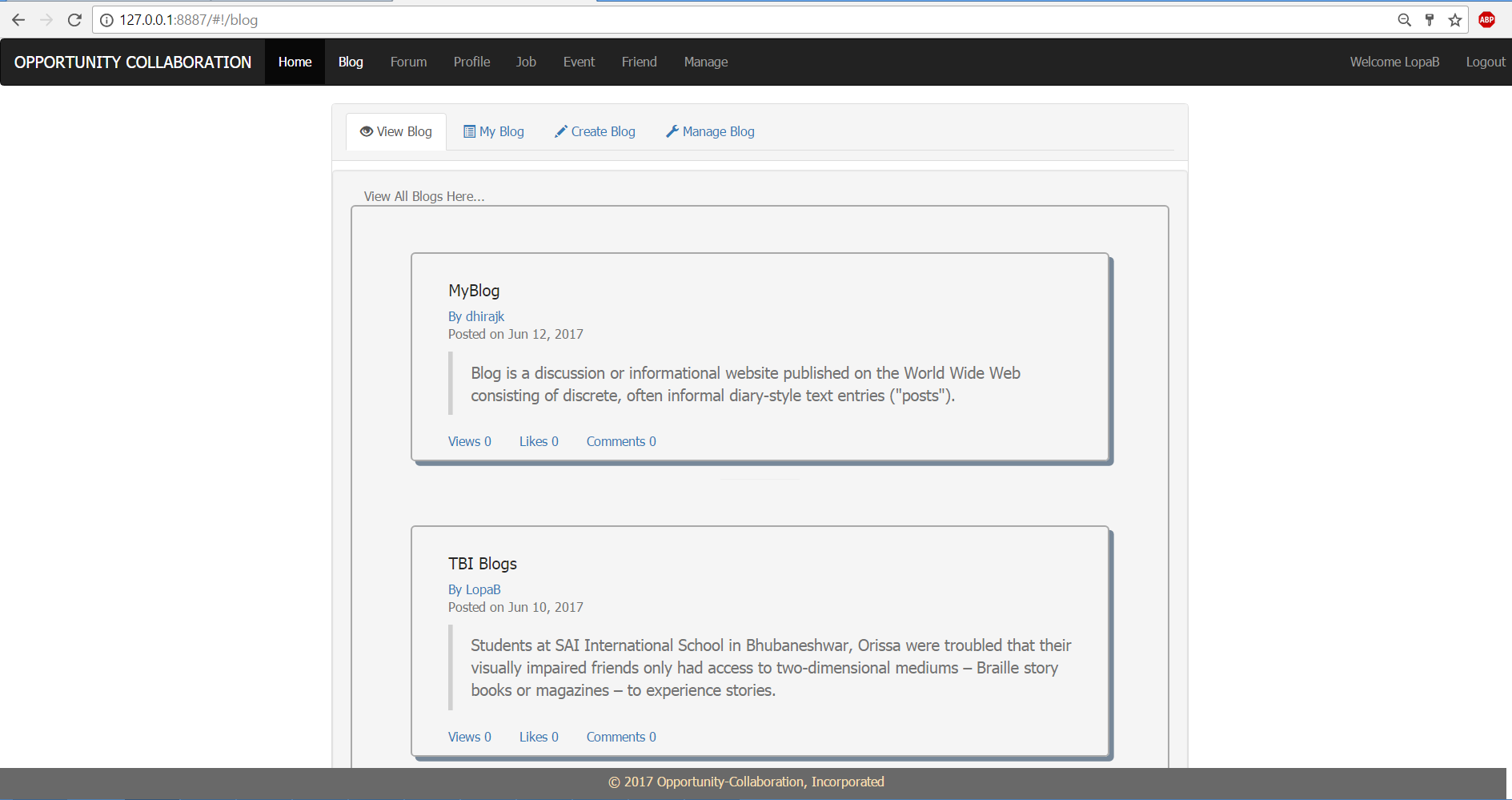
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular RequestController to Angular RequestService which calls API method for view to approve users by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(viewUsers.html).
4. Admin can approve user from this page.
5. Activate Blog by ADMIN



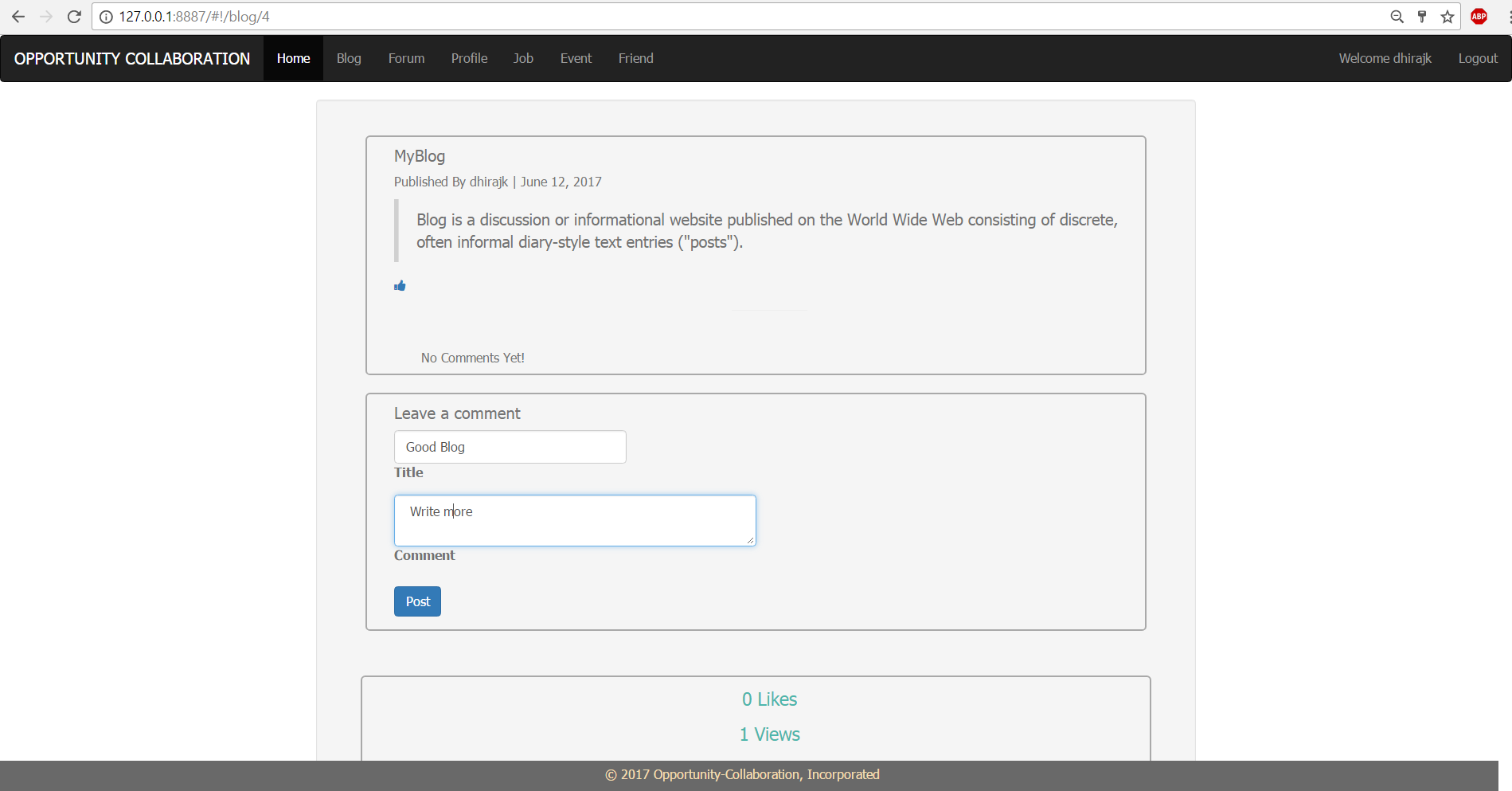
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular RequestController to Angular RequestService which calls API method for view to approve blog by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(activateBlog.html).
4. Admin can approve blogs from this page.
5. Create Blog



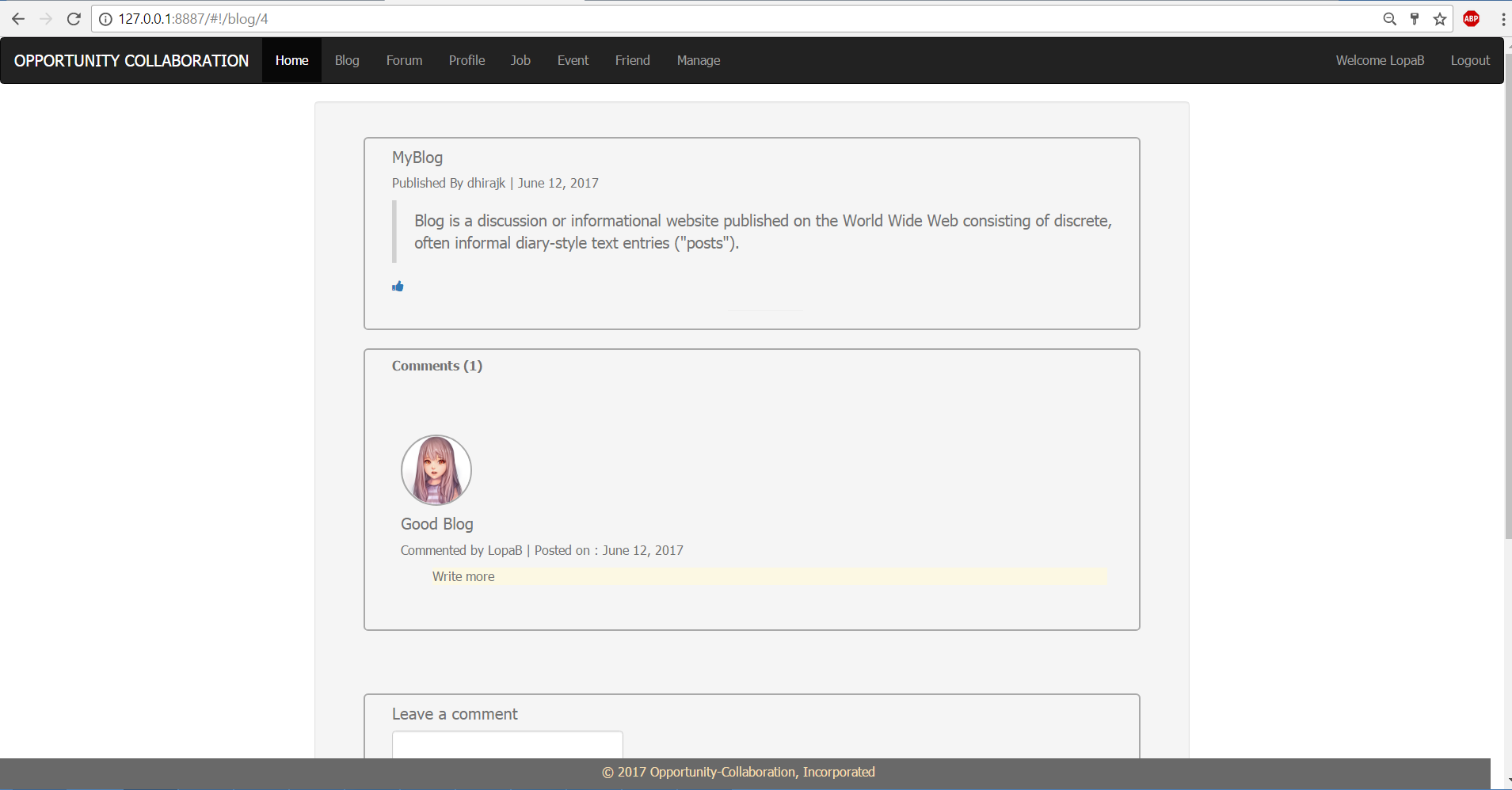
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular BlogController to Angular BlogService which calls API method for create blog.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(createBlog.html).
4. Any user can create blog here.
5. View All Blogs



