eYSIP2018

e-Yantra App

Interns: Manan Anand, Ekagrata
Mentors: Vikrant Fernandes, Tushar Shah,Omkar Manjrekar
Duration of Internship: 08/06/2018 – 13/07/2018

e-Yantra App

Abstract

e-Yantra Community is expanding at an exponential rate. Its important for this creative community to work together and make the world a better place to live. The project aims to build an app that can bring this community together and connect with each other. Working together will always benefit the society as well as the individual.

Completion status

- Completed Google Sign-in through the android phone.
- Setup a Server on Amazon Web Services
- Server Side scripting setup using laravel
- Integrated a News Feed Page into the app.
- Successfully Integrated a chat forum for better communication between various people.
- Talks and videos integrated into the app.
- e-LSI labs across the country were shown on a map and information for anyone wanting to setup a lab is provided.
- Different projects under e-YRDC are fetched from the server
- Calendar integrated into the app for events and deadlines. Data for which is fetched from the server Database.
- Integrated Firebase Cloud Messaging into the app for notifications about different events or deadlines.

- Integrated search functionality into the app for better navigation in the app.

1.1 Software used

- **Android Studio**
- Version:3.1 Download Android Studio,
- Installation steps:
  - For Windows Users:
    - If you downloaded an .exe file (recommended), double-click to launch it.
      If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio ¿ bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).
    - Follow the setup wizard in Android Studio and install any SDK packages that it recommends.
  - For Linux Users:
    - Unpack the .zip file you downloaded to an appropriate location for your applications, such as within /usr/local/ for your user profile, or /opt/ for shared users.
    - To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory, and execute studio.sh.
    - Select whether you want to import previous Android Studio settings or not, then click OK.
    - The Android Studio Setup Wizard guides you though the rest of the setup, which includes downloading Android SDK components that are required for development.
- **Amazon EC2** Get Started,
- Getting Started with the AWS Management Console
  - Set up and log into your AWS account
  - Launch an Amazon EC2 instance
  - Configure your instance

  - Connect to your instance
  - Terminate instances

- **Xampp (on server)**
- Version: 7.2.7 Download,
- **Laravel**
- Version: 5.6 Install
- Installation steps:
  - For Windows and Linux Users:
    * Laravel utilizes Composer to manage its dependencies. So, before using Laravel, make sure you have Composer installed on your machine.
    * First, download the Laravel installer using Composer:
      composer global require "laravel/installer"
    * Alternatively, you may also install Laravel by issuing the Composer create-project command in your terminal:
      composer create-project –prefer-dist laravel/laravel blog

1.2 Software and Code

Github link for the repository of code

The Software involved can be divided into 2 major parts:

- Android(Front End)

  Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers.

  - Retrofit

```
1 public interface GitHubService {
2   @GET("users/{user}/repos")
3   Call<List<Repo>> listRepos(@Path("user") String
      user);
4 }
```

Snippet 1.1: Retrofit turns your HTTP API into a Java interface

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com/")
    .build();

GitHubService service =
    retrofit.create(GitHubService.class);
```

Snippet 1.2: The Retrofit class generates an implementation of the GitHubService interface.

```
Call<List<Repo>> repos = service.listRepos("octocat");
```

Snippet 1.3: Each Call from the created GitHubService can make a synchronous or asynchronous HTTP request to the remote webserver.

– Google SignIn

```
private class VerifySignIn extends AsyncTask<Void,
    Void, String> {

        public VerifySignIn() {

        }

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
        }

        @Override
        protected String doInBackground(Void... voids)
    {
            GoogleSignInOptions gso = new
    GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                    .requestEmail()
                    .build();
            googleSignInClient =
    GoogleSignIn.getClient(MainActivity.this,gso);
            GoogleSignInAccount account =
    GoogleSignIn.getLastSignedInAccount(MainActivity.this);
            if(account == null){
                return "n";
            }else{
                return "y";
            }
        }

```

```
26          @Override
27          protected void onPostExecute(String s) {
28              super.onPostExecute(s);
29              if (s.equals("n")){
30                  signIn.setVisibility(View.VISIBLE);
31                  signinText.setVisibility(View.VISIBLE);
32              }else if(s.equals("y")){
33                  Intent intent= new
    Intent(MainActivity.this,Main2Activity.class);
34                  Bundle bundle= new Bundle();
35              bundle.putString("name",name);
36              bundle.putString("email",email);
37              bundle.putString("photo",photo);
38              intent.putExtras(bundle);
39               startActivity(intent);
40                  finish();
41              }
42          }
43      }
44
```

Snippet 1.4: Verifying whether the user is already signed in through google or not. If yes redirecting to the apps landing page else popping up the google sign in button.

– Youtube Integration

