APPOINTMENT APP

# Project Details

This project helps to solve the problem of creating appointments with coop advisors for the students.

Its a user-friendly application, where students can search for their appointments using different search parameters, also this app allows to update and delete the appointments made by students.

# GitHub URL

# <https://github.com/AjimolDas/EnterpriseProject>

# Team Members:

|  |  |
| --- | --- |
| Student ID | Student Name |
| C0745042 | AJIMOL DAS |

# NoSQL Database Information

Database Name: salsesmongo

|  |  |  |
| --- | --- | --- |
| Collection Name | Collection Details and #of Documents available | Team Member(s) worked on it |
| Books | Details of Books in library.  #8 | Ajimol Das |
| Student | Details of student in college.  #8 | Ajimol Das |
| Instructors | Details of instructors in college.  #8 | Ajimol Das |
| Post | Details of post in college.  #3 | Ajimol Das |
| Customers | Details of customers in a store.  #8 | Ajimol Das |
| Employee | Details of employee in a company.  #8 | Ajimol Das |
| Todo | Details of appointment for coop advisors made by students in college.  #8 | Ajimol Das |

database backup file 

# NodeJS Service Information

|  |  |  |
| --- | --- | --- |
| Service Path | Details | Team Member(s) worked on it |
| '/api/books' | Api to retrieve all documents from books collection | Ajimol Das |
| '/api/student/:sid' | Api to retrieve student collection details according to the input parameter | Ajimol Das |
| '/api/employee/update' | Api to update employee collection | Ajimol Das |
| '/api/updatedcustomers' | Api to update customers collection | Ajimol Das |
| '/api/Newstudent' | Api to insert a new student document to student collection | Ajimol Das |
| '/api/NewCustomer’ | Api to insert a new customer to customer collection | Ajimol Das |
| '/api/deleteCustomer' | Api to delete a customer document from customer collection | Ajimol Das |
| '/api/deleteStudent' | Api to delete a student document from student collection | Ajimol Das |

Service Codes: 

# Application or Website details

SCREENSHOTS

Homepage with search option, List of all appointments and form for creating a new appointment.

Here i have given all details into the form and clicking the create button now.



After creating a appointment its right away reflected in the appointment list.