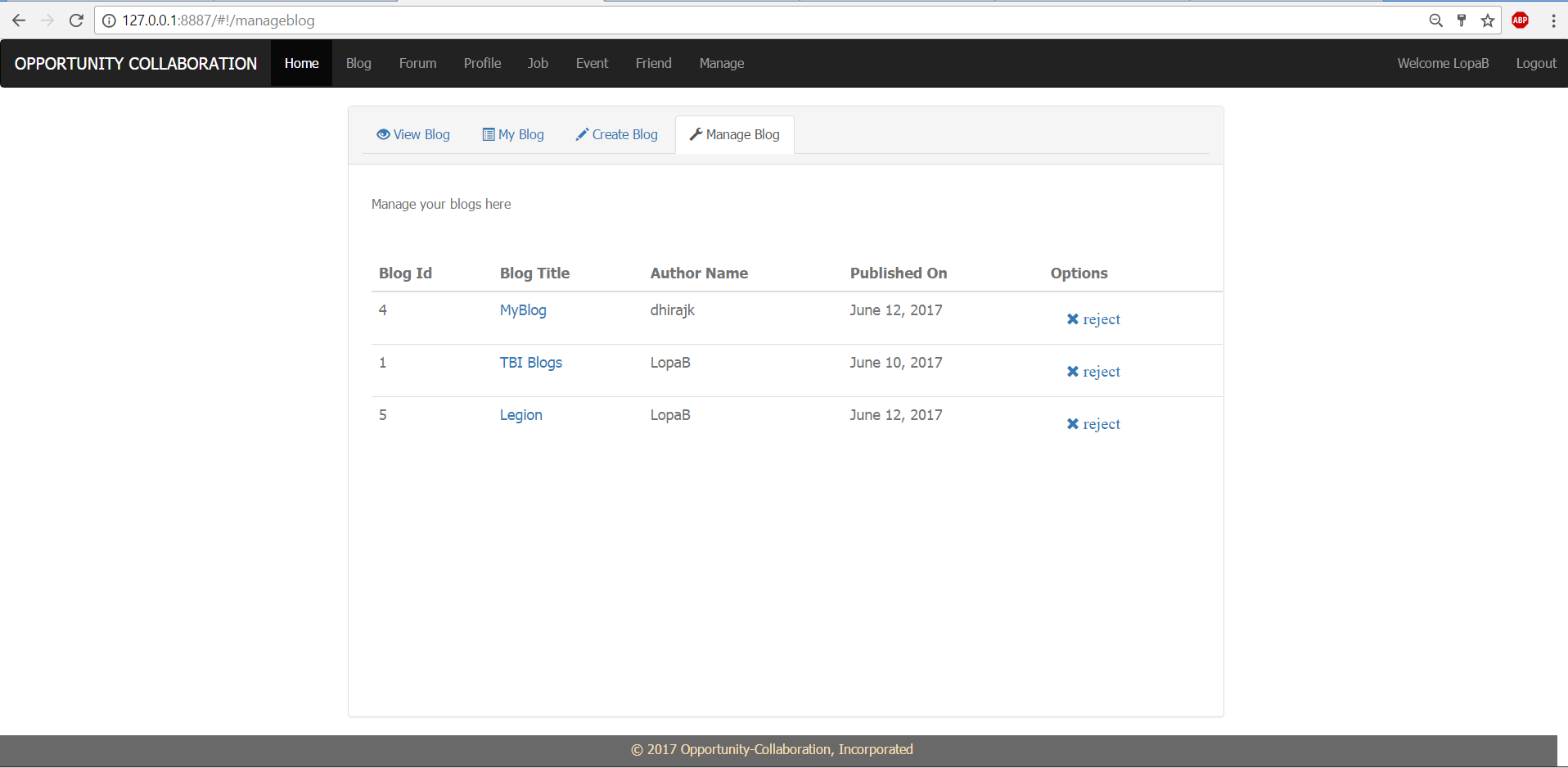
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular BlogController to Angular BlogService which calls API method for view to view all approved blogs.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(viewBlog.html).
4. Any user can view blog here. And can redirect to the blog detail page.
5. User’s Blog page



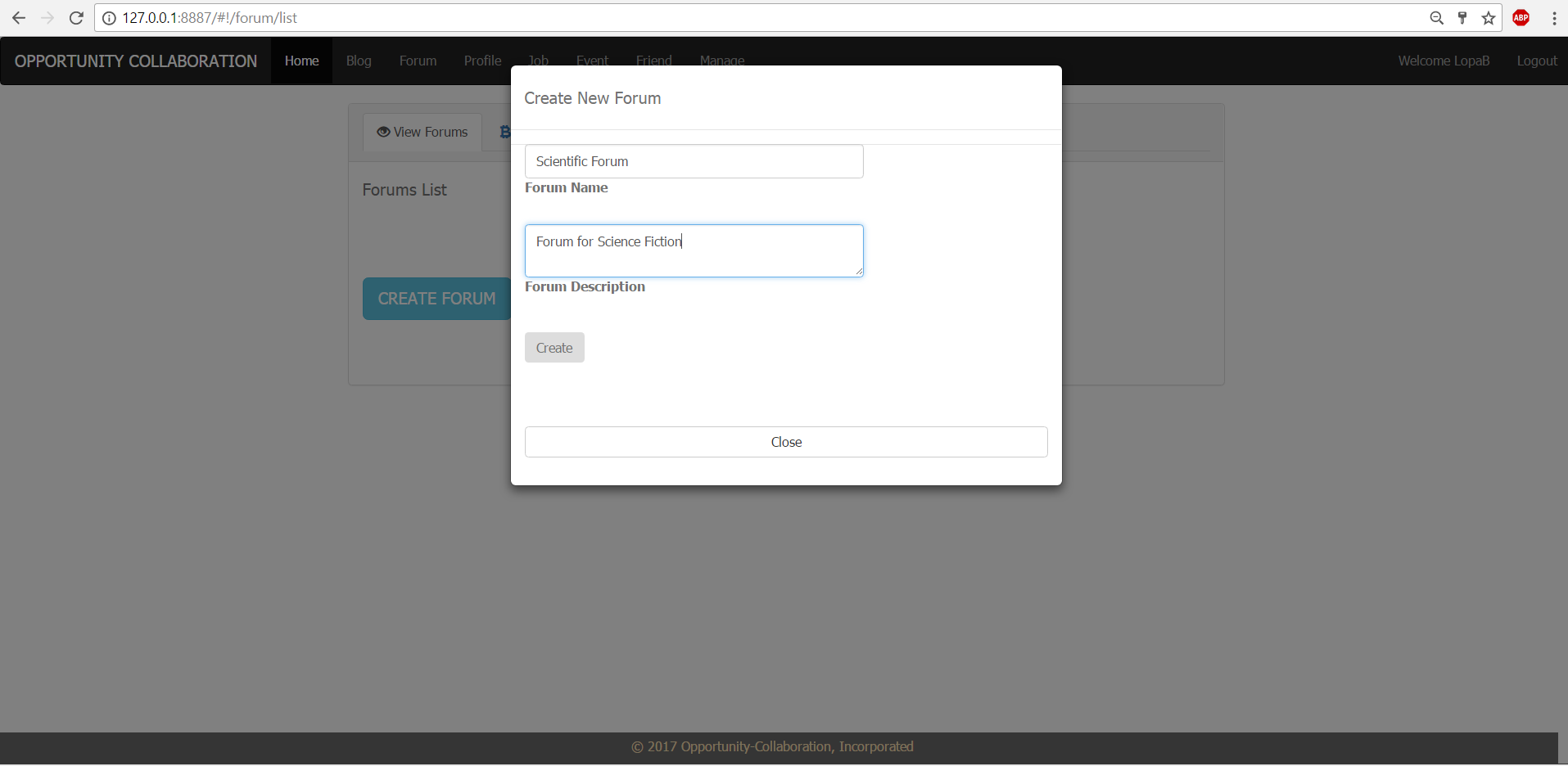
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular BlogController to Angular BlogService which calls API method for view to approve blog by ADMIN and USER.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(singleBlog.html).
4. Any user can view blog detail, post comments, like the blog.
5. Blog detail page after comment post



