SOFTWARE SYSTEMS LAB COURSE PROJECT

Notify Me - By Runtime Terror

Akshat Gupta
190050010

Gaurav Krishan Gupta
190050036

Piyush Raj
190050083

Rachit bagga
19B090008

1 ABSTRACT

In this Project, we built a Centralized Notification System to bridge the communication gap between instructors and students. We have developed different android apps and websites for both instructors and students. Main features of instructor's interface include sending notifications, creating classes along with adding TA's and removing suspicious students. For students, we developed an interface for receiving notifications for the courses in which they are enrolled. All the interfaces have token authentication and hence, no "mischievous monkeys" can exploit the system. Other Salient Features include priority-based notifications, changing and resetting password using email based OTP system.

2 EXTRA FEATURES

- We have made an Android app for dashboard.
- We have made a website for student app.
- Added a extra feature of forget password via email based OTP authentication.
- Sending high priority notifications after a span of 30 minutes(20 seconds now for demonstration purpose) until they are acknowledged.

3 IMPLEMENTATION DETAILS

We have used Django framework for Back-end , AngularJS framework for Web development and flutter for App development. Specific Implementation details are as follows :

3.1 BACKEND- DJANGO

For authentication purposes, We are using the built-in User model(in django.contrib.auth). We have made our own two user models for instructors and students , both of these have a onetoonefield of built-in user so that a user can either be student or instructor. Also an instructor can either be TA or Professor and for each student fcmtoken is stored at time of login. Each course has 3 manytomany fields to user each for instructors, students and TAs and each course has a unique name and a unique invitation code. We also have a notification model (for any event instructor has to set) which stores sender and course as a manytoone Field and Students who have acknowledged as manytomany field , other fields include Title,Content,Priority and createdDate.
Our Django Backend is deployed at a Heroku server at https://notifyme69.herokuapp.com/.
Brief Description of all the API endpoints:

- url:/api/register/app/ - regist_view :This is only for registration of students , this takes username email and password in a post request , and registers the student in data base.

- url:/api/login/dashboard/- This takes username and password of the instructor(i.e. TA or Professor) and if the authentication is successful it returns the authentication token.For the authentication to be successful user need to be a instructor and hence no student can log in to the dashboard.

- url:/api/login/app/- This takes username and password of the students and on successful authentication returns the authentication token. It also saves the fcm token in the database.

- url:/api/get_courses/: if a student is authenticated i.e. if the header of request contains authentication token than returns all the courses for username corresponding to the student.

- url:/api/join_course/: if a student is authenticated i.e. if the header of request contains authentication token and if body contains valid invitation code than adds the student in the corresponding course.

- url:/api/get_notifs/: For android app: if a student is authenticated i.e. if the header of request contains authentication token and if body contains valid course name than returns list of notifications in decreasing order of time.

- url:/api/get_notifsWeb/: For student website : if a student is authenticated i.e. if the header of request contains authentication token and if body contains valid course name than returns list of notifications in decreasing order of time.

- url:/api/inst/get_notifs/: For instructor: if a instructor is authenticated i.e. if the header of request contains authentication token and if body contains valid course name than returns list of notifications in decreasing order of time.

- url:/api/inst/get_courses/: if a instructor is authenticated i.e. if the header of request contains authentication token than returns all the courses for username corresponding to the instructor.

- url:/api/add_course/: if a instructor is authenticated i.e. if the header of request contains authentication token than a course will be added in the database with a unique course name and invitation code as received in

- url:/api/get_notif_details/: if a student is authenticated i.e. if the header of request contains authentication token and body contains notification id, coursename then it returns all the details stored as a response.

- url:/api/get_notif_detailsWeb/: For student website : if a student is authenticated i.e. if the header of request contains authentication token and body contains notification id, coursename then it returns all the details stored in the response.

- url:/api/inst/get_notif_details/: if a instructor is authenticated and body contains coursename and notification id then returns all the details stored in the response

- url:/api/send_notif/: For instructor: if a instructor is authenticated and body contains coursename and notification details then send the notification to all the students enrolled in the course.

- url:/api/getStudentsTas/: For instructors website: if a instructor is authenticated and body contains coursename then returns all the students and TAs stored in the response.

- url:/api/getStudents/: For instructors app: if a instructor is authenticated and body contains coursename then returns all the students stored in the response.

- url:/api/getTas/: For instructors app: if a instructor is authenticated and body contains coursename then returns all the TAs stored in the response.

- url:/api/addTa/: For instructors : if a instructor is authenticated and body contains username of a user which has a ta account than it will be added in the database.

