Auto Attendance Management System

CS 816 Software Production Engineering

Jay Kakkad
(MT2018043)

Sayantan Majumder
(MT2018107)

Sandip Kar
(MT2018102)

IIIT Bangalore

Contents

1. Abstract

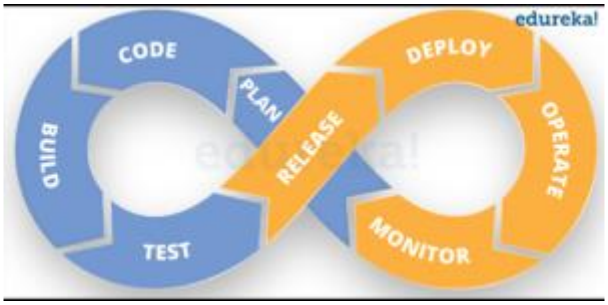
Actual goal of this project is to overcome problems due to manual method of taking down attendance which comprises faculty taking down attendance of students in attendance register. We will try to solve above problem by using technology so that the process gains an automatic way of doing that particular task and thereby increasing overall efficiency.

Maintenance of records of students along with monitoring of class attendance is an area of administration that requires significant amount of time and efforts for management. Automated Attendance Management System performs the daily activities of attendance analysis, for which facial recognition is an important aspect. The prevalent techniques and methodologies for detecting and recognizing face fail to overcome issues such as scaling, pose, illumination, variations, rotation, and occlusions. The proposed system provides features such as detection of faces, extraction of the features, detection of extracted features, analysis of students' attendance and monthly attendance report generation.

2. Introduction

a. Why DevOps

DevOps describes a culture and set of processes that bring development and operations teams together to complete software development. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches. And, it’s gaining popularity at a rapid rate.



DevOps Lifecycle can be broadly broken down into the below DevOps Stages:

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Virtualization and Containerization

These stages are the building blocks to achieve DevOps as a whole.

Here are the top five reasons why the industries have been so quick to adopt DevOps principles:

1. Shorter Development Cycles, Faster Innovation

When development and operations teams are in separate silos, it’s usually difficult to tell if an application is ready for operations. When development teams simply turn over an application, the operations’ cycle times are extended needlessly.

With a combined development and operations team, applications are ready for use much more quickly. This is important, since companies succeed based on their ability to innovate faster than their competitors do. In fact, Kevin Murphy from Red Hat estimates that shorter development cycles translate to bringing an application to market 60 percent faster than with traditional approaches.

2. Reduced Deployment Failures, Rollbacks, and Time to Recover

Part of the reason teams experience deployment failures is due to programming defects. The shorter development cycles with DevOps promote more frequent code releases. This, in turn, makes it easier to spot code defects. Therefore, teams can reduce the number of deployment failures using agile programming principles that call for collaboration and modular programming. Rollbacks are similarly easier to manage because, when necessary, only some modules are affected.

Time to recover is an important issue, because some failure has to be expected. But recovery is much faster when the development and operations teams have been working together, exchanging ideas and accounting for both teams’ challenges during development.

3. Improved Communication and Collaboration

DevOps improves the software development culture. Combined teams are happier and more productive. The culture becomes focused on performance rather than individual goals. When the teams trust each other, they can experiment and innovate more effectively. The teams can focus on getting the product to market or into production, and their KPIs should be structured accordingly.

It’s no longer a matter of “turning over” the application to operations and waiting to see what happens. Operations doesn’t need to wait for a different team to troubleshoot and fix a problem. The process becomes increasingly seamless as all individuals work toward a common goal.

4. Increased Efficiencies

Increased efficiency helps to speed the development process and make it less prone to error. There are ways to automate DevOps tasks. Continuous integration servers automate the process of testing code, reducing the amount of manual work required. This means that software engineers can focus on completing tasks that can’t be automated.

Acceleration tools are another opportunity for increasing efficiency. For example:

Scalable infrastructures, such as cloud-based platforms, increase the access the team has to hardware resources. As a result, testing and deployment operations speed up.

Build acceleration tools can be used to compile code more quickly.

Parallel workflows can be embedded into the continuous delivery chain to avoid delays; one team waits for another to complete its work.

Using one environment avoids the useless task of transferring data between environments. This means you don’t have to use one environment for development, a different environment for testing, and a third for deployment.

5. Reduced Costs and IT Headcount

All of the DevOps benefits translate to reduced overall costs and IT headcount requirements. According to Kevin Murphy from Red Hat, DevOps development teams require 35 percent less IT staff and 30 percent lower IT costs.

3. About the application

The traditional manual methods of monitoring student attendance in lectures are tedious as the signed attendance sheets have to be manually logged into a computer system for analysis. This is time consuming and prone to inaccuracies as some students in the department often fill attendance of their absent colleagues and hence rendering this method ineffective in tracking the students' class attendance. Use of the face detection and recognition system in lieu of the traditional methods will provide a fast and effective method of capturing and managing student attendance accurately while offering a secure, stable and robust storage of the system records, where upon authorization one can access it for purposes like administration, monitoring and generating analysis report.

The proposed system will provide an optimized platform that will help users to easily mark attendance, view user data and their daily attendance records. The proposed system will provide secure attendance management via face recognition techniques.

Face recognition will be the core feature of the proposed system and this system will compute and take attendance on daily basis. Also, we have added an interesting feature in the system that will take automatic attendance both at the starting and ending of the lecture, and then based on the presence in both the time the attendance will be marked, i.e. if the person is successfully detected both at the starting and ending , then only the attendance will be marked , else the person will be marked absent.

The interval of the lecture i.e. the starting and the ending time can be set as per the requirements.

The proposed system will allow users to print the desired attendance analysis report based on weekly, monthly and semester time periods.

The overall objective is to develop an automated attendance management system comprising of a desktop application working in conjunction with a web portal to perform the following tasks:

- To detect faces real time.
- To recognize the detected face by using suitable algorithm.
- To mark the attendance only after the successful detection both at the starting and ending time, else the candidate will be marked absent.
- To update the daily attendance, register after a successful match.

4. Software Development Life Cycle

a. Scope of the project

The scope of the project is the system on which software is installed, i.e. the project can be used as a desktop application and on web portal. It will work for a particular institute. But later on the project can be modified to operate online for any institute, or even corporate sectors.

b. Project architecture & workflows

Plan of work

The system consists of a camera that takes real time view of the student's face moving into the classroom and sends it to the administration server using the web service. For the database, the input are the image of the student, the pre- processing of the image is done and then the features of face are extracted using Local Binary Pattern (LBP) and Histogram of Oriented Gradients(HOG), the features are eyes, nose, and mouth, and then it is subjected to the Support Vector Machine (SVM) classifier. After all this process the images of the student are stored in the database. In the administration server, photo is processed. From the video we generate number of frame/images. The pre-processing of the image/frame is done and then image subjected to the Face Detection where faces from the image/frame are detected. The features of the face are to be extracted in the feature extraction module using LBP and HOG; the features are eyes, nose, and mouth. Then the SVM training is done on the faces. Here all the faces are detected from the input image and the algorithm compares them one by one with the face database. If the features of the face match with the face of the database then the attendance is marked from where anyone can access and use it for different purposes. Teachers come in the class and just press a button to start the attendance process and the system automatically gets the attendance without even the intensions of students and teacher. In this way a lot of time is saved and this is highly securing process no one can mark the attendance of other. Camera takes the video continuously to detect and recognize all the students in the classroom. In order to avoid the false detection we are using the Ada boost technique (is a machine learning meta-algorithm). Using this technique enhance the efficiency and accuracy of the detection process. After detection of face changes are made in database. In attendance record table new entry is made which includes presence status, userID and date.