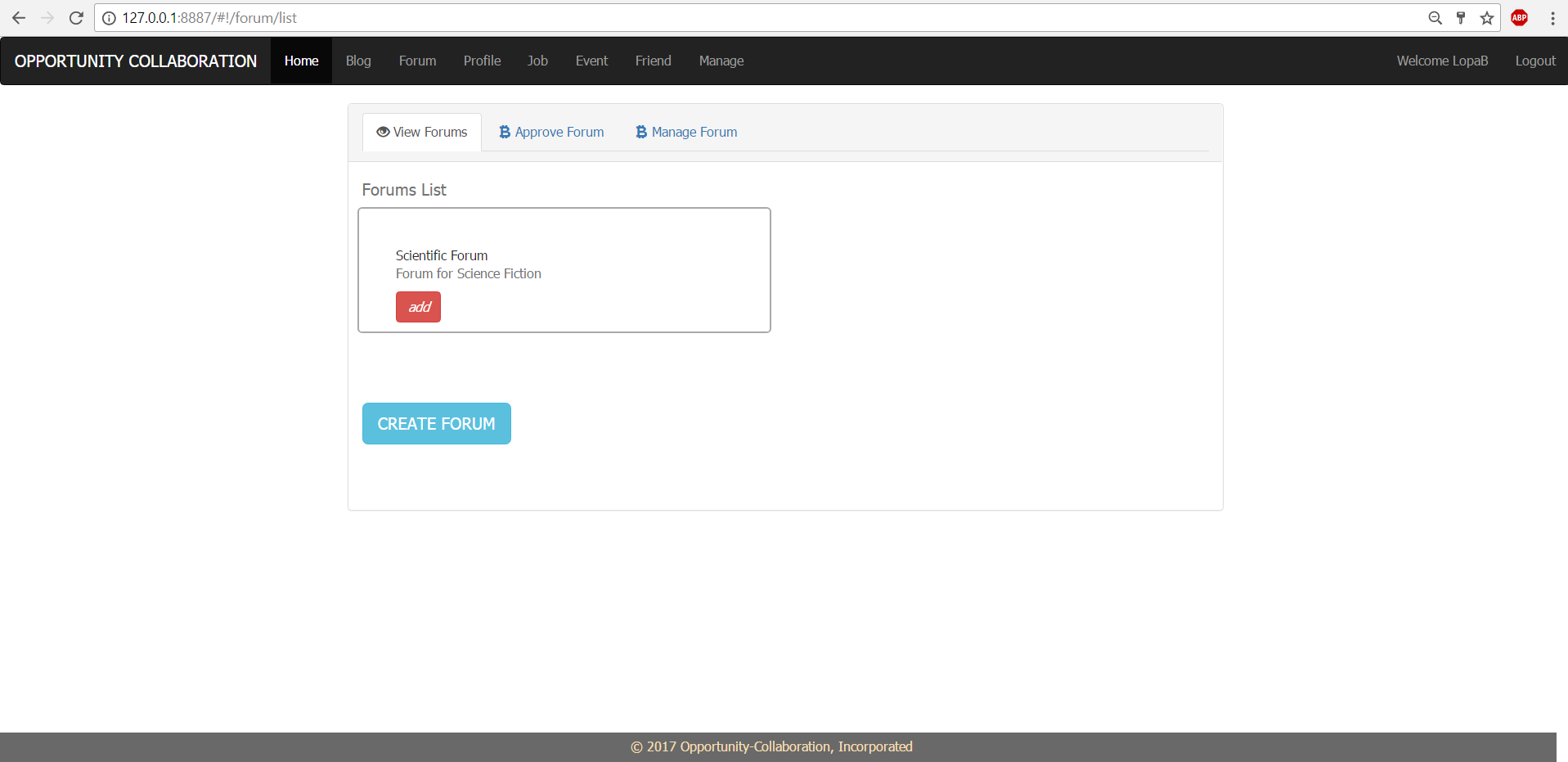
1. Manage Blog by Admin



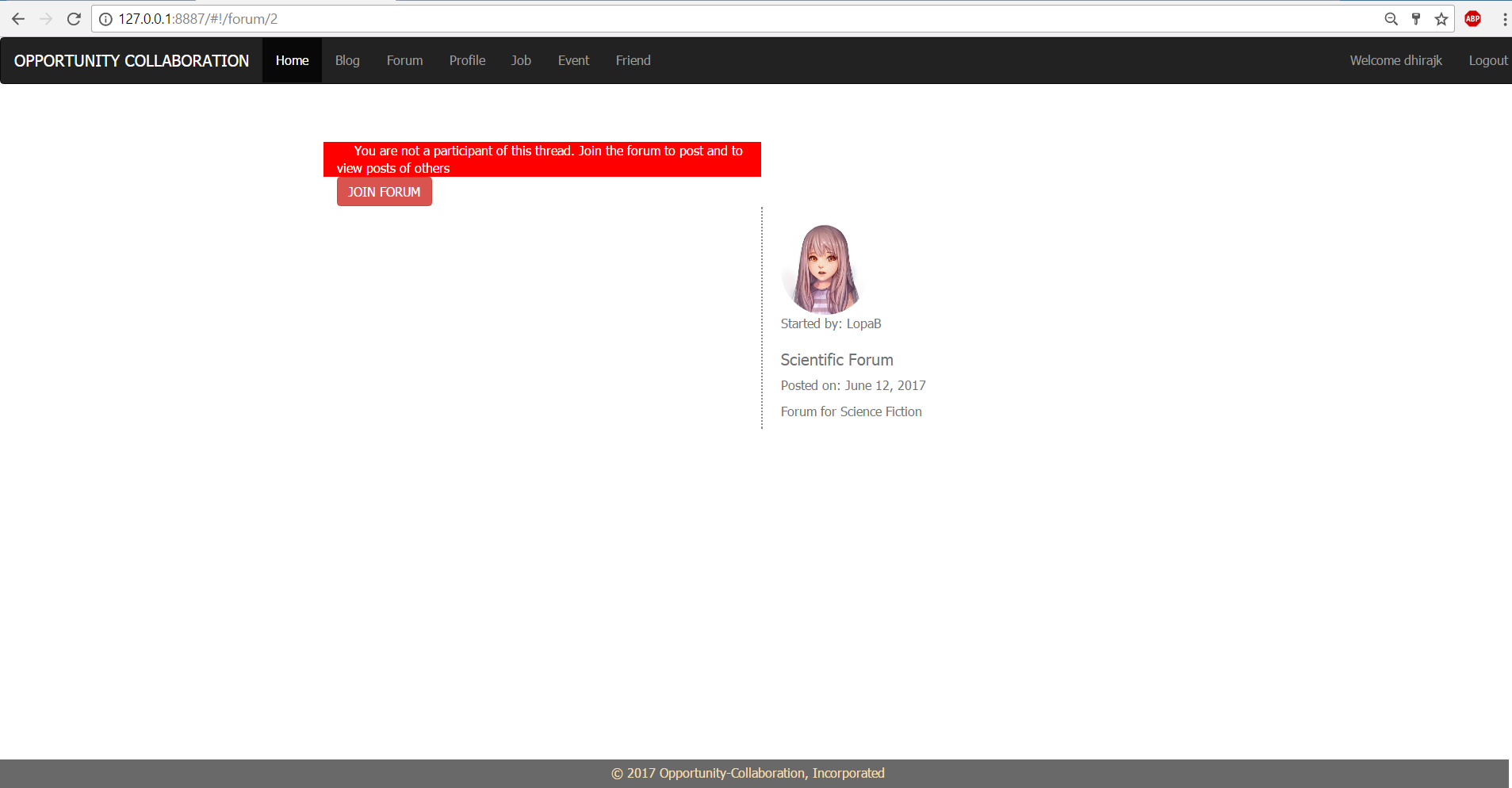
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular AdminController to Angular AdminService which calls API method for view to approve blogs by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(manageBlog.html).
4. ADMIN can view all approved blogs for rejection.
5. Create Forum by ADMIN



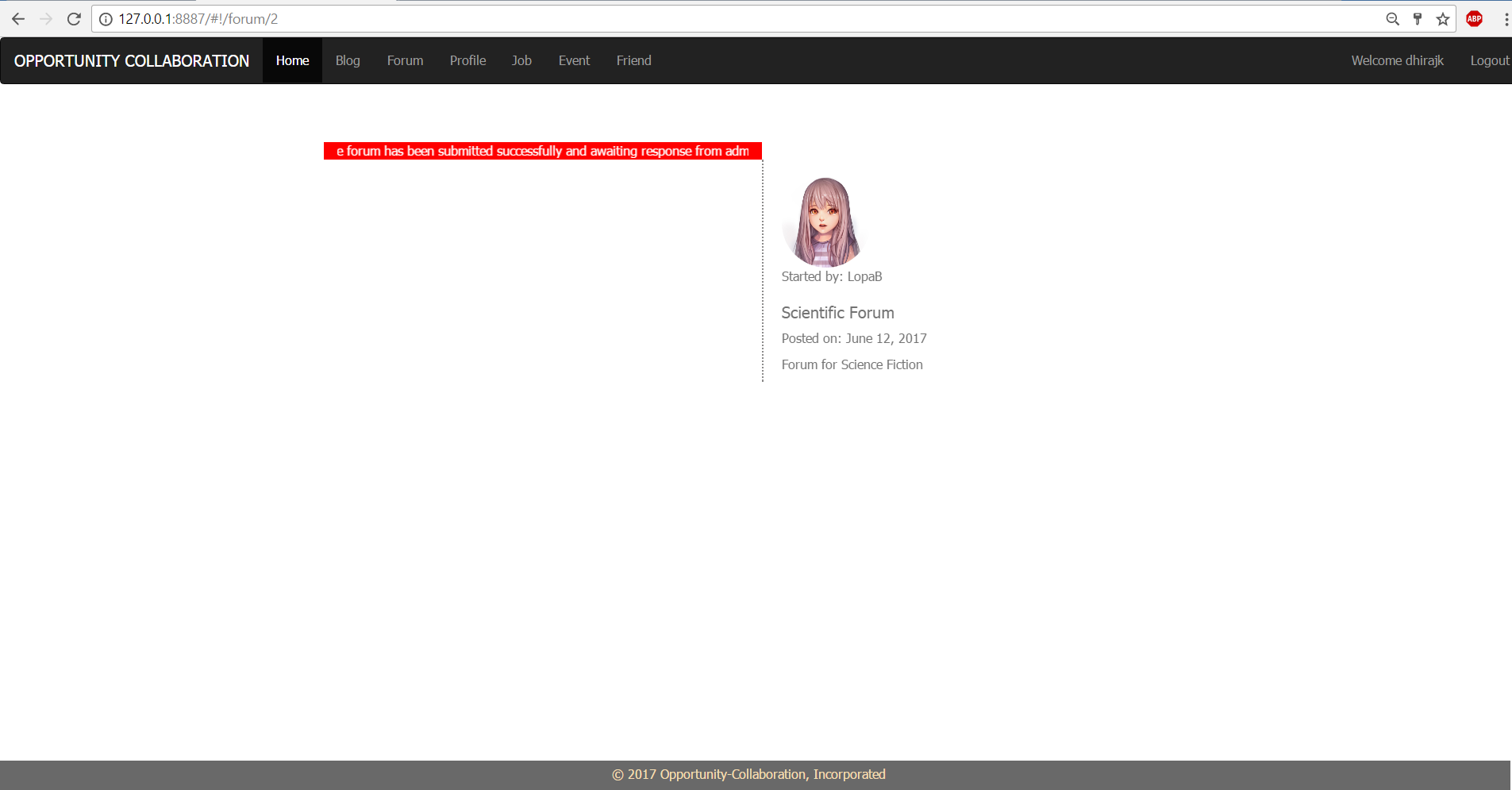
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular ForumController to Angular ForumService which calls API method for create forum by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(forumlist.html).
4. ADMIN can create forum here.
5. Forum List page



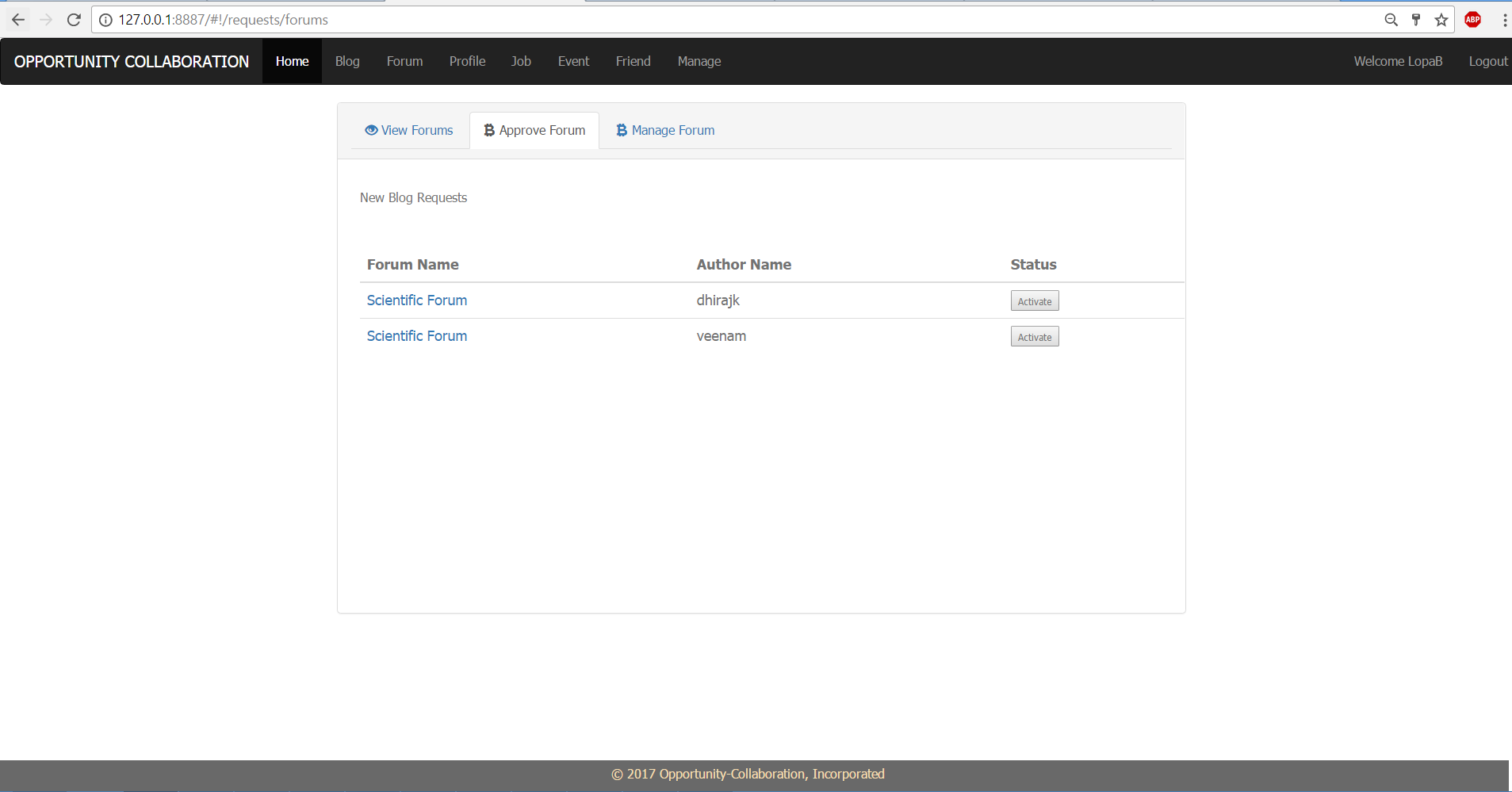
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular ForumController to Angular ForumService which calls API method for view to view all approved forums.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(forumlist.html).
4. Any user can view approved forums here. And can redirect to the forum detail page to post.
5. Forum page before join