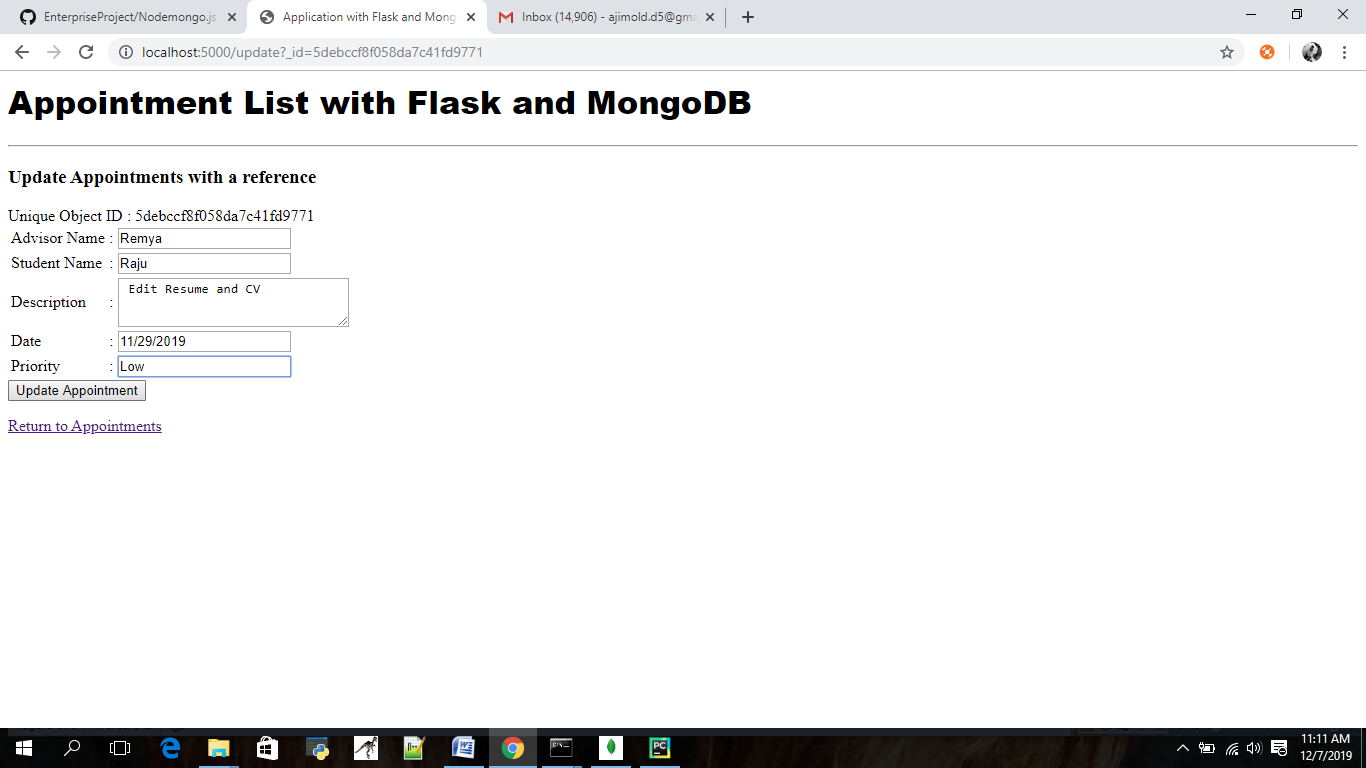
Before clicking delete button



After deleting the last appointment in the list



Updating the appointment using update button



After updating



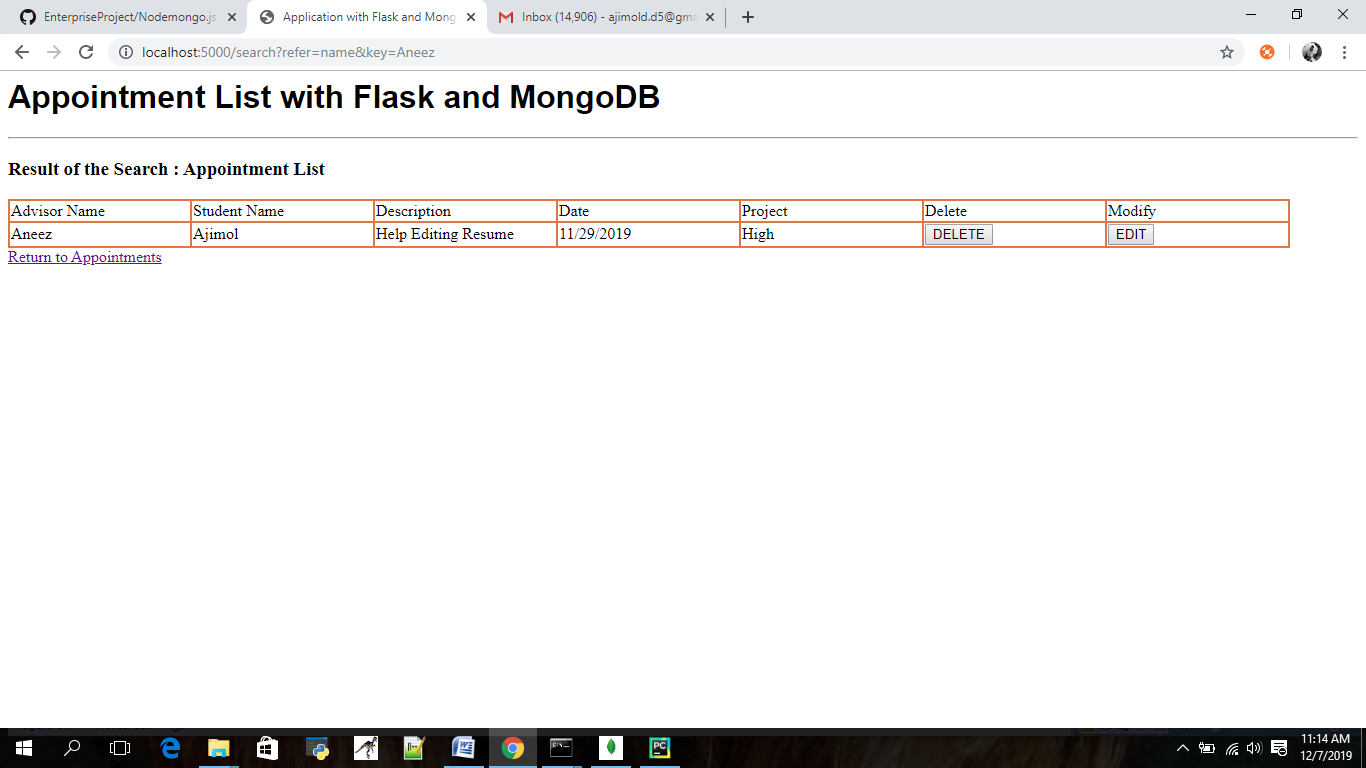
Search option with all fields in appointment list.