```
1  public class YoutubePlayerActivity extends
       YouTubeBaseActivity {
2       YouTubePlayerView youTubePlayerView;
3       YouTubePlayer.OnInitializedListener
       onInitializedListener;
4       String url;
5
6       @Override
7       protected void onCreate(Bundle savedInstanceState)
       {
8           super.onCreate(savedInstanceState);
9
       setContentView(R.layout.activity_youtube_player);
10          Intent intent= getIntent();
11          Bundle bundle= intent.getExtras();
12           url= bundle.getString("url");
13
14
15          youTubePlayerView =
       findViewById(R.id.youtubeplayerview);
16
```

```
17              onInitializedListener = new
    YouTubePlayer.OnInitializedListener() {
18                  @Override
19                  public void
    onInitializationSuccess(YouTubePlayer.Provider
    provider, YouTubePlayer youTubePlayer, boolean b) {
20                      youTubePlayer.setFullscreen(true);
21                      youTubePlayer.loadVideo(url);
22                  }
23
24                  @Override
25                  public void
    onInitializationFailure(YouTubePlayer.Provider
    provider, YouTubeInitializationResult
    youTubeInitializationResult) {
26                      Log.d("youtube","Initialization
    failure");
27
28                  }
29              };
30
    youTubePlayerView.initialize("@string/api_key",
    onInitializedListener);
31      }
32  }
```

Snippet 1.5: Activity extending YouTubeBaseActivity and integrating Youtube player View to play the Youtube video directly into the app.

– Google Maps

```
1
2   public class MapsActivity extends FragmentActivity
      implements OnMapReadyCallback {
3
4       private GoogleMap mMap;
5
6
7       @Override
8       protected void onCreate(Bundle savedInstanceState)
      {
9           super.onCreate(savedInstanceState);
10          setContentView(R.layout.activity_maps);
11
12          SupportMapFragment mapFragment =
      (SupportMapFragment) getSupportFragmentManager()
13                  .findFragmentById(R.id.map);
14          mapFragment.getMapAsync(this);
```

```
15          clgs= new ArrayList<>();
16
17      }
18
19
20      @Override
21      public void onMapReady(GoogleMap googleMap) {
22          mMap = googleMap;
23
24
25          LatLng base = new LatLng(22.9734, 78.6569);
26
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(base,4));
27
28          fetchColleges();  // functions for fetching
    colleges and adding markers on them.
29          mMap.setOnMarkerClickListener(new
    GoogleMap.OnMarkerClickListener() {
30              @Override
31              public boolean onMarkerClick(Marker
    marker) {
32
33                  // Displaying Dialog Box showing College
    information
34                  return false;
35              }
36          });
37
38      }
39  }
```

Snippet 1.6: MapsActivity implementing OnMapReadyCallback. And popping up markers on appropriate lat longs.

– Downloading

```
1
2   download= findViewById(R.id.download);
3           download.setOnClickListener(new
    View.OnClickListener() {
4               @Override
5               public void onClick(View view) {
6
7                   Intent shareIntent = new Intent();
8
    shareIntent.setAction(Intent.ACTION_VIEW);
9                   String url =
    "http://www.e-yantra.org/img/e-Yantra_Pamphlet2018.pdf";
```

