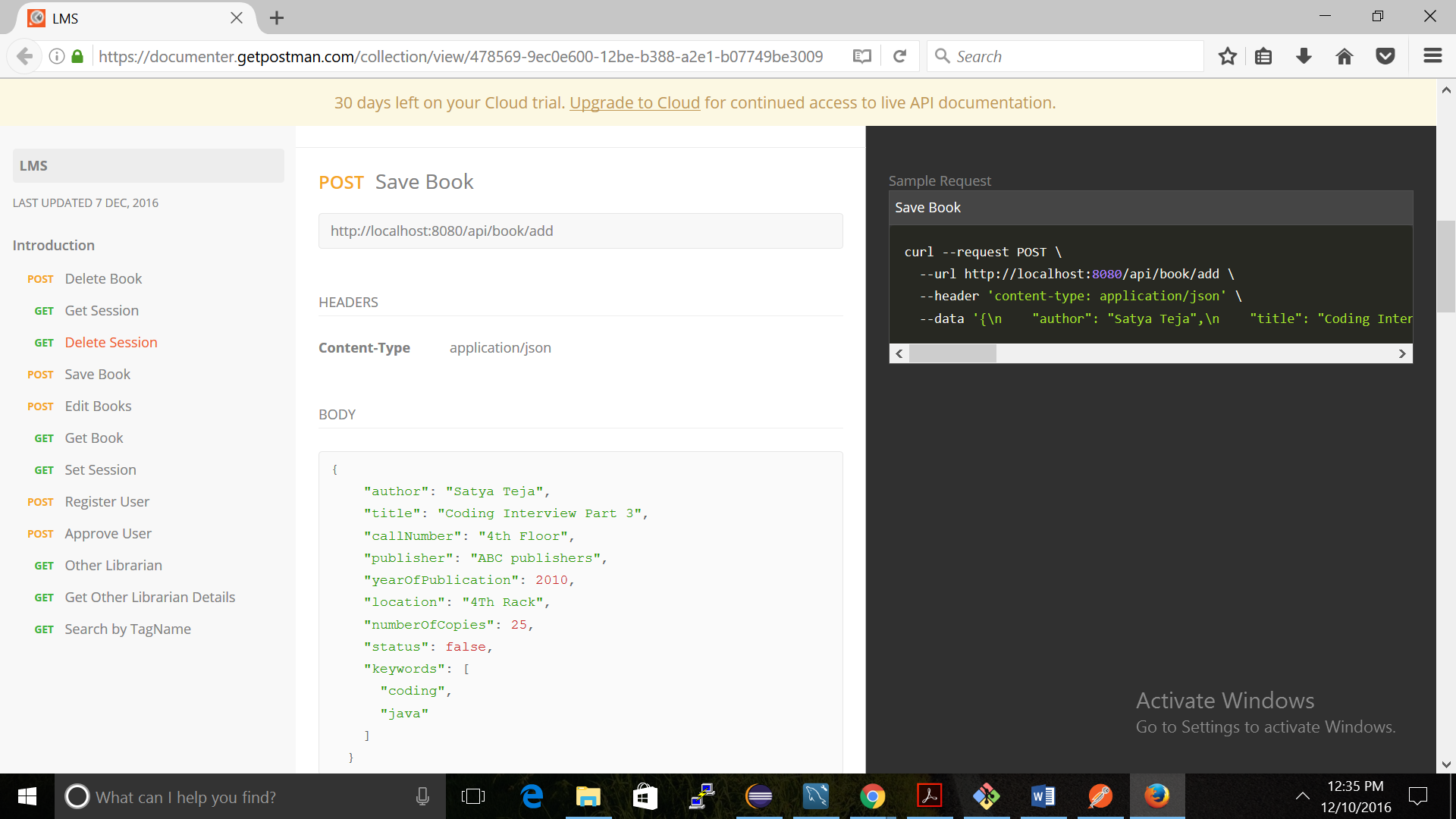
# Testing Plan

Testing or Quality Assurance is an important checkpoint at every stage of the development activity. Without proper testing we can’t make sure that the product can be consumed by the end user. We have done unit testing, integration testing, functional testing and performance testing to make sure that the Smart Library that we developed is flawless and meets the end user’s expectation. Below mentioned are the details of testing activities carried out by the team.

## Unit Testing

Unit testing has been carried out at API level, we have developed all the services as reusable REST services so that each API’s can be individually tested. We have used POSTMAN Rest Client which creates a collaborative API Testing platform, which is very easy to use and at the same time intuitive. The services created by one person can be tested by other team members, as the POSTMAN Rest Client gives you a clear documentation of each API written thereby facilitating easy testing. Below shown is a snap of how POSTMAN client works.



## Integration Testing

As mentioned in the introduction we followed REST architecture in which the Backend service implementation is decoupled from the front end. So there is a need of extensive testing when we do the integration of backend services with the front end JSP pages. We put a collaborative effort and gave ample time to make sure that we cover the minute details of the integration aspect.

## Functional Testing

Functionality testing is an aspect of Quality Assurance. We have done extensive functional testing, by validating against each requirement. All the requirement has been individually tested manually covering both positive and negative scenarios. In order to eliminate bugs, we made sure that we are testing the other person’s module. This way we can uncover more bugs. After the functional testing we had one final round of acceptance testing before we went ahead with the performance testing.

## Performance Testing

Performance Testing is an important aspect to make sure before we release the product to the end users. This gives information regarding how much the load our application can withstand and also we will get inputs regarding how effectively we have to scale our application depending on the estimated end users. We used Apache Jmeter to perform load testing, we have taken a couple of services and captured the response time for different concurrency levels. Our application was able to handle up to 1000 without any performance issues. The results of the performance testing is shown in the below graphs.

**Login Request**

**Get Books**

# 2. Lessons learned and possible future work

## Lessons learned

Smart Library was a wonderful project which helped us to understand the implementation details of the Spring MVC concepts that we studied in the class. The modules which we implemented were

1. Spring Session Implementation
2. Spring MVC
3. JPA Concepts
4. Angular Integration with the JSP pages

We also used collaborative effort, configuration management features using Github for our development process which was indeed a good lesson learned. We have also deployed our application in Amazon Ec2 which helped us understand about Amazon Relational database services and also deployment methods for Amazon Ec2.

## Possible Future Work

1. Use of Continuous Integration (Jenkins) for automating the deployment to Amazon Ec2 whenever there is a commit in Github. Right now, we have to manually do the work of deployment to cloud platform.
2. Finding out the bottleneck in the application and working on performance related issues in the application. Using APM (Application performance Management) tools like Appdynamics, New Relic etc.
3. Enhancing the UI after getting inputs from the end users.