Materials / Tools required

Software:

- Angular CLI
- Node.js
- Express.js
- MongoDB
- Python Interpreter
- Sublime Text (or any text editor)

Programming Language:

- MEAN (Mongo + Angular + Express + Node)
- Python

Technology:

- Pandas
- Scikit Learn
- OpenCV
- Bootstrap
- Material (for Angular)

Data Dictionary

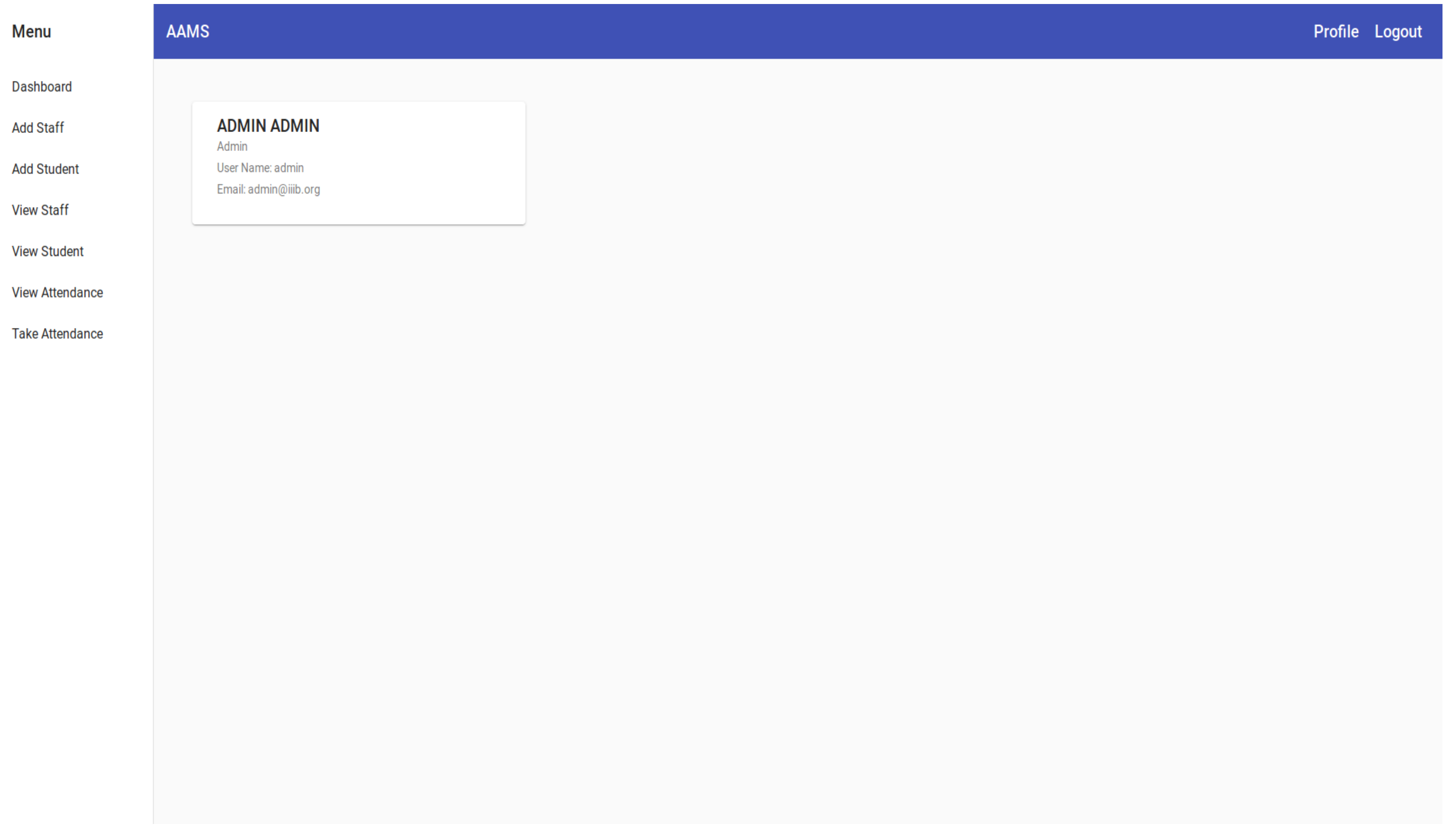
LOGIN : This table will contain important data of user which will be recorded during registration.

Field	Type	Max length	Other
userName	varchar	255	Primary key
firstName	varchar	255	
lastName	varchar	255	
email	varchar	255	NA
password	varchar	255	Hashed Format
userType	varchar	255	Admin/Staff/Student
Department	varchar	255	NA
Program	varchar	255	MS/PhD/MTech /iMTech

Attendance : This table is created dynamically based on the date and contains all information about attendance.

Field	Type	Max length	Other
userName	varchar	20	Primary Key
P/A	varchar	20	NA
MarkedTime	varchar	40	NA