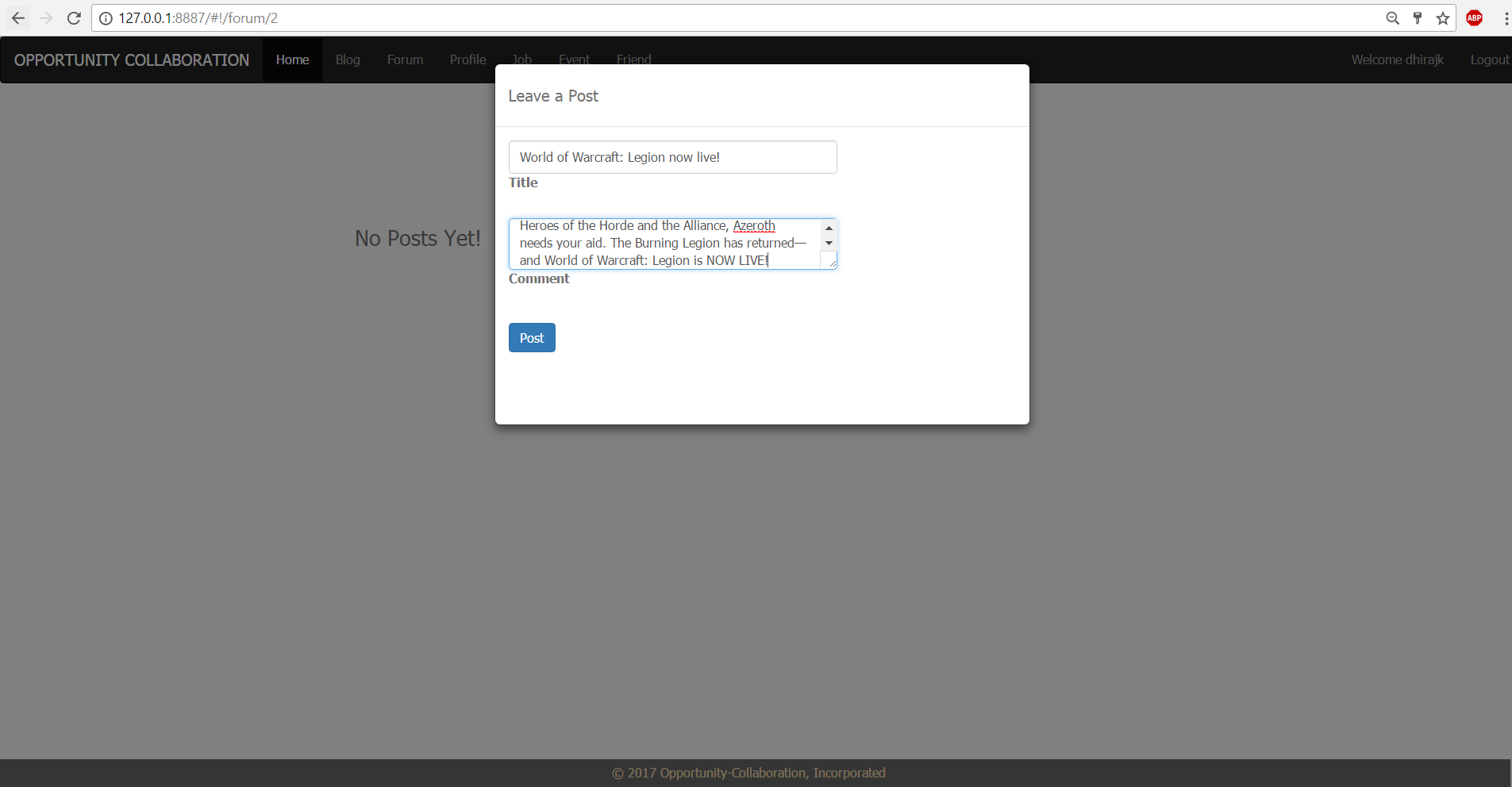
1. After Join, before approve by Admin



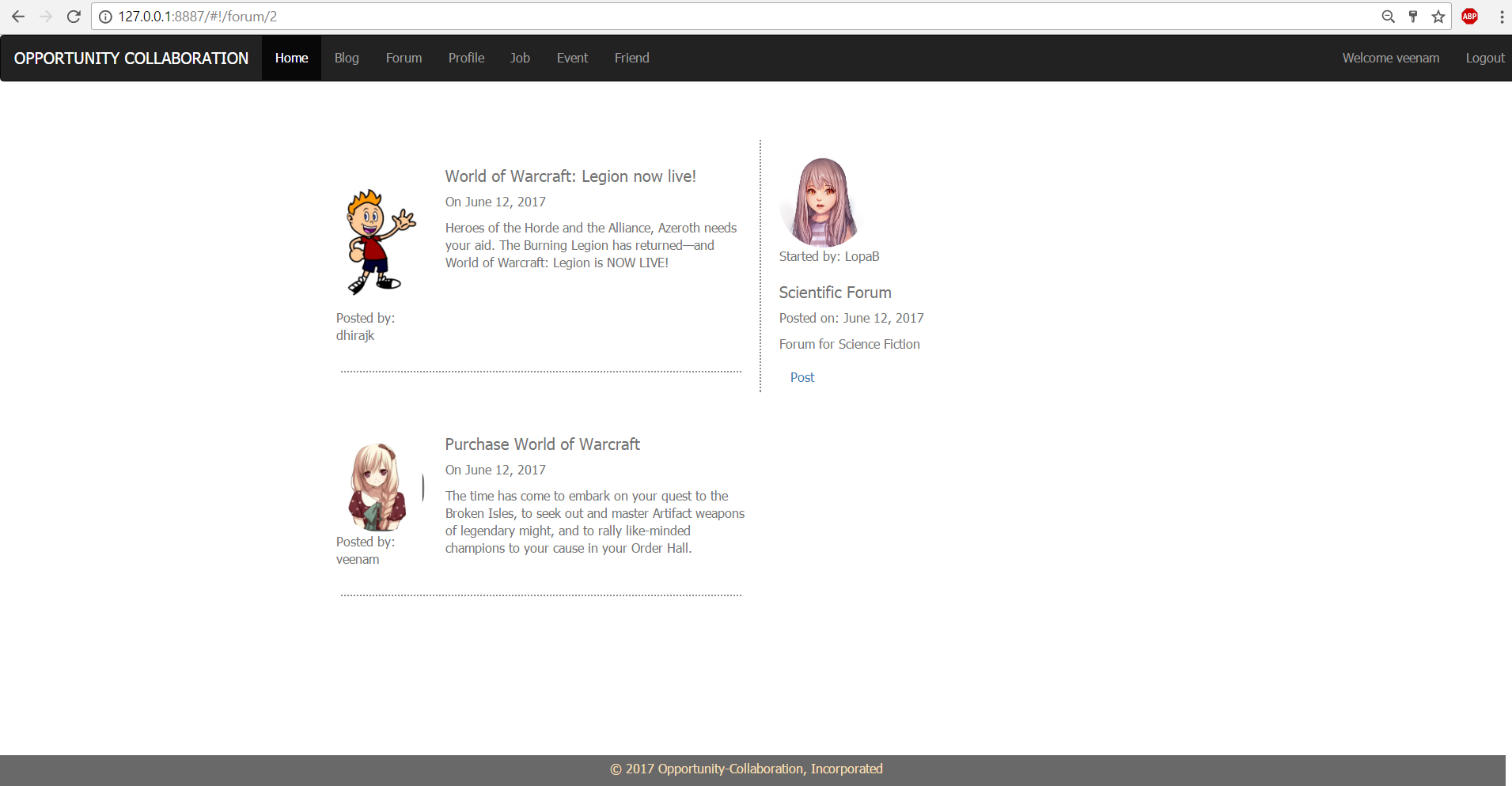
1. Admin Approval page for Forum



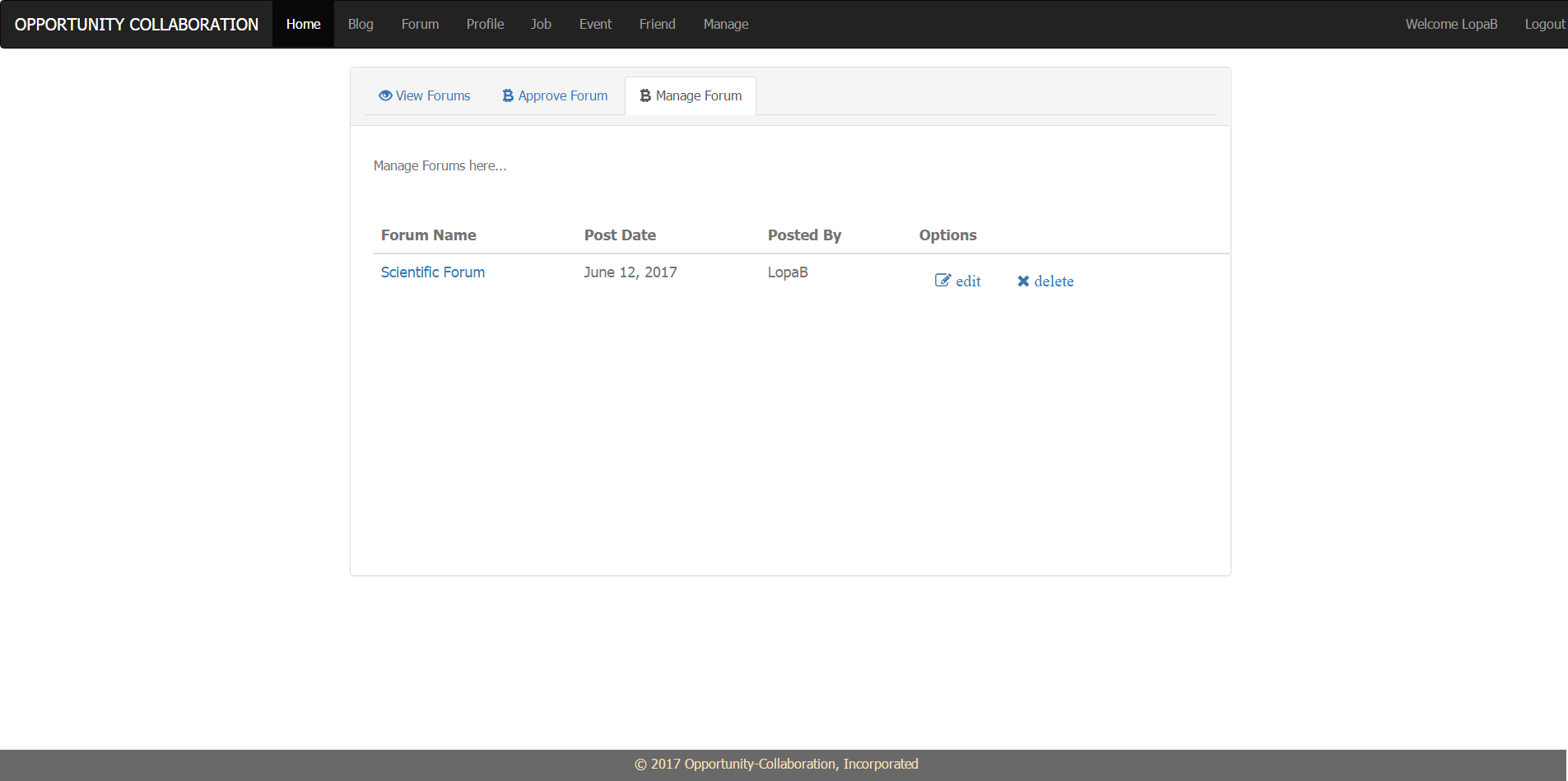
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular RequestController to Angular RequestService which calls API method for view to approve forums by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(forumRequest.html).
4. Admin can approve forums for all users to avail from this page.
5. Forum Post by user