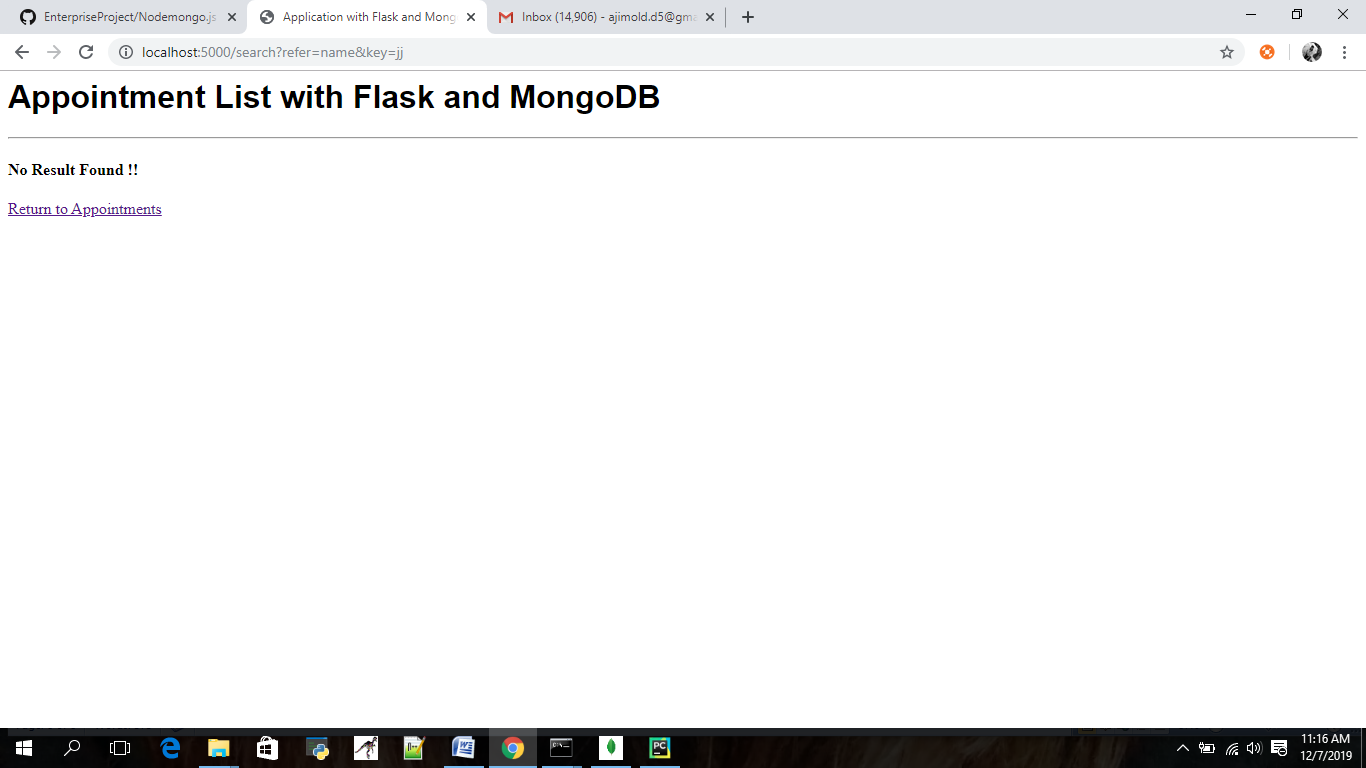
Searching according to the advisors name



Search results page



Searching data which is not in the list



**Attached the codes.**

Code files..







# How this application will be deployed into Cloud?

Among all the cloud service providers Amazon is considered is the most powerful and flexible solution. AWS's virtual cloud platform comes with most of the attributes of an actual computer including hardware (CPU(s) & GPU(s) for processing, hard-disk/SSD for storage & local/RAM for memory); an operating system to choose from and pre-loaded apps like web servers, databases, CRM, etc.