- url:/api/removestudent/: if a instructor is authenticated and body contains coursename ,username of a user which is a enrolled student in the given course then it will be removed from that course.

- url:/api/changepassword/: if a request body contains correct username and corresponding password with a new password then password is changed in the database

- url:/api/logout/ if a student is authenticated then on this request fcmtoken is cleared from the database so no further notification is sent.

- url:/api/otp/ On this request username and email are matched then a otp is sent to the email.
- url:/api/confirmotp/ In this request otp is matched and new password is set in the user database.
- url:/api/verify/ This verifies whether username provided is instructor of the given coursename.

3.2 FLUTTER

For implementation of all the components, We have sent HttpRequest to the Django Back-end and receive the desired data in json format, which is further processed to get the functional form. We are using built-in Django token authentication for authenticating the users i.e. requests after logging in requires authentication token received at the time of log-in to be present in the request headers. Brief description of the features:

- **Login Page**-We have developed a login page which authenticates the user from django back-end and stores user model(username,auth-token) in the local device database and hence remembers the login session unless explicitly logged out by the user.
- **Sign Up Page**-A user can sign up after entering a unique username along with email and password.This email is used for resetting the password in case the user forgets.(This feature is exclusive for student app)
- **Forgot Password**- A user can reset the password after validating the otp sent on registered email.(This feature is exclusive for student app)
- **Home Page**- This page displays the list of all the courses in which the user is registered.This page also gives (student app)students an option to register in a course and (dashboard app) prof to create a course.
- **Logout button**-By clicking the user is logged out,(student app)fcm token is deleted from database to ensure that the device does not receive notification in logged out state.
- **change password**-A user can change the password after entering correct username and password.
- **Send Notification Page**- Provides a form through which instructor/TA can send the notification to students enrolled in that course. Also it contains a drop down list in the app bar through which we can navigate to further pages like Notification List Page, TA list page and Students list page.
- **Notification List Page**-This page gives the list of notifications for a given course. Notifications have been distinguished based on priorities and acknowledgement using different font styles. (student app) On clicking a notification it

navigates to complete details of notification.(dashboard app)On clicking a notification it navigates to the acknowledgement page.Here by clicking on "i" button,details for a notification can be seen.

- **TA List**- Displays the list of TA's in that particular course. This page also gives professor an option to add a TA.
- **Student List**- Displays the list of students enrolled in that course. Professor (Not TA) has the permission to even remove the student by directly pressing the list tile containing the student name. }

3.3 ANGULARJS

- For communication of Front-end with back-end we have made use of Rest-Api's and we are securing the Api's by using Django built-in authentication system.
- Basically whenever a registered user do login it gets a token from backend which is encoded by a Hash algorithm and is unique for every user.
- Any Api will only be accessible when this token has been sent as a Header in the post request.

Brief description of the features is as follows :

- **Login Page**-Developed a login page which authenticates the user from django back-end and stores user model (username,authentication-token) in session storage and hence remembers even when we refresh a page.
- **Signup Page**-A user can sign up after entering a using username along with email and password.This feature is exclusive for student website ,Instructors has to register on django built-in admin panel.
- **Home Page**-This page displays the list of all the courses in which the user is registered.This page also give Profs (dashboard) an option to create a course and students (students website) an option to register in a course.It also have buttons like logout (which removes the auth-token and redirect to login page) and change-password (which prompts the user to enter username,old password,new password).
- **Courses Page**-This page gives option of adding a TA or removing a student to Profs(Not allowed to TA) along with options to Send a Notification to the users registered in the course , get list of users in the course and see list of previous notifications sent by instructors .In students website only list of notifications is seen.
- **Notification Page**-This pages gives details of a particular notification like title , content , time , sent by . In Dashboard we also get to see list of students who have acknowledged that notification and it students website we get to see whether the notification was urgent or not.

- **Routing**-We have implemented authentication guards so that only logged users are able to see main website and unregistered users are not able to manipulate url's.

4 References

- https://ohuru.tech/blog/2020/4/19/flutter-signuplogin-application-with-django-backend-3/
- https://simpleisbetterthancomplex.com/
- https://medium.com/flutter-community/flutter-push-pop-push-1bb718b13c31
- https://flutter.dev/docs/cookbook/navigation/navigation-basics
- https://medium.com/@jun.chenying/flutter-tutorial-part3-push-notification-with-firebase-cloud-messaging-fcm-2fbdd84d3a5e
- Many short tutorials on youtube.