Dashboard



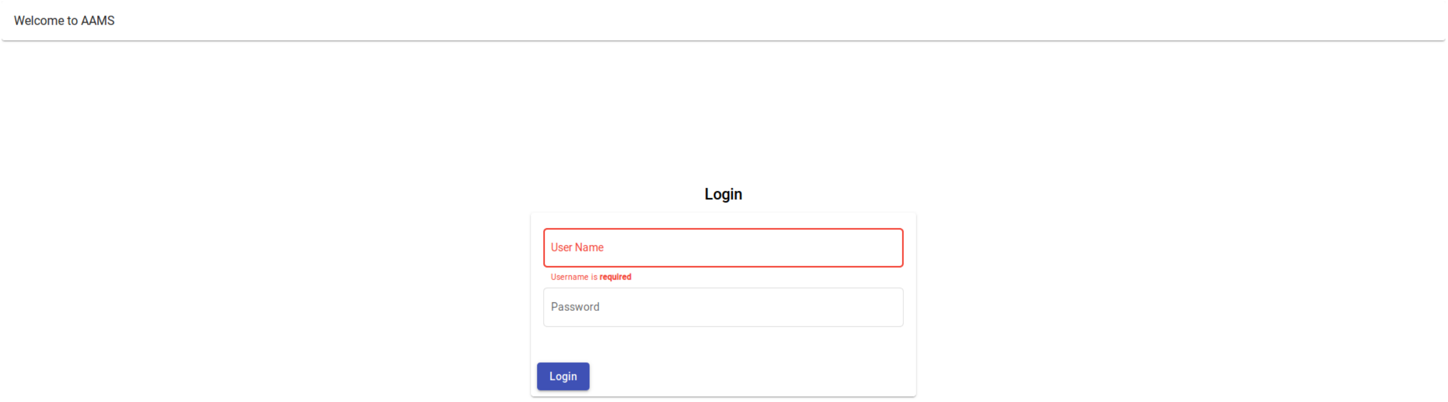
ADMIN ADMIN

Admin

User Name: admin

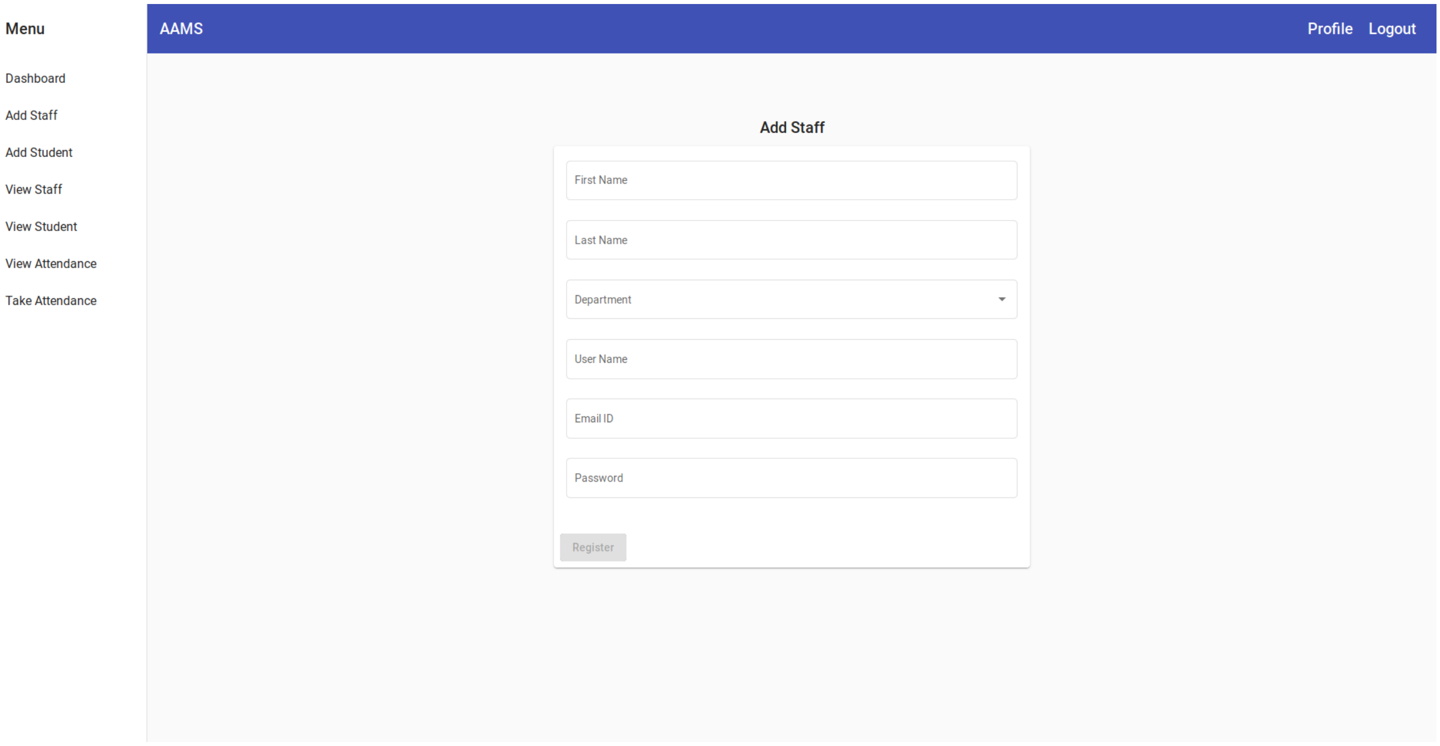
Email: admin@iiib.org

Login Form

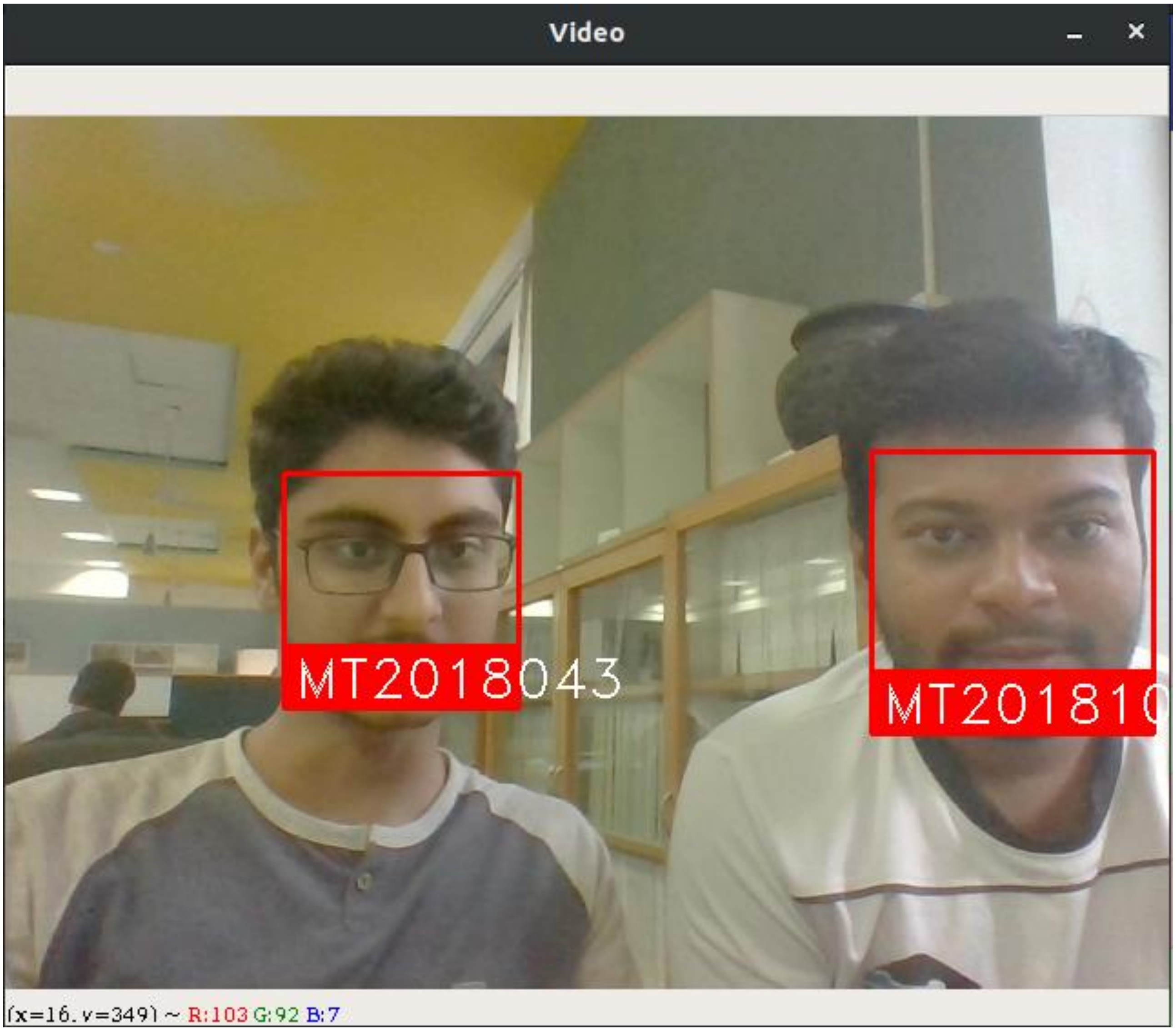


Username is required

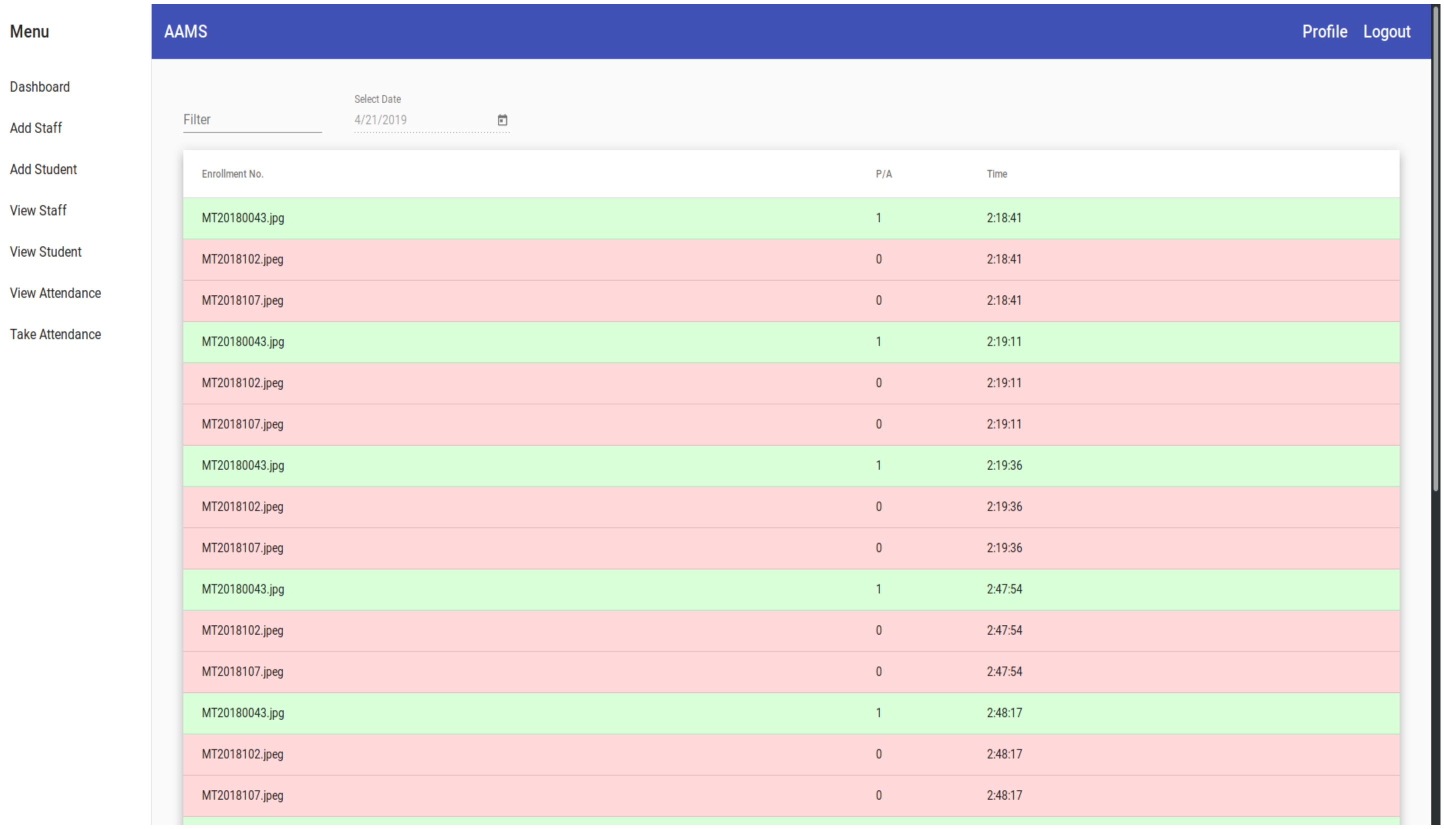
Staff registration form



Facial Recognition



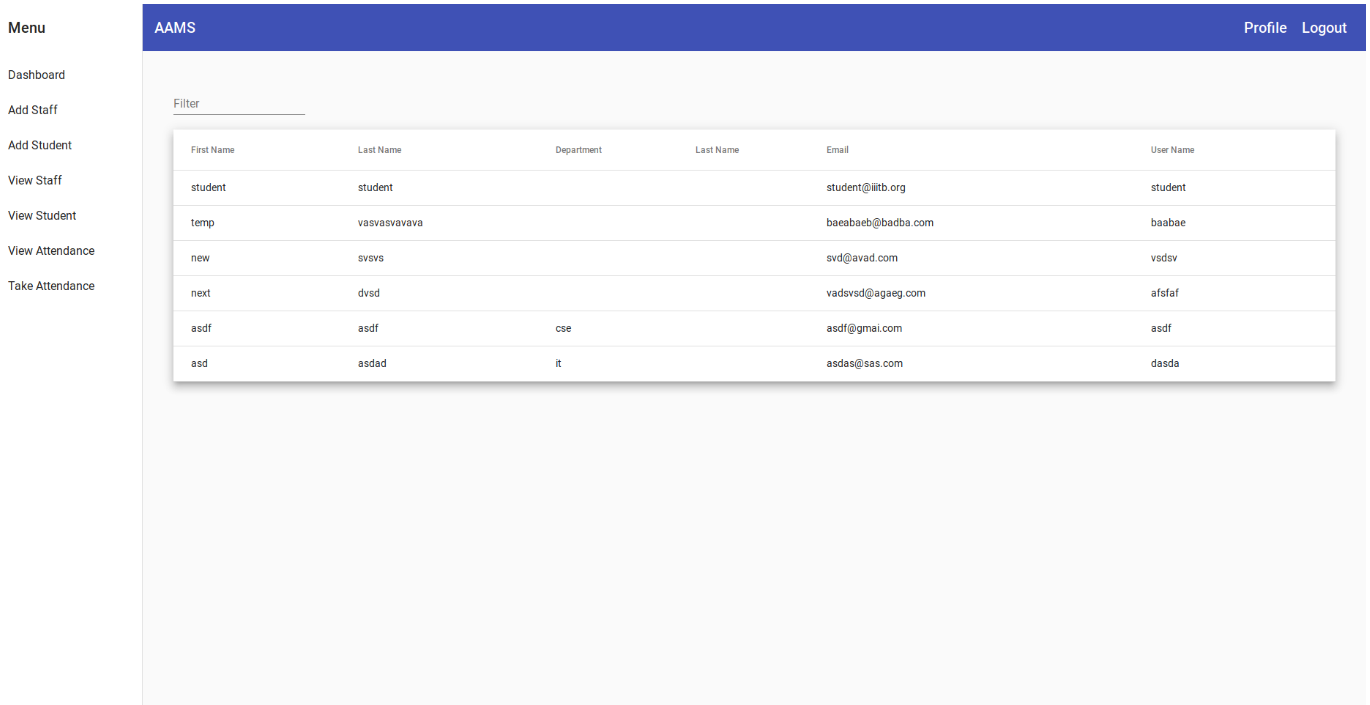
Attendance View



Select Date 4/21/2019

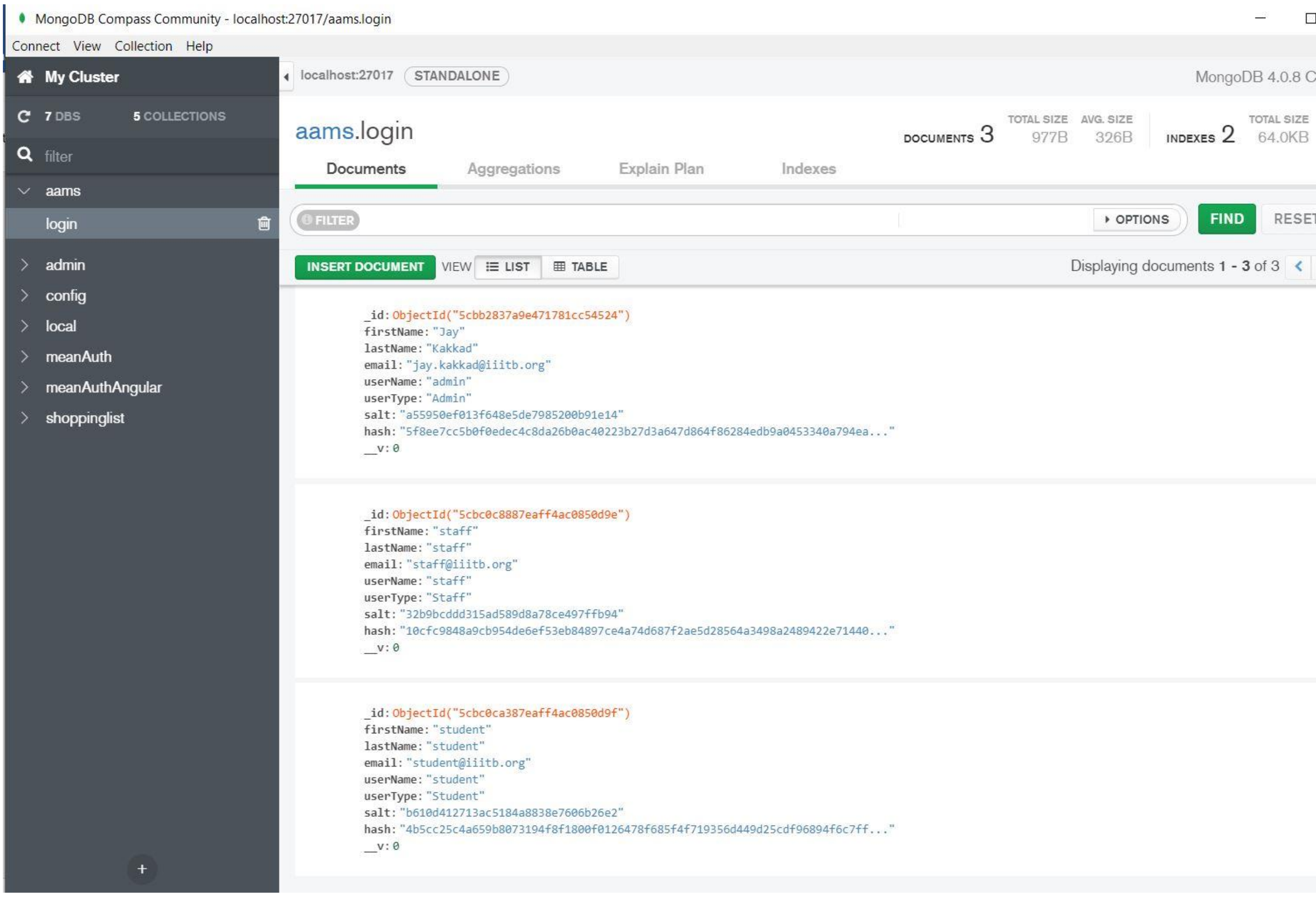
Enrollment No.	P/A	Time
MT20180043.jpg	1	2:18:41
MT2018102.jpeg	0	2:18:41
MT2018107.jpeg	0	2:18:41
MT20180043.jpg	1	2:19:11
MT2018102.jpeg	0	2:19:11
MT2018107.jpeg	0	2:19:11
MT20180043.jpg	1	2:19:36
MT2018102.jpeg	0	2:19:36
MT2018107.jpeg	0	2:19:36
MT20180043.jpg	1	2:47:54
MT2018102.jpeg	0	2:47:54
MT2018107.jpeg	0	2:47:54
MT20180043.jpg	1	2:48:17
MT2018102.jpeg	0	2:48:17
MT2018107.jpeg	0	2:48:17

Student Details View



First Name	Last Name	Department	Last Name	Email	User Name
student	student			student@iiitb.org	student
temp	vasvasvavava			baeabaeb@badba.com	baabae
new	svsvs			svd@avad.com	vsdsv
next	dvsd			vadsvsd@agaeg.com	afsfaf
asdf	asdf	cse		asdf@gmai.com	asdf
asd	asdad	it		asdas@sas.com	dasda

MongoDB Storage



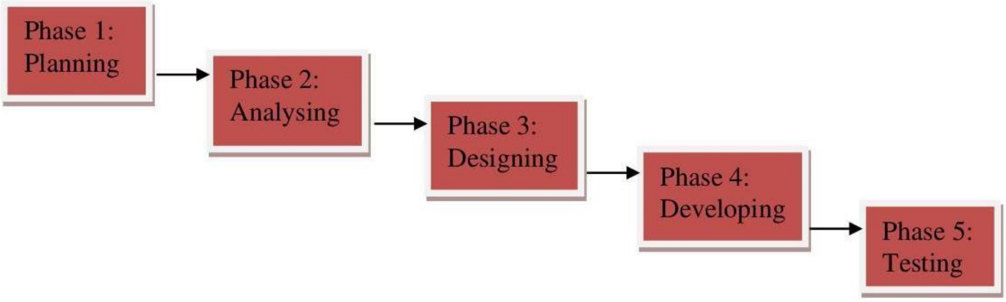
```
_id: ObjectId("5cbb2837a9e471781cc54524")
firstName: "Jay"