Amazon lends its services to individuals, companies, and governments. Amazon Web Services allow their subscribers to enjoy a full-fledged virtual cluster of computers, at any time, based on their requirements. The entire service is enabled through the internet.

My application is using flask for creating a python app with mongo db so i founded its best to **use elastic beanstalk**

**Deploying a New Application Version**

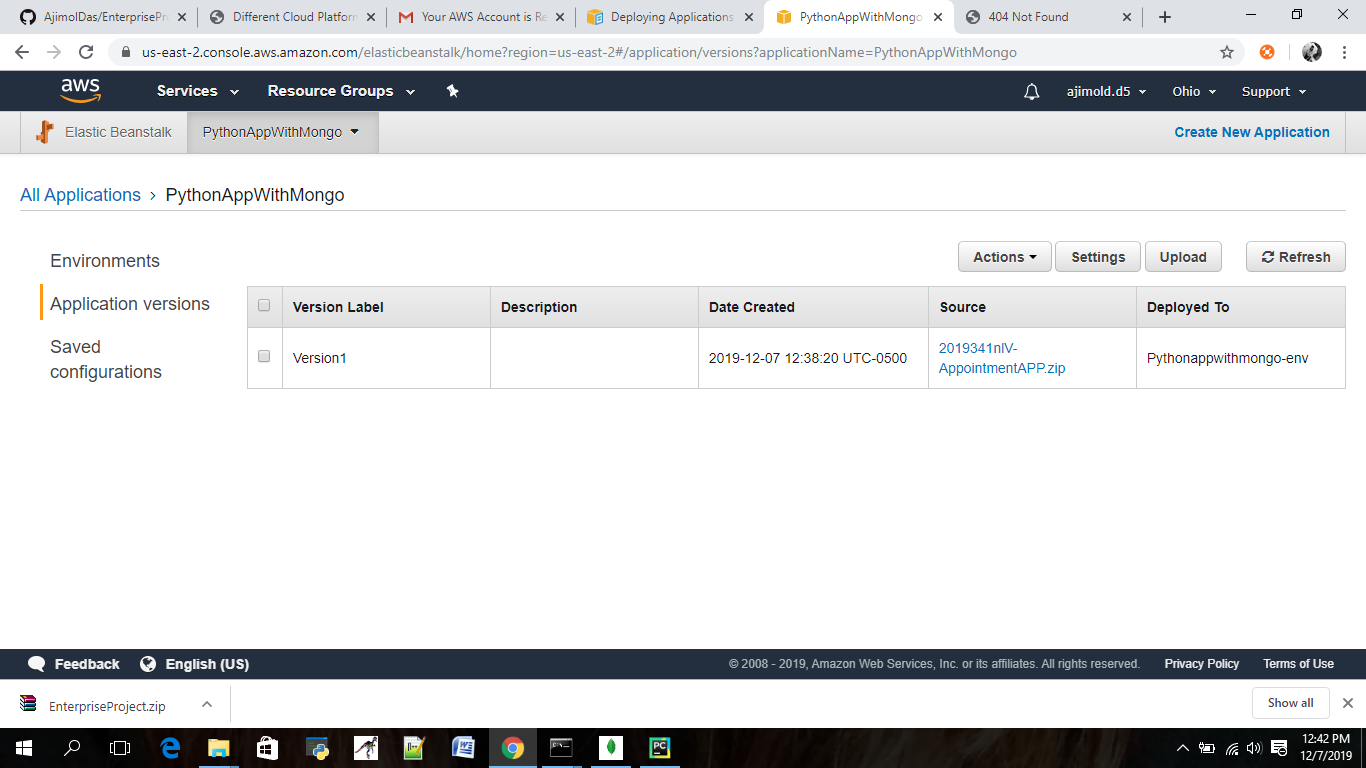
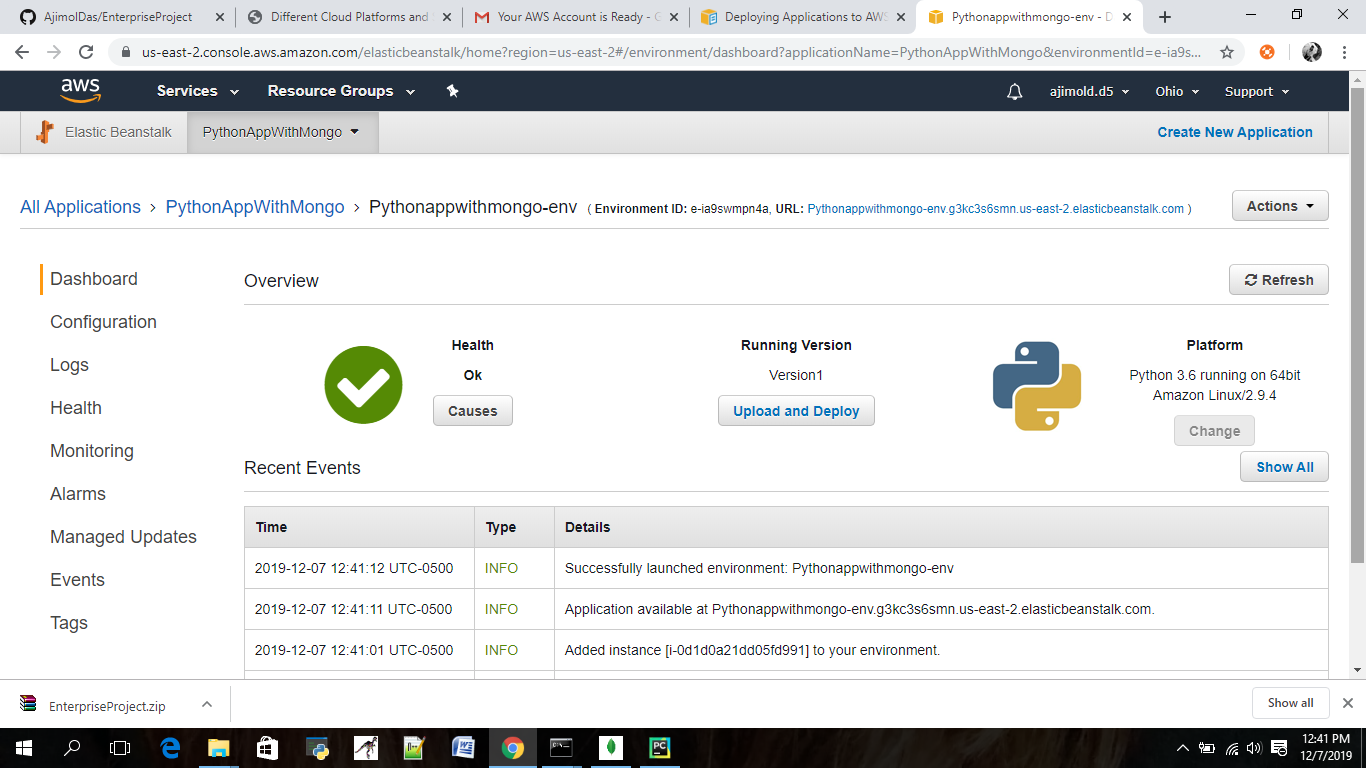
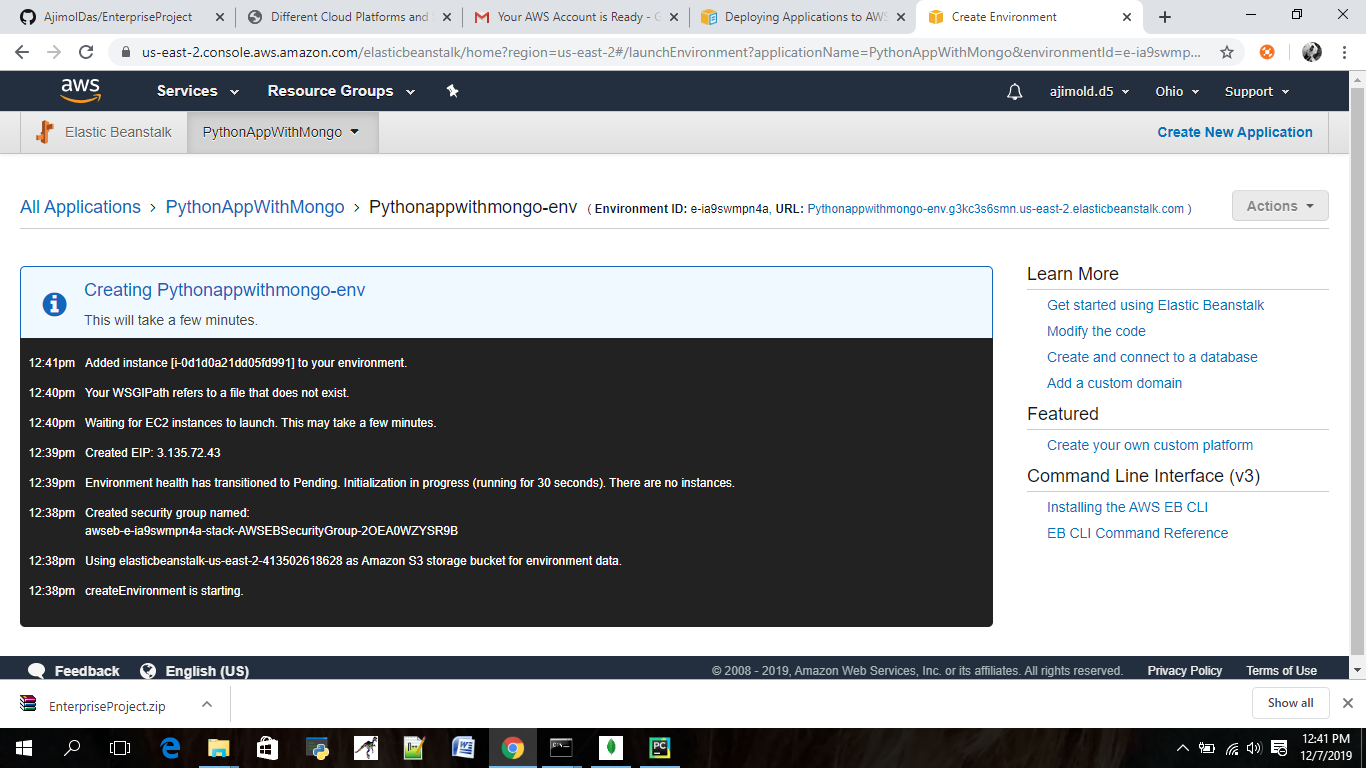