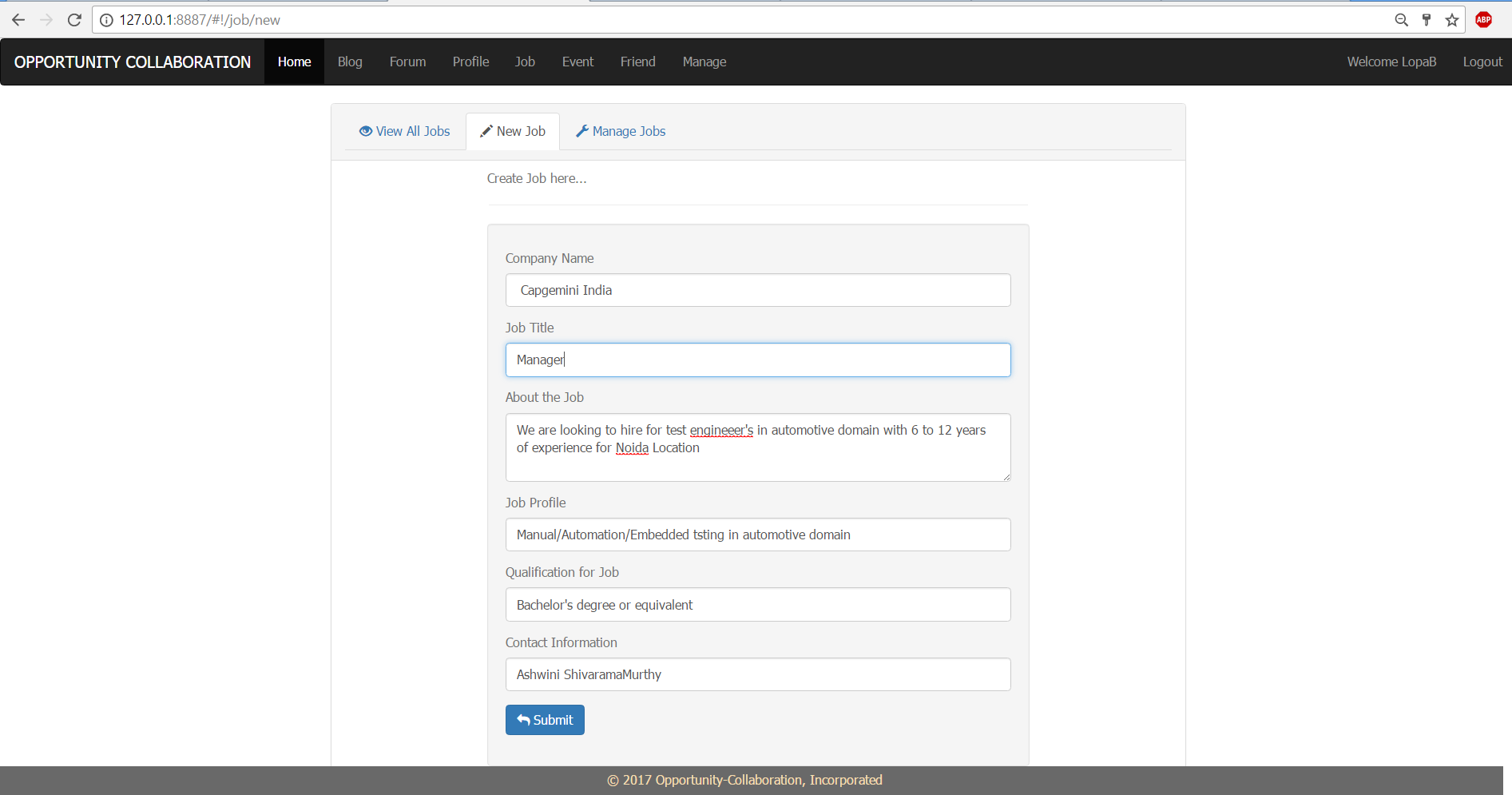
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular ForumController to Angular ForumService which calls API method for view to approve blog by ADMIN and USER.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(forum.html).
4. Any user can post for forum.
5. Forum page after post



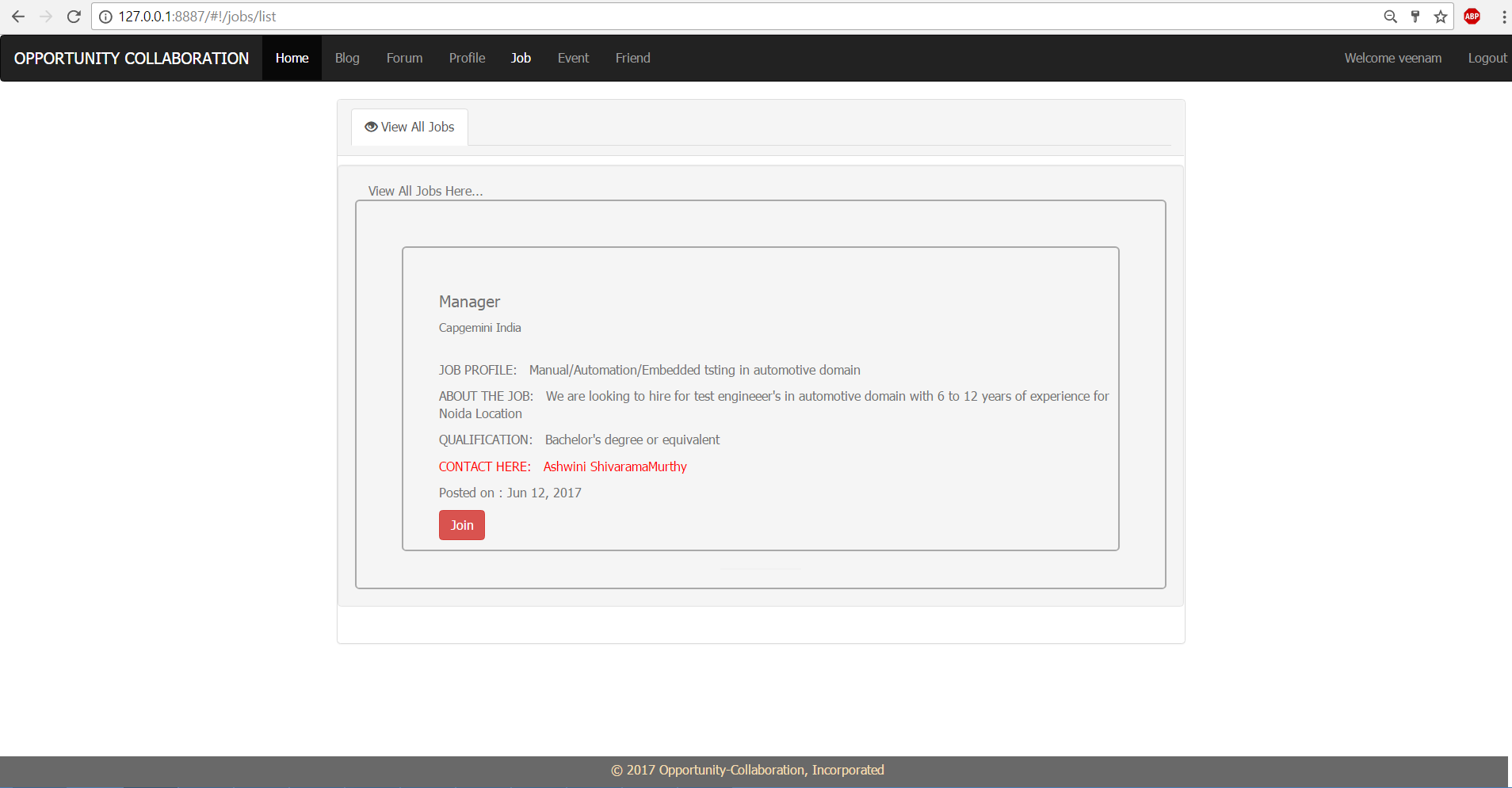
1. Manage Forum Page by ADMIN



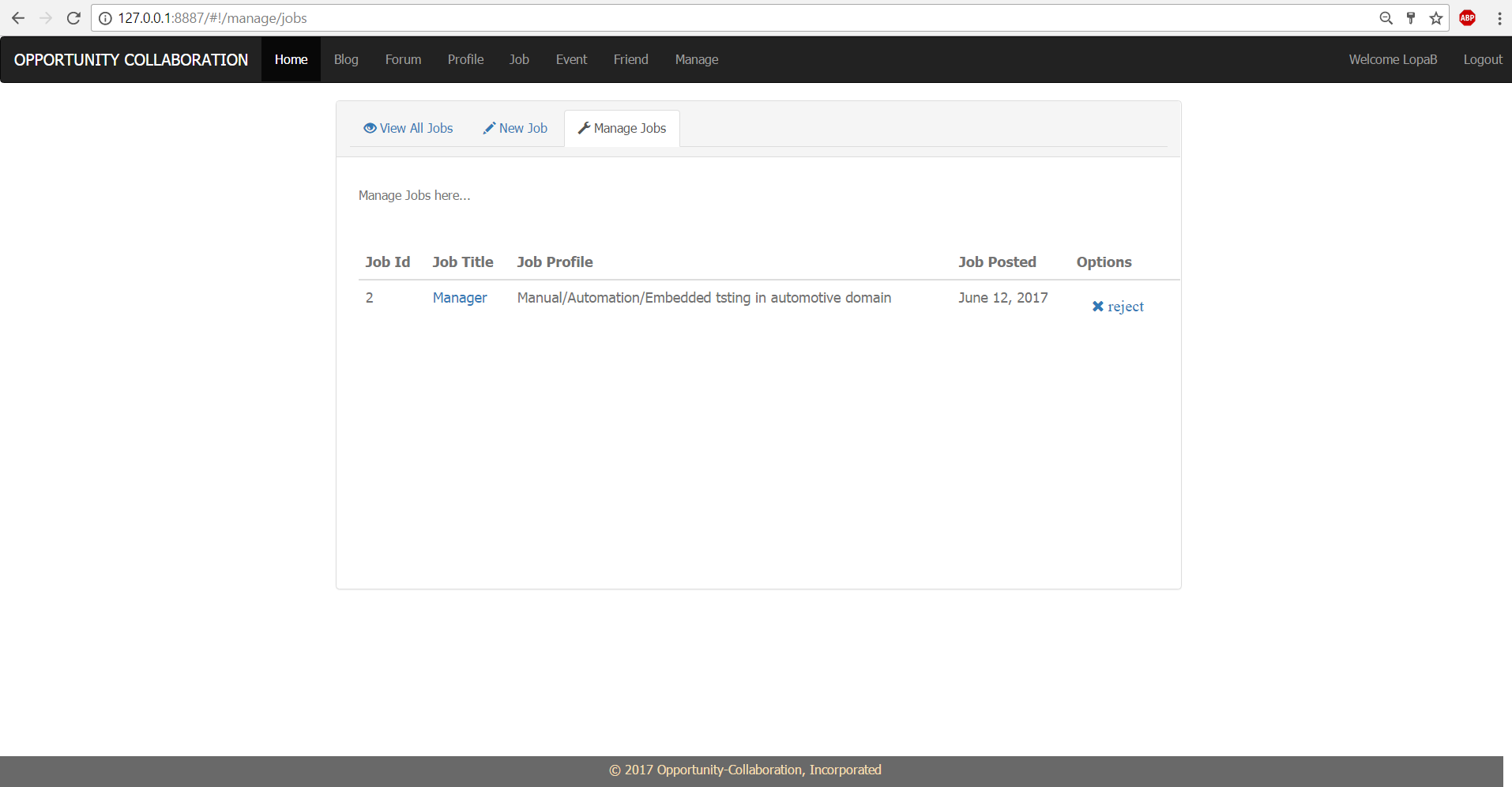
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular ForumController to Angular ForumService which calls API method for view to delete and update forums by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(manageForum.html).
4. ADMIN can view all approved forums for update and delete.
5. Create new job by admin