lastName: "Kakkad"
email: "jay.kakkad@iiitb.org"
userName: "admin"
userType: "Admin"
salt: "a55950ef013f648e5de7985200b91e14"
hash: "5f8ee7cc5b0f0edec4c8da26b0ac40223b27d3a647d864f86284edb9a0453340a794ea..."
__v: 0

_id: ObjectId("5cbc0c8887eaff4ac0850d9e")
firstName: "staff"
lastName: "staff"
email: "staff@iiitb.org"
userName: "staff"
userType: "Staff"
salt: "32b9bcddd315ad589d8a78ce497ffb94"
hash: "10cfc9848a9cb954de6ef53eb84897ce4a74d687f2ae5d28564a3498a2489422e71440..."
__v: 0

_id: ObjectId("5cbc0ca387eaff4ac0850d9f")
firstName: "student"
lastName: "student"
email: "student@iiitb.org"
userName: "student"
userType: "Student"
salt: "b610d412713ac5184a8838e7606b26e2"
hash: "4b5cc25c4a659b8073194f8f1800f0126478f685f4f719356d449d25cdf96894f6c7ff..."
__v: 0
```

Software engineering paradigm applied

Systems Development Life Cycle (SDLC) is the most common process adopted to develop a project and not surprisingly, this project is following this model too. To be precise, waterfall model is being applied. Waterfall model is a sequential model process where the input of a phase actually results from the previous phase.



SDLC Phases

- The planning stage determines the objectives of the project and whether the project should be given the green light to proceed. This is where the proposal submission comes into picture.
- After obtaining the approval, the next phase is analysis. Gathering and analysing the system and user requirements is essential for entry to the design step.
- With the user requirements gathering completed, there is a need to prepare the resources for the project. Be it software or hardware components, careful consideration and selection is to be taken care at this stage. The decision on the resources to be used is further elaborated under the subsections below.
- The next step is to design the system and database structure. Results from the analysis and preparation that were concluded from the previous stage are put into action. With the user requirements in mind, the user interface is designed to suit their easy navigation needs.
- In addition, the number of tables, attributes, primary and unique keys of the database is listed. After completing the design, actual coding begins. Database is created and codes are written. Some of the codes required improvement to it so these are being developed at this fourth stage of the waterfall model.
- With the development completed, testing will begin. The codes and database are tested to ensure the results obtained are as intended.

Advantages:

1) Reduce errors.