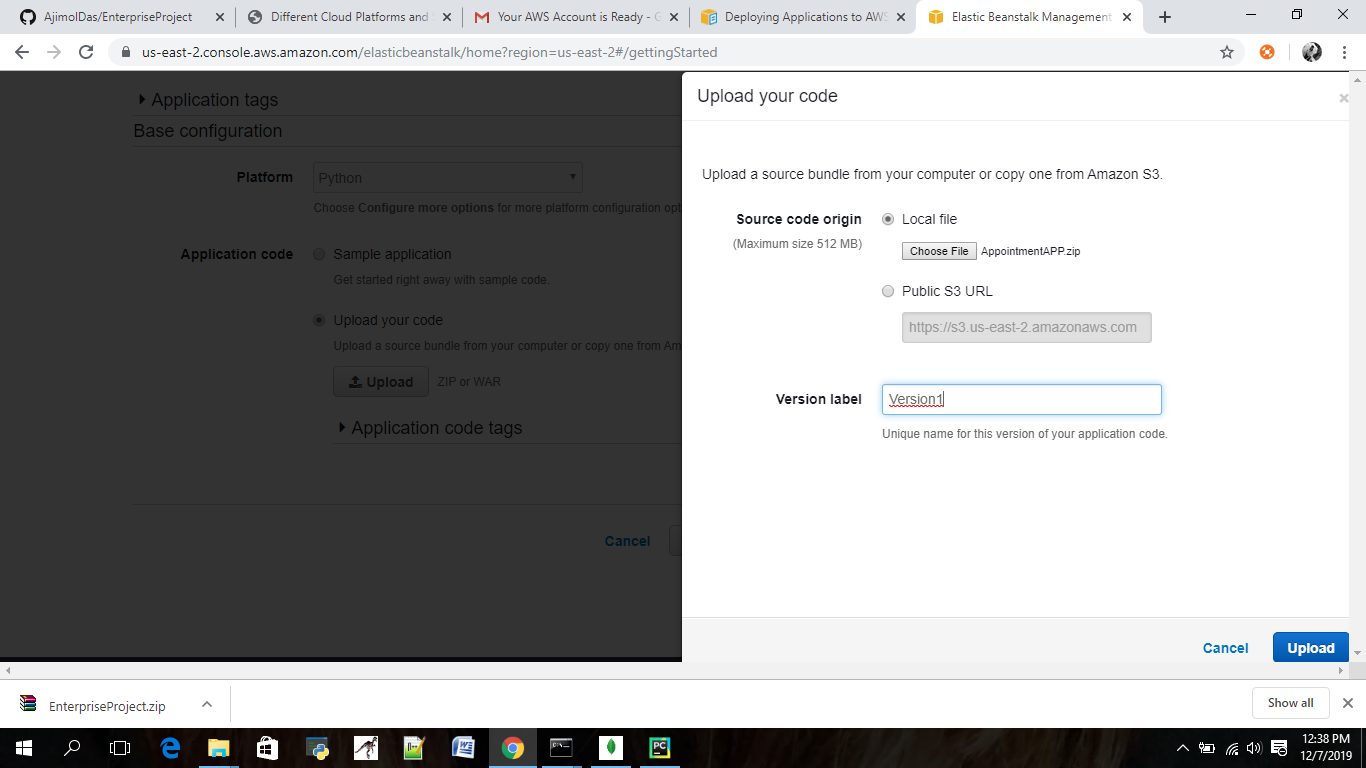
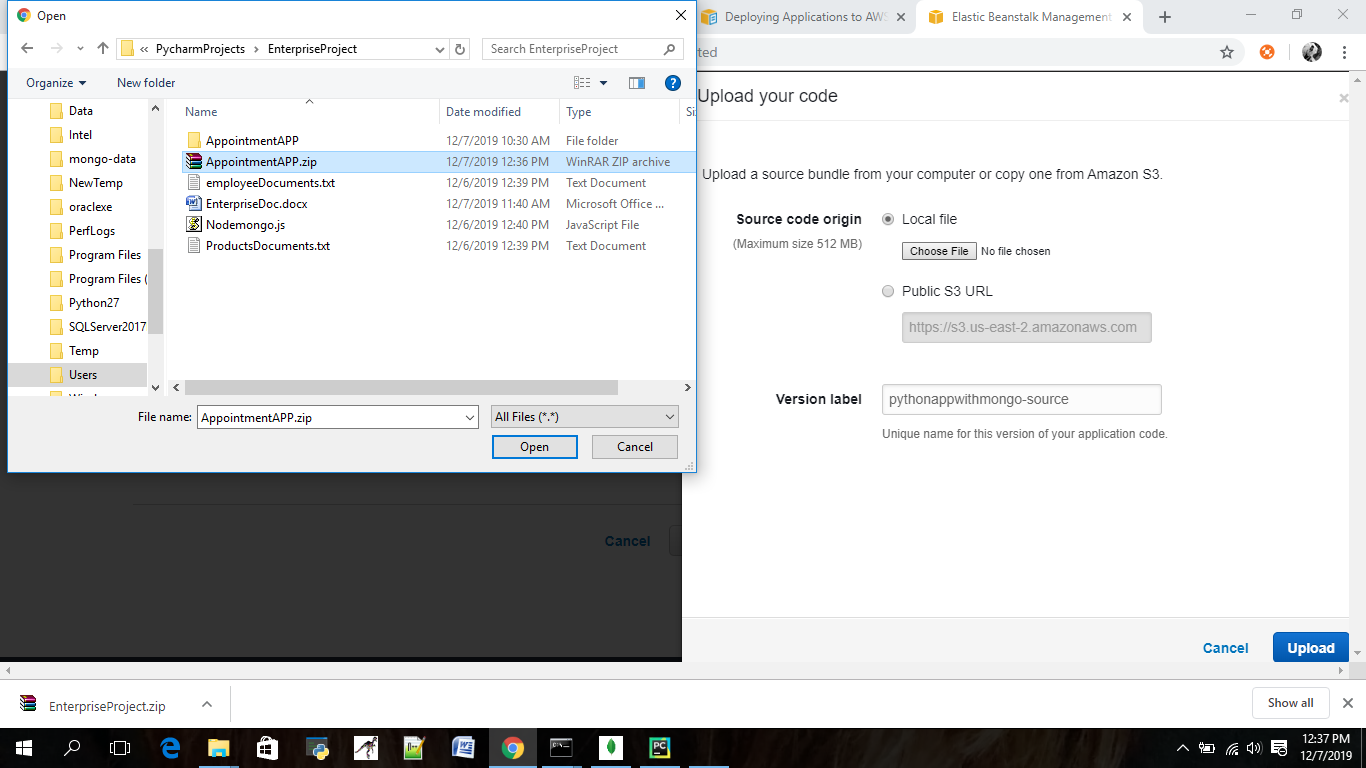
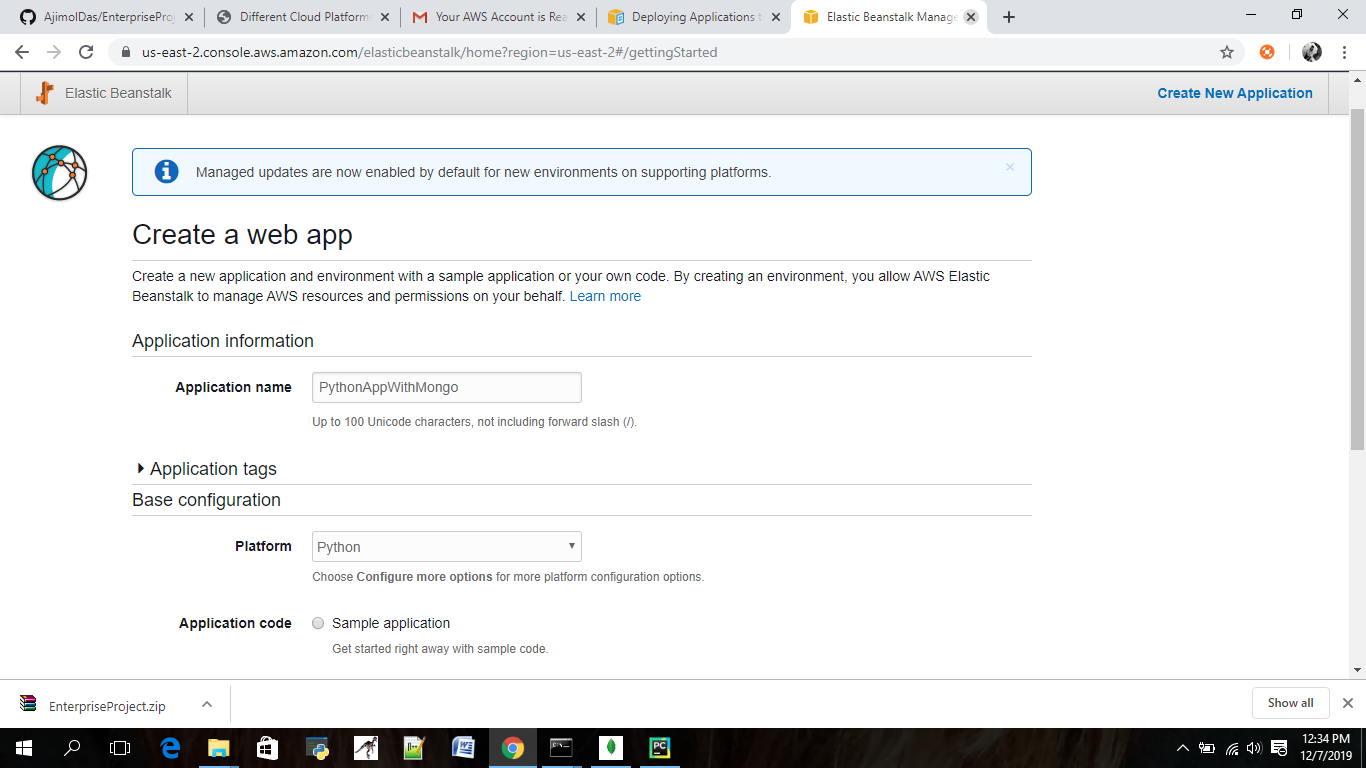
You can perform deployments from your environment's dashboard.

**To deploy a new application version to an Elastic Beanstalk environment**

1. Open the [Elastic Beanstalk console](https://console.aws.amazon.com/elasticbeanstalk).
2. Navigate to the [management page](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environments-console.html) for your environment.
3. Choose **Upload and Deploy**.
4. Use the on-screen form to upload the application source bundle.
5. Choose **Deploy**.

I deployed my application in elastic beanstalk in the name PythonAppwithmongo.

Providing the screenshots below…



<https://us-east-2.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-2#/application/versions?applicationName=PythonAppWithMongo>