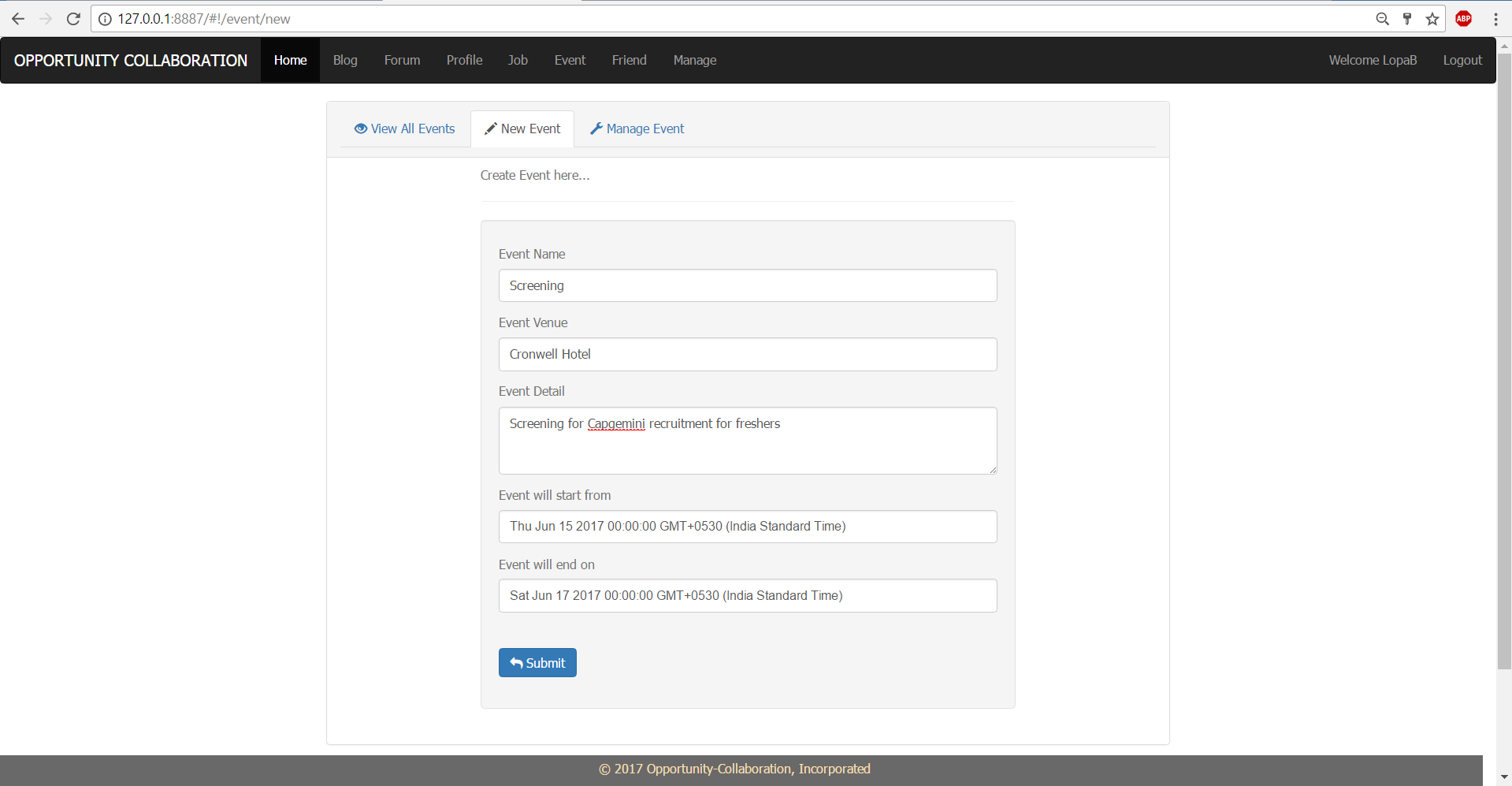
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular JobController to Angular JobService which calls API method for create job by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(newJob.html).
4. ADMIN can create job here.
5. View Job page



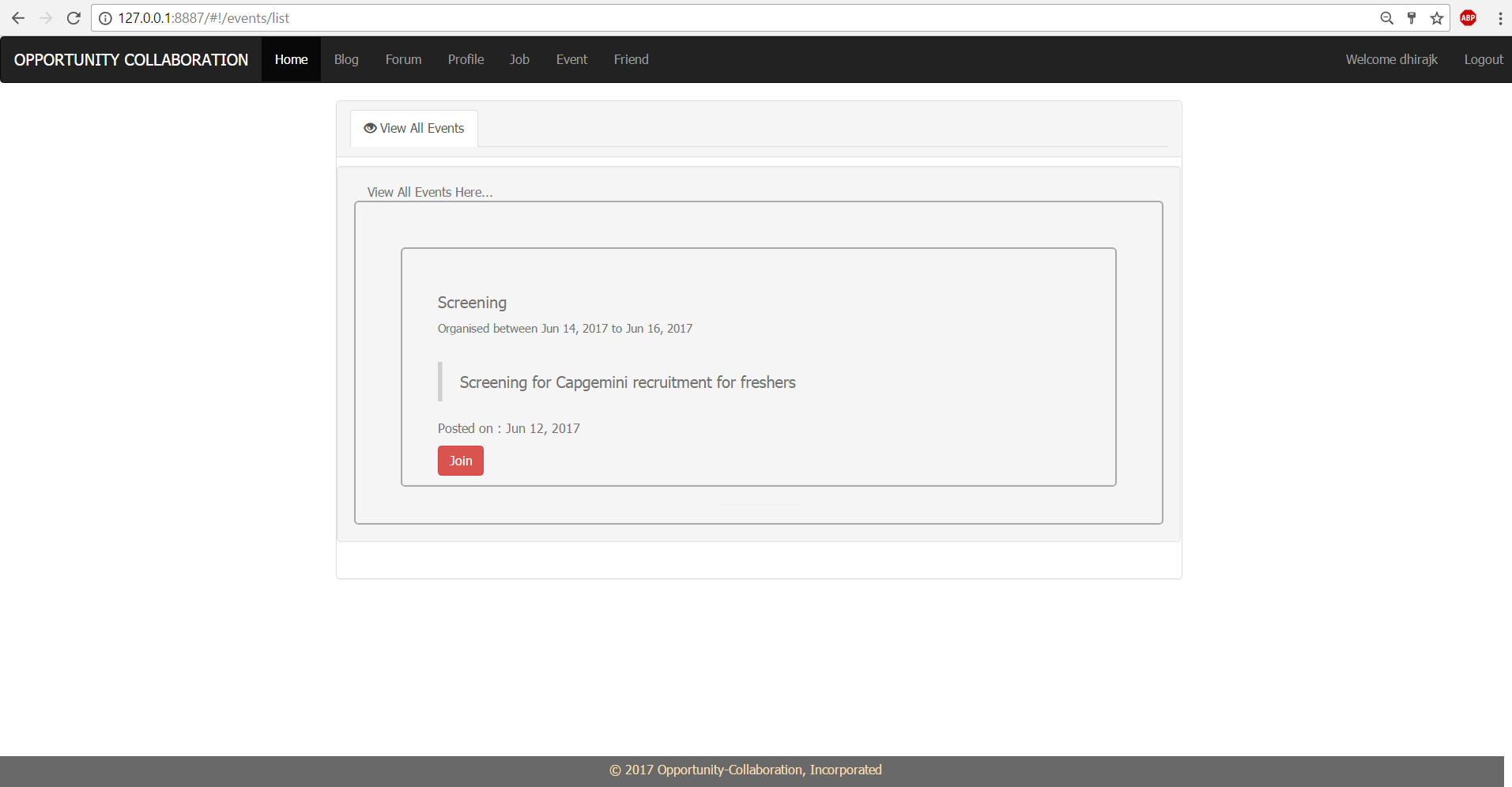
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular JobController to Angular JobService which calls API method for view to view all approved forums.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(joblist.html).
4. Any user can view jobs here. And can apply for job.
5. Job Manage by Admin