Time and Attendance software reduces the risk of human error and ensures and easy, impartial, and orderly approach in addressing specific needs without any confusion. In fact, Time and Attendance software has been shown to have an accuracy rate of more than 99% versus manual systems by eliminating errors in data entry and calculations.

2) Increase security.

Time and attendance software together with biometric data collection devices may be used to control employee access to certain areas within a facility and track employee entry. Biometric data collection devices eliminate buddy punching and also helps reduce costly liabilities, including theft of equipment or property.

3) Increase productivity.

Productivity increases because the process is seamless and makes day-to-day operations more efficient and convenient. Eliminating legacy practices frees up employees time, decreases staffing overhead, and provides supervisors with timely labor data to more effectively manage their operations.

4) Save money.

5) Accurate and Fast Identification

Industrial Leading Facial Recognition Algorithm, match more data than fingerprint, FAR<0.0001%

6) High Usability and Security

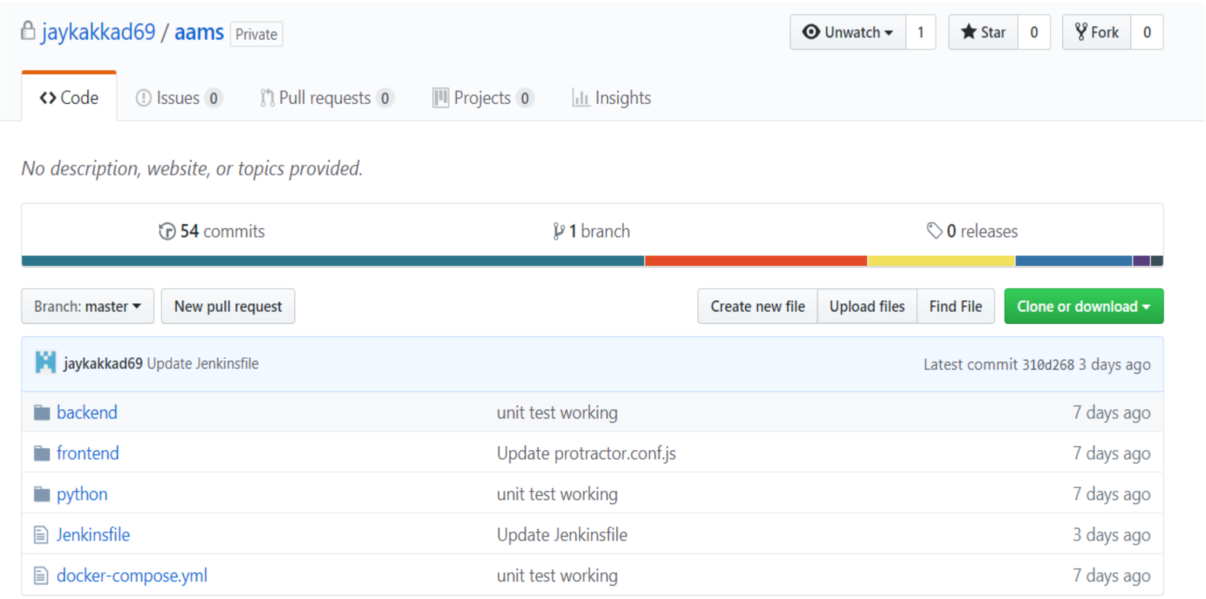
Failure to enroll and acquire rate is less than 0.0001%, fingerprint technology will have problems for enrollment with cold, wet, desquamation, elder, around 5% people cannot get enrolled with fingerprint technology Incident track able for security with photo which captured by camera, there is no evidence with fingerprint technology to track the incident.

7) User friendly design

c. SCM

We are using GitHub as the source control management repository.
Following is the github repository for our project:

https://github.com/jaykakkad69/aams



No description, website, or topics provided.

54 commits 1 branch 0 releases

jaykakkad69 Update Jenkinsfile Latest commit 310d268 3 days ago

backend	unit test working	7 days ago
frontend	Update protractor.conf.js	7 days ago
python	unit test working	7 days ago
Jenkinsfile	Update Jenkinsfile	3 days ago
docker-compose.yml	unit test working	7 days ago

d. Build

For building the application , we are using ‘ng build’ that uses the webpack build tool of npm packages and is used for building angular applications.

The Angular Framework, Angular CLI, and components used by Angular applications are packaged as npm packages and distributed via the npm registry. The Angular CLI creates a default package.json file, which specifies a starter set of packages that work well together and jointly support many common application scenarios.

e. Test

We are doing 4 types of testing:

1. Linting

Linting is the process of checking the source code for Programmatic as well as Stylistic errors. This is most helpful in identifying some common and uncommon mistakes that are made during coding.

A Lint or a Linter is a program that supports linting (verifying code quality). They are available for most languages like JavaScript, CSS, HTML, Python, etc.

The tool that we are using for linting is JSLint.

2. Unit Testing

Below are some benefits of unit testing:

- **Improve the design of implementations.**
  Coding a feature without giving it a lot of thought to the design is a very common mistake among developers. Using unit testing is going to enforce to think and re-think the design, and if you are using TDD the impact is even bigger.
- **Allows refactoring.**
  Since we already have tests that ensure everything is working as expected, we can easily add changes to that code with the certainty that you are not adding any bugs.
- **Add new features without breaking anything.**
  When you are adding a new feature you can run the tests to ensure that you aren't breaking any other part of the application.

The unit testing tools for our project are Jasmine and Karma.

Jasmine-core. Jasmine is the framework we have used to create our tests. It has a bunch of functionalities to allow us the write different kinds of tests.

Karma. Karma is a task runner for our tests. It uses a configuration file in order to set the startup file, the reporters, the testing framework, the browser among other things.

3. End-to-end Testing

For end to end testing in angular projects, we have a tool called **Protractor**.

Protractor is a Node.js program, and runs end-to-end tests that are also written in JavaScript and run with node. Protractor uses WebDriver to control browsers and simulate user actions.

Protractor uses Jasmine for its test syntax. As in unit testing, a test file is comprised of one or more it blocks that describe the requirements of your application. it blocks are made of **commands** and **expectations**. Commands tell Protractor to do something with the application such as navigate to a page or click on a button. Expectations tell Protractor to assert something about the application's state, such as the value of a field or the current URL.

If any expectation within an ‘it’ block fails, the runner marks the it as "failed" and continues on to the next block.

4. Compile-browser Testing

    In this stage, we are building the application using ‘ng build’ to check that the application is successfully running on browser.

f. Artefacts

We are using various artefacts for each testing in Jenkins pipeline as well as for deploying. Below are the docker images that we are using as artefacts:

1. docker 'node:10.14.0-alpine' for fetching dependencies in Jenkins pipeline.
2. docker 'node:10.14.0-alpine' for linting of JavaScript and TypeScript files.
3. docker 'circleci/node:12.0.0-stretch-browsers' for Unit Testing.
4. docker 'circleci/node:12.0.0-stretch-browsers' for E2E Testing.
5. docker 'node:10.14.0-alpine' for Compile-Browser Testing

g. Deploy

For deploying we are using docker-compose to create three containers for the docker images named angular, express and mongo respectively for the frontend, backend and the database of our application.