```
10                  shareIntent.setData(Uri.parse(url));
11
    startActivity(Intent.createChooser(shareIntent,"Choose
    your Browser"));
12  });
13
14
15  \item Mail
16  \begin{lstlisting}[language=java,caption={Directly
    mailing at a particular email address using intent
    filters}]
17
18   mailus= findViewById(R.id.mailus);
19         mailus.setOnClickListener(new
    View.OnClickListener() {
20              @Override
21              public void onClick(View view) {
22                  Intent shareIntent = new Intent();
23
    shareIntent.setAction(Intent.ACTION_SENDTO);
24
    shareIntent.setData(Uri.parse("mailto:helpdesk@e-yantra.org"));
25
26
    shareIntent.putExtra(Intent.EXTRA_SUBJECT,"For
    setting up lab");
27
28
    startActivity(Intent.createChooser(shareIntent,"Open
    Mail"));
29              }
30          });
```

Snippet 1.7: Downloading Pamphlets and PDFs using intent filters

- Laravel(Back End)

  Its a PHP Framework saving a lot of time designing the code from scratch. its advantages are

  - Building Authentication and Authorization Systems
  - URL Routing Configuration
  - Configuration Error and Exception Handling
  - Scheduling Tasks Configuration and Management

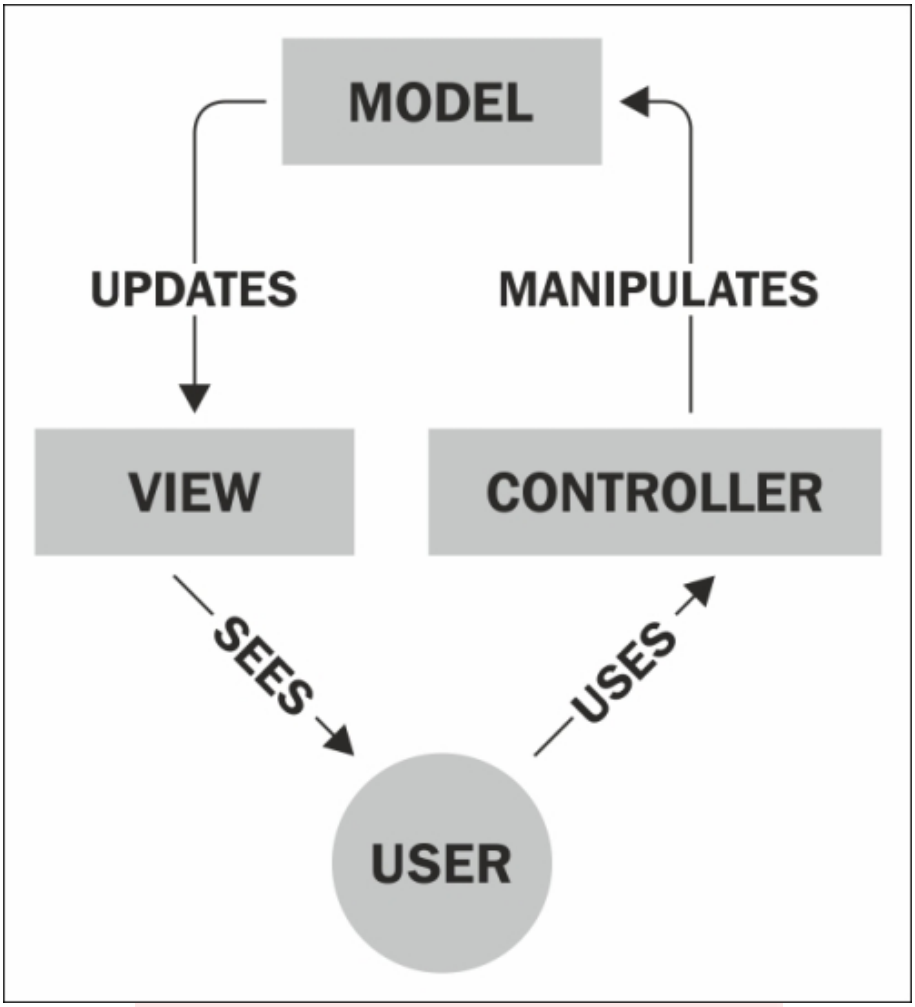

Figure 1.1: It works on MODEL VIEW CONTROLLER

```
Route::get('/', function () {
    return view('welcome');
});
Route::get('maps','Eyantra\ResourceController@latlong');
//Route for Maps API
Route::get('youtube','Eyantra\ResourceController@youtube');
//Route for Youtube API
Route::get('projects','Eyantra\ResourceController@project');
//Route for Projects API
Route::get('tutorials','Eyantra\ResourceController@tut');
//Route for Tutorials API
```

Snippet 1.8: Routes can easily be defined. These Routes here show different API's created for the android app.

```
class CalenderController extends Controller
{
  public function show(Request $request)
  {
    $input=$request->date;
    $data = Calender::where('Date',"$input")->get();
    echo json_encode($data);
  }
}
```

Snippet 1.9: This is how a controller function looks like. This function is called when the user hits on the corresponding Route. All the calculations and assigning of variables can take place here.