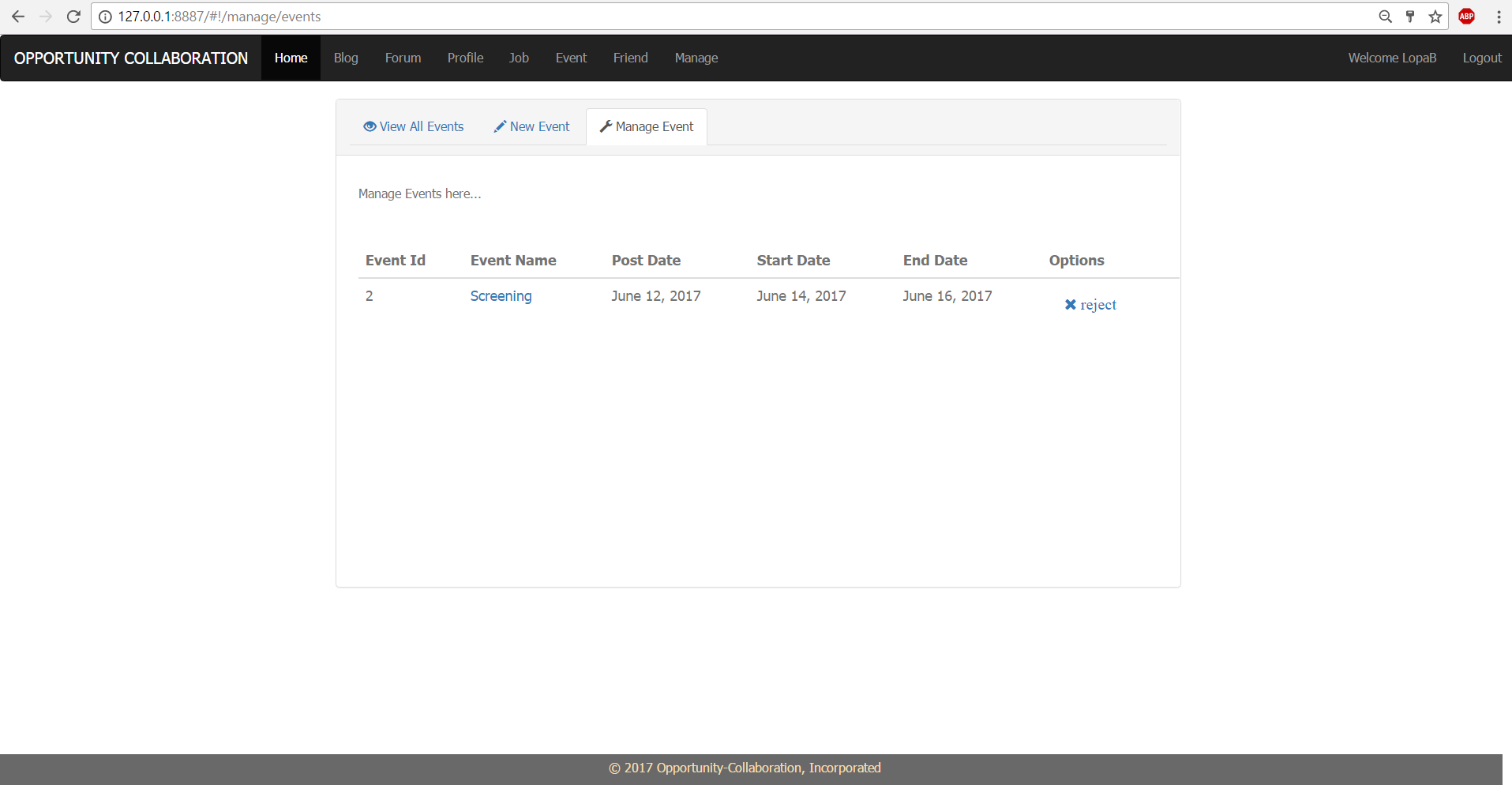
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular AdminController to Angular AdminService which calls API method for view to jobs by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(manageJob.html).
4. ADMIN can view all approved jobs for rejection.
5. Create new Event by Admin



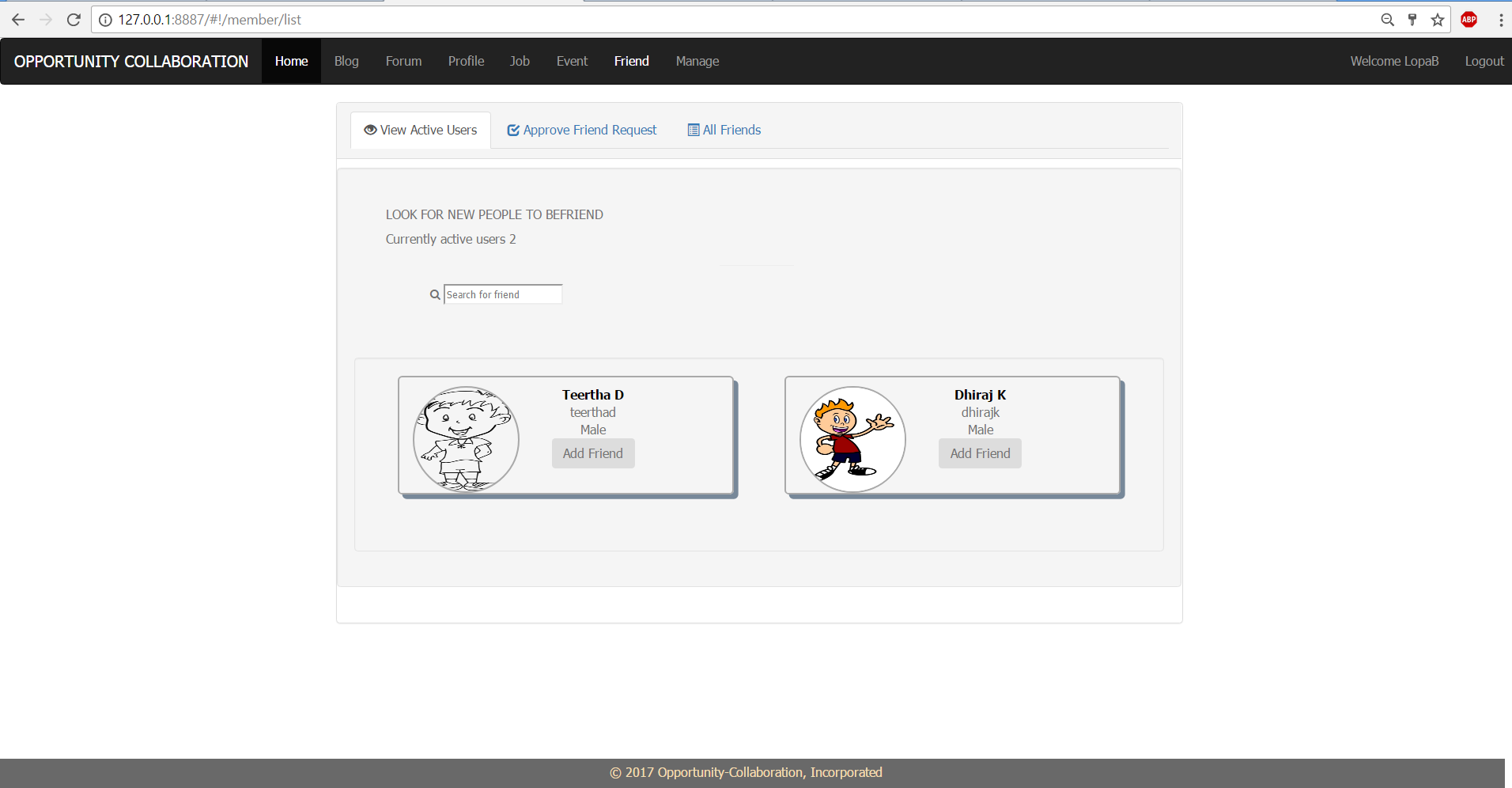
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular EventController to Angular EventService which calls API method for create event by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(newevent.html).
4. ADMIN can create event here.
5. View all events and Join



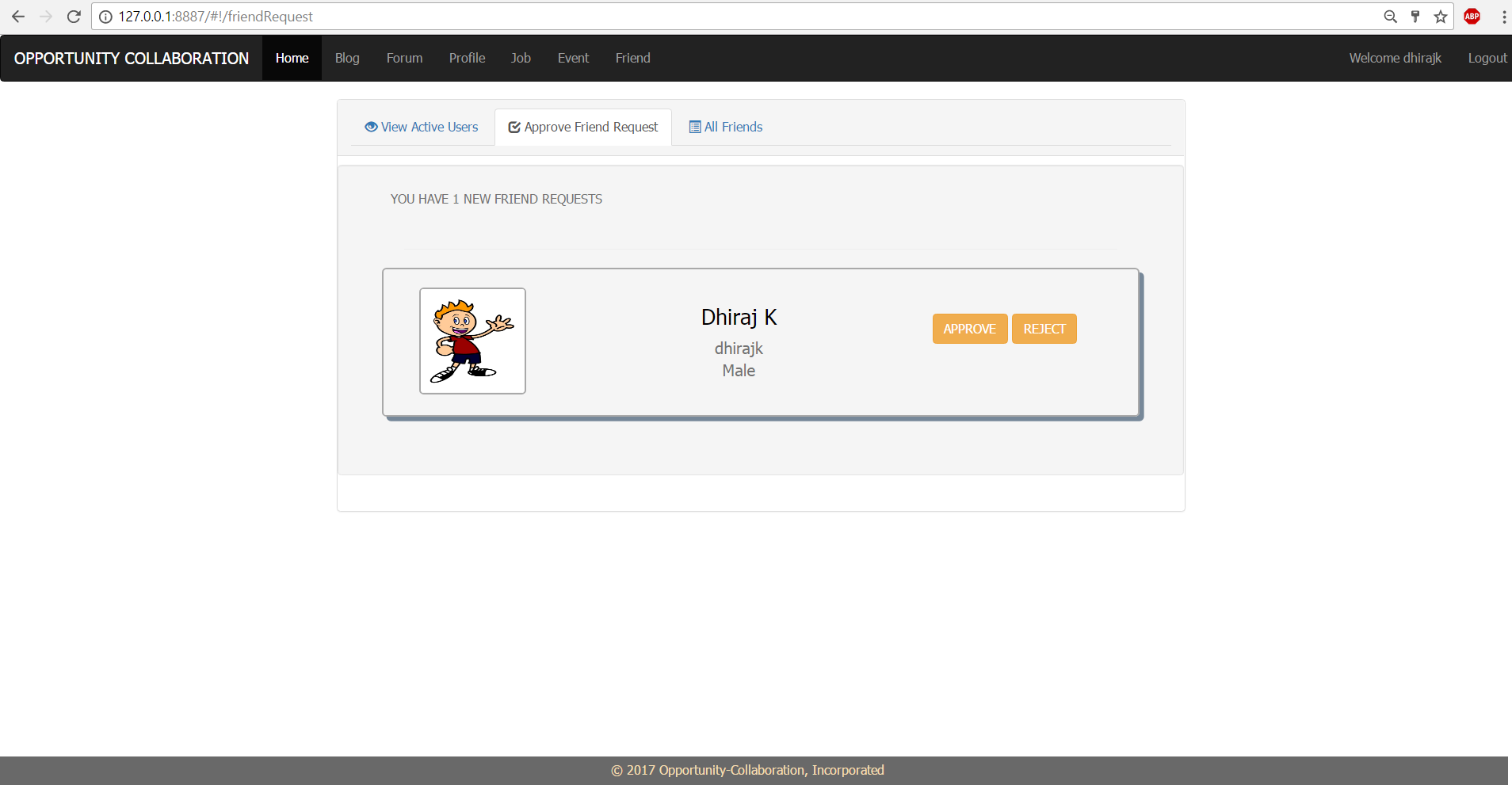
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular EventController to Angular EventService which calls API method for view to view all events.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(eventlist.html).
4. Any user can view events here. And can join request for event.
5. Manage Event by ADMIN