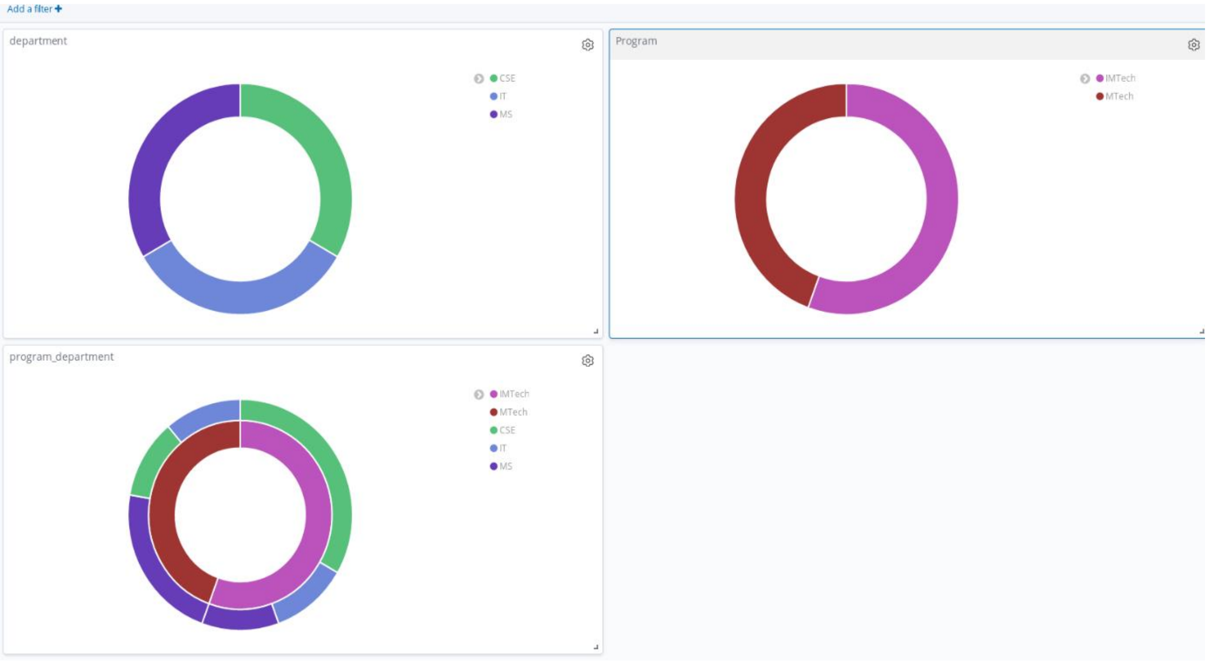
In the Jenkins pipeline we are using Rundeck to deploy the application from a remote machine to the docker containers.

h. Monitor

We are using ELK i.e. Elasticsearch, Logstash and Kibana for continuous monitoring.

Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.

Below is the screenshot of Kibana where are collecting the information of the database and representing in the visual form:


Add a filter ✚
department
CSE
IT
MS
Program
IMTech
MTech
program_department
IMTech
MTech
CSE
IT
MS

5. CI/ CD Pipeline

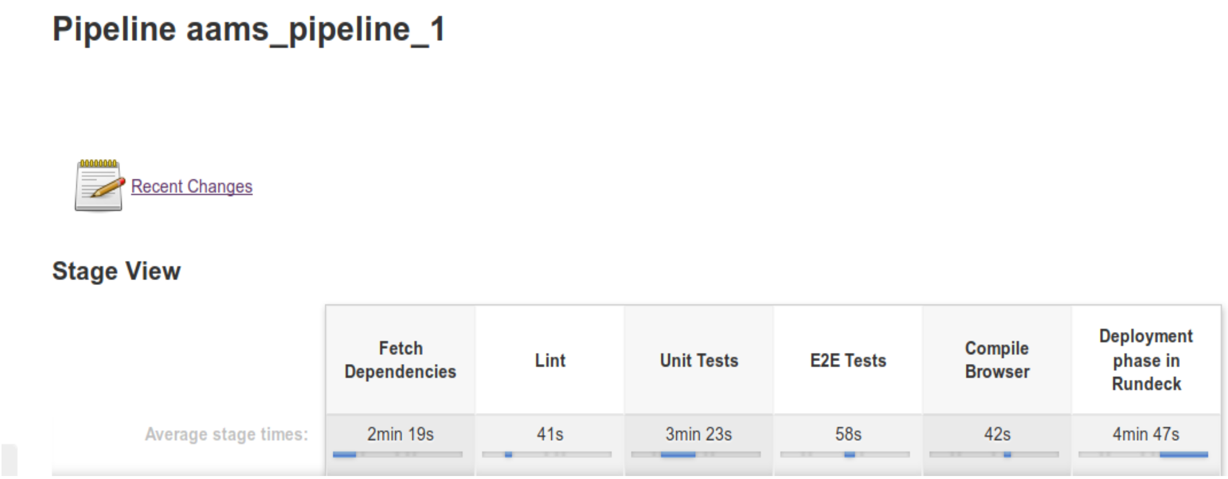
We are integrating all the above steps using Jenkins by creating a declarative pipeline using jenkinsfile.

Also we are deploying the application in separate docker containers by running the docker-compose file present in remote machine using Rundeck that has been integrated with Jenkins.

For continuous checking for any changes committed in the code repository, we are using Ngrok for creation of public IP and then providing the public IP in git webhook for detection of any changes in the code committed in the GitHub.

Thus, the pipeline is automatically triggered whenever there are any changes in the code present in github using the git webhook.

Below is the screenshot of the Jenkins pipeline:

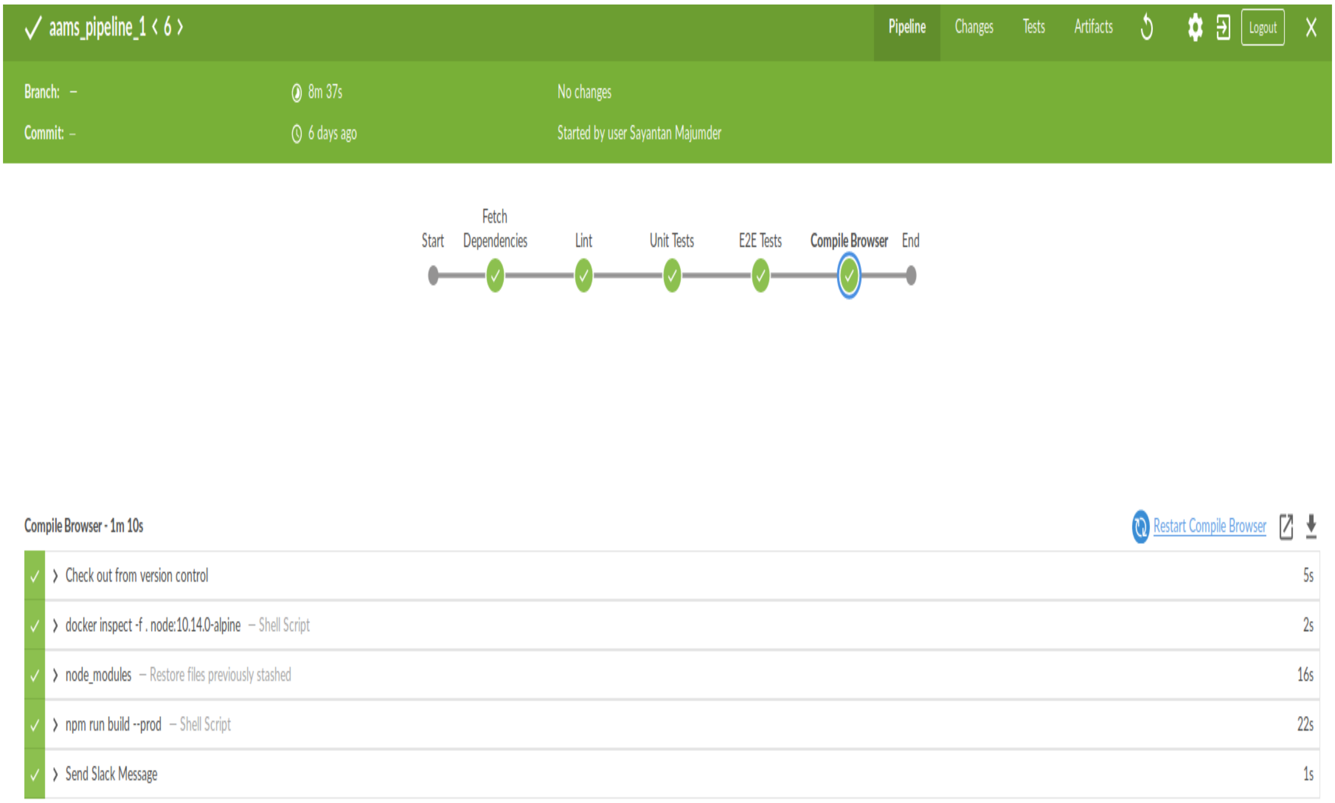


Pipeline aams_pipeline_1

Recent Changes

Stage View

	Fetch Dependencies	Lint	Unit Tests	E2E Tests	Compile Browser	Deployment phase in Rundeck
Average stage times:	2min 19s	41s	3min 23s	58s	42s	4min 47s



Branch: –

Commit: –

8m 37s

6 days ago

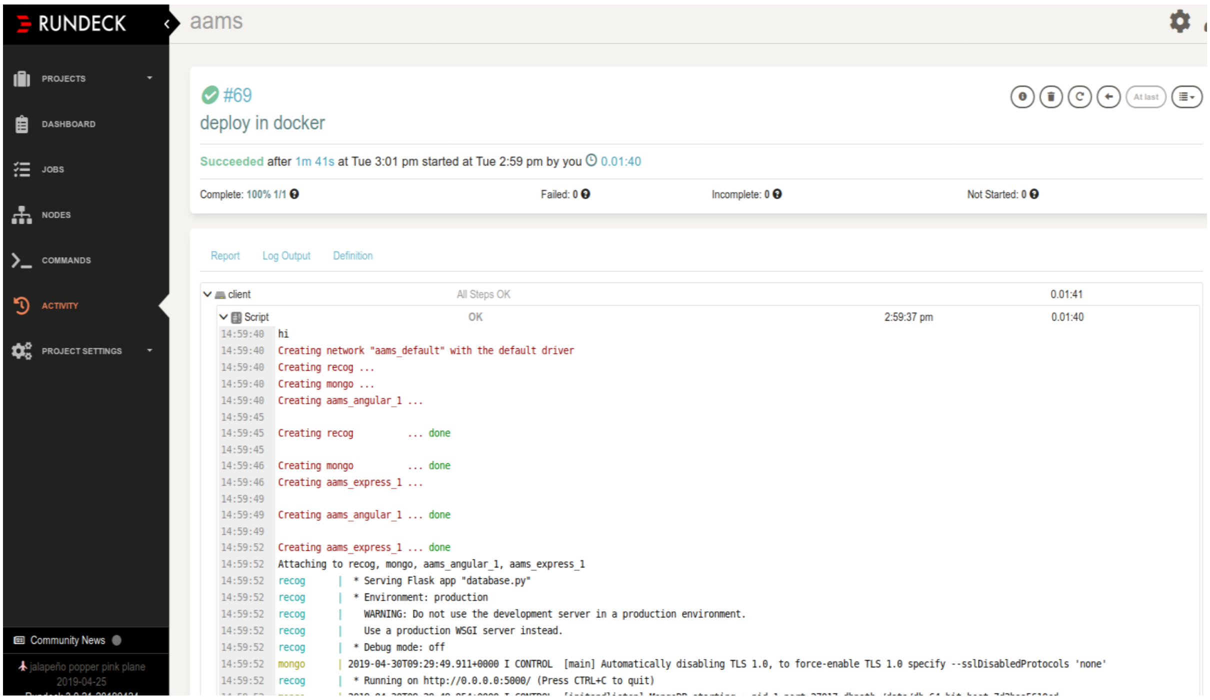
No changes

Started by user Sayantan Majumder

Compile Browser - 1m 10s

✓ › Check out from version control	5s
✓ › docker inspect -f . node:10.14.0-alpine – Shell Script	2s
✓ › node_modules – Restore files previously stashed	16s
✓ › npm run build --prod – Shell Script	22s
✓ › Send Slack Message	1s

Below is the Rundeck job that is being triggered from Jenkins pipeline.



#69

deploy in docker

Succeeded after 1m 41s at Tue 3:01 pm started at Tue 2:59 pm by you 0.01:40

Complete: 100% 1/1 Failed: 0 Incomplete: 0 Not Started: 0

```
client                All Steps OK                                  0.01:41
  Script              OK                          2:59:37 pm        0.01:40
14:59:40  hi
14:59:40  Creating network "aams_default" with the default driver
14:59:40  Creating recog ...
14:59:40  Creating mongo ...
14:59:40  Creating aams_angular_1 ...
14:59:45
14:59:45  Creating recog          ... done
14:59:45
14:59:46  Creating mongo          ... done
14:59:46  Creating aams_express_1 ...
14:59:49
14:59:49  Creating aams_angular_1 ... done
14:59:49
14:59:52  Creating aams_express_1 ... done
14:59:52  Attaching to recog, mongo, aams_angular_1, aams_express_1
14:59:52  recog      |  * Serving Flask app "database.py"
14:59:52  recog      |  * Environment: production
14:59:52  recog      |    WARNING: Do not use the development server in a production environment.
14:59:52  recog      |    Use a production WSGI server instead.
14:59:52  recog      |  * Debug mode: off
14:59:52  mongo      | 2019-04-30T09:29:49.911+0000 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
14:59:52  recog      |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Here we are using Rundeck for deploying the application in docker containers.

6. Results and Discussion

It takes around 10-15 mins for fully deploying the app from the code repository (e.g. github)
to the docker containers i.e. the Jenkins pipeline which takes around 10-15 mins to run. Here the full project is deployed in the dockers, and dockers are light weight virtual machines and dockers are also independent of host operating system. So this project can be deployed anywhere as it needs only docker support.

This application can also be used anywhere in the academic institutes as well as corporate sectors on minor modification of the structure. If there is enough database support, then this application can be used anywhere for taking of attendance.

7. Future Work

Currently this application is not taking separate attendance for different subjects. we can modify this to do so. And this can be also configured to take attendance of staff also. Using attendance, we can also work on generating monthly report for each student and show some graphs accordingly.

8. Conclusion

After configuring this project on a system, we just need to upload the code to the GitHub for any modification, as there is Jenkins pipeline working it will automatically do all the work for us and we can get the live product without any downtime. This application can work in any sectors for taking attendance without any manual intervention and can even used in large scale without much delay.

9. References

1. www.corradin.net/index.php/2017/11/angular-5-ci-and-cd-with-docker/

2. https://medium.com/educatex-engineering/angular-material-app-with-docker-swarm-and-jenkins-ci-cd-deployment-part-1-5c85191af7e7

3. https://medium.com/@tiangolo/angular-in-docker-with-nginx-supporting-environments-built-with-multi-stage-docker-builds-bb9f1724e984

4. http://jasonwatmore.com/post/2018/11/22/angular-7-role-based-authorization-tutorial-with-example#authentication-service-ts

5. https://github.com/coreui/coreui-free-angular-admin-template