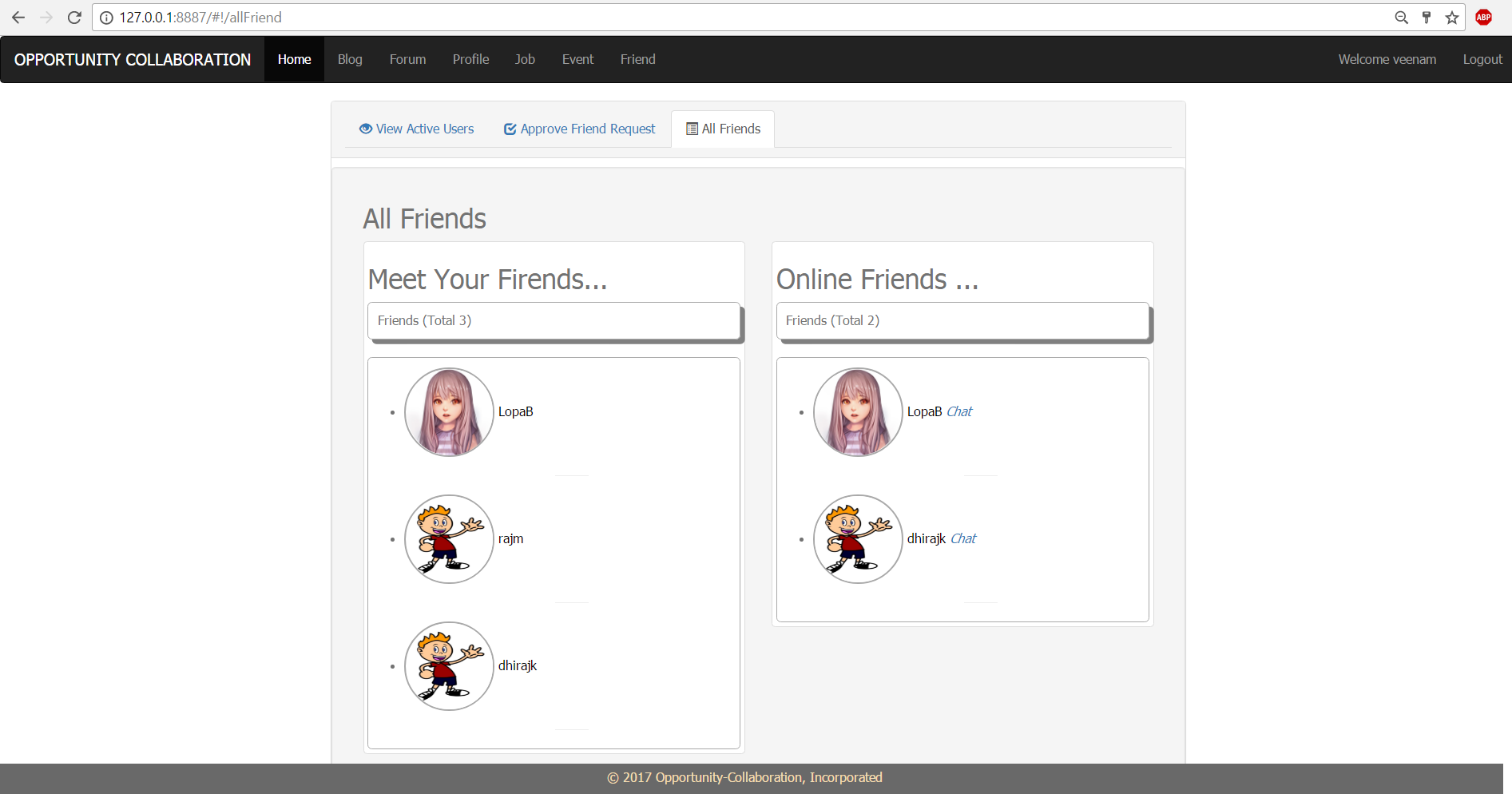
1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular AdminController to Angular AdminService which calls API method for view to manage events by ADMIN.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(manageEvent.html).
4. ADMIN can view all events for rejection.
5. To be friend page



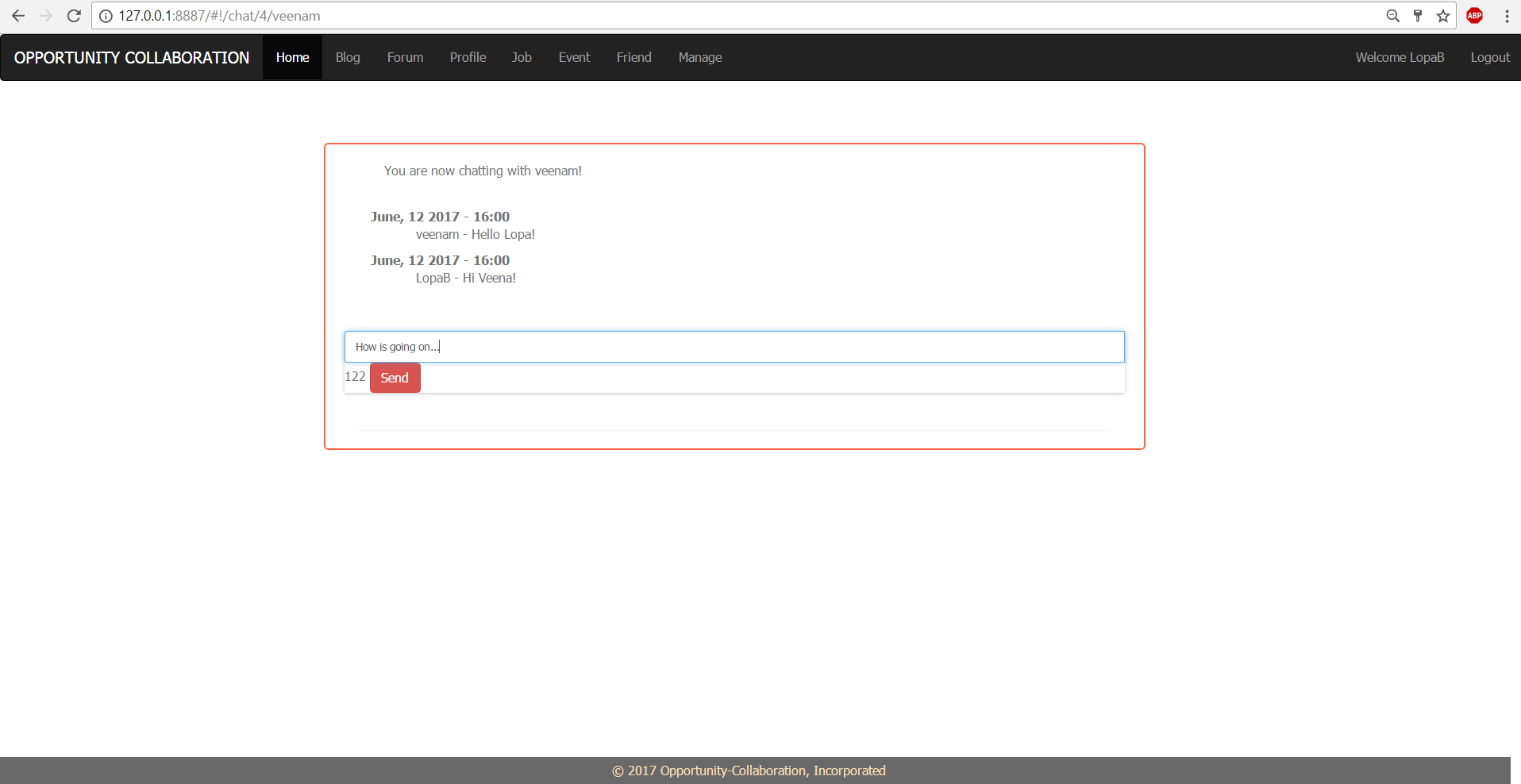
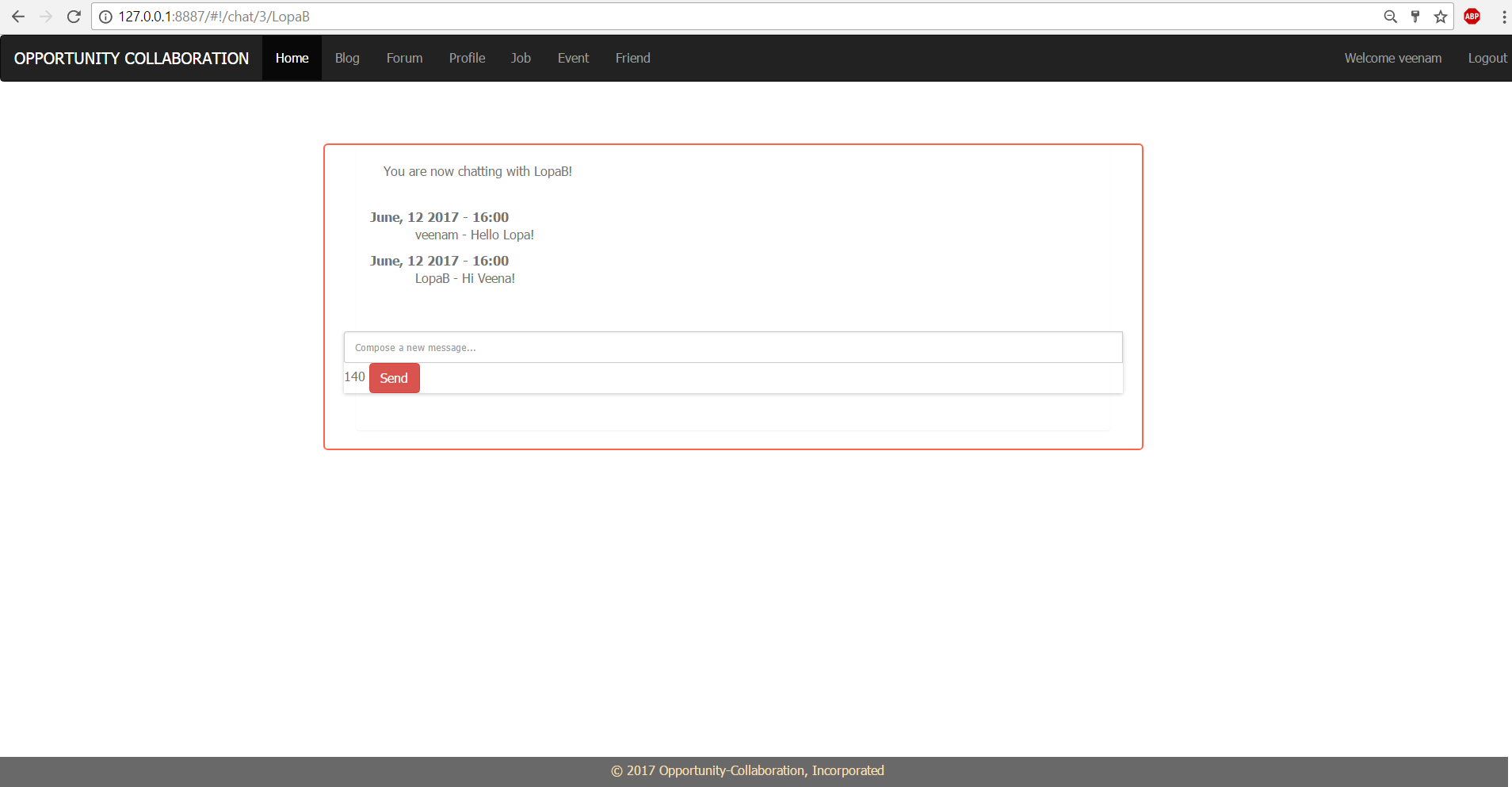
1. Friend Request



1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular FriendController to Angular FriendService which calls API method for view all friend requests.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(friendRequest.html).
4. Any user can view all friend request. User can Accept/Reject friend request from this page.
5. Friends Page



1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular UserController to Angular UserService which calls API method for view all friends with friends who are online.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(friendRequest.html).
4. Any user can view all friend and further can start chat with online friends.
5. Chat



1. Technologies used to design the page are HTML, CSS, Bootstrap, JavaScript, AngularJS.
2. This page is bind with Angular ChatController to Angular ChatService which calls API method for chat with friends.
3. This page is built up on navigation bar(navbar.html) , footer(footer.html) and content page(chat.html).
4. Any user can do chat with friend.
5. Used STOMP and SOCKJS for angular interface asynchronous message subscribing and publishing.
6. Used Websocket at backend to enable broker and service.
7. Used spring-websocket, spring-messaging, javax.websocket-api dependencies in backend project to perform asynchronous messaging.

Conclusion

It was a wonderful and learning experience for me while working on this project. This project took me through the various phases of project development, implement known and unknown technologies and gave me real insight into the world of software engineering. The joy of working and thrill involved while tackling the various jargons and challenges gave me a feel of developers industry.

It was due to this project I came to know how professional software’s are designed and developed. I am confident enough to handle student’s query during their practices. I will every time make myself up to date to guide students.

I enjoyed each and every bit of work I had put into this project. This project is further extendable.

==============================================================================