```
class Comment extends Model
{
    protected $table='comments';
}
```

Snippet 1.10: This is how a model is connected to the table in the database.

1.3 Use and Demo

- The User Interface Of the app has been designed in android studio. It has an easy to use interface.
- The navigation drawer makes it easy for the user to navigate to the various sections of the app.
- The app can also be used for knowing various e-LSI labs across India through the map already integrated into the map.
- Various tutorials for Firebird V and different electronic devices can be found on the app.
- Latest news on technological advancements and report of important events already occured at e-yantra can be found on the news feed.

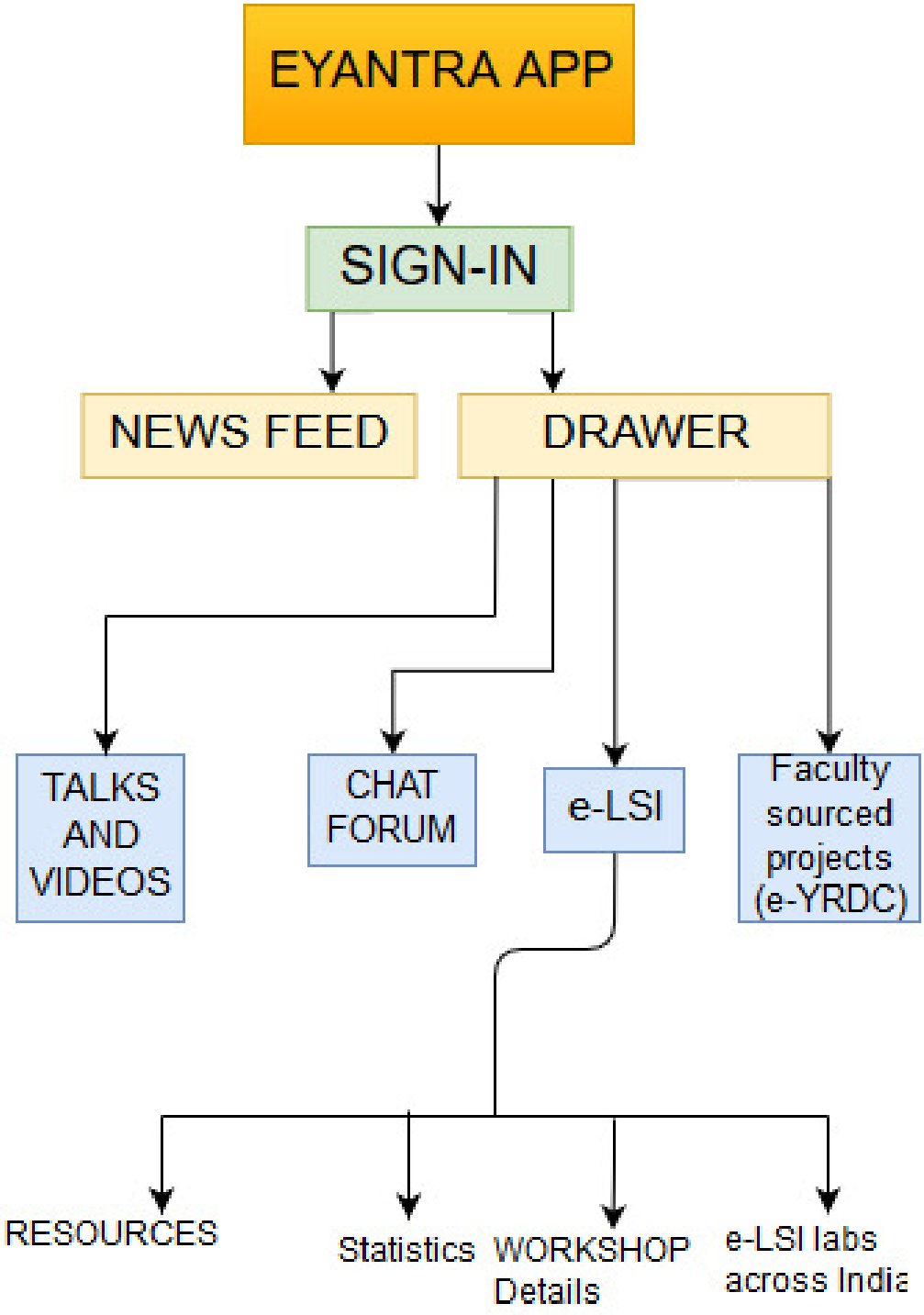

Figure 1.2: Flow of the App

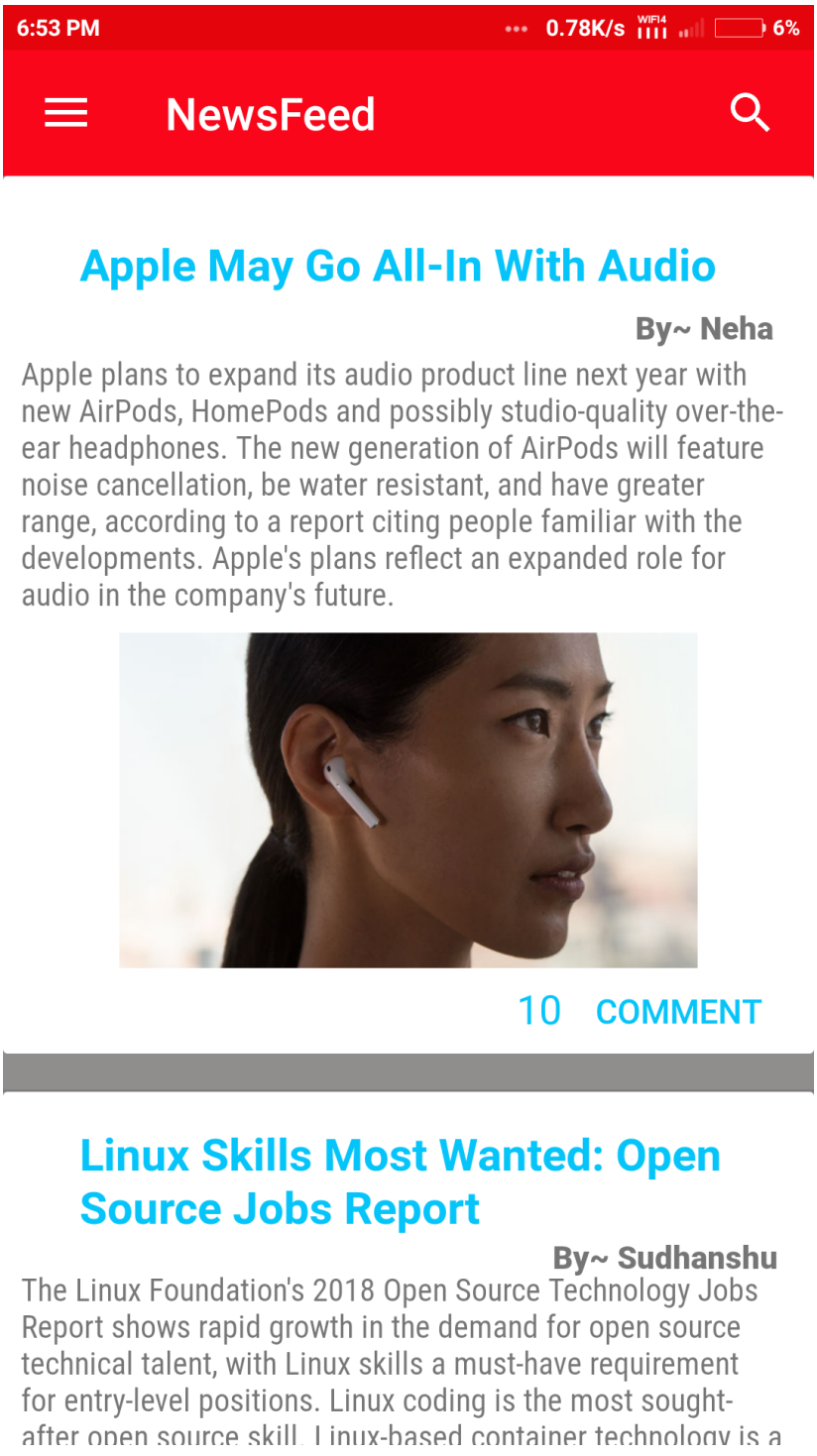

Figure 1.3: News Feed in the app where latest news about technology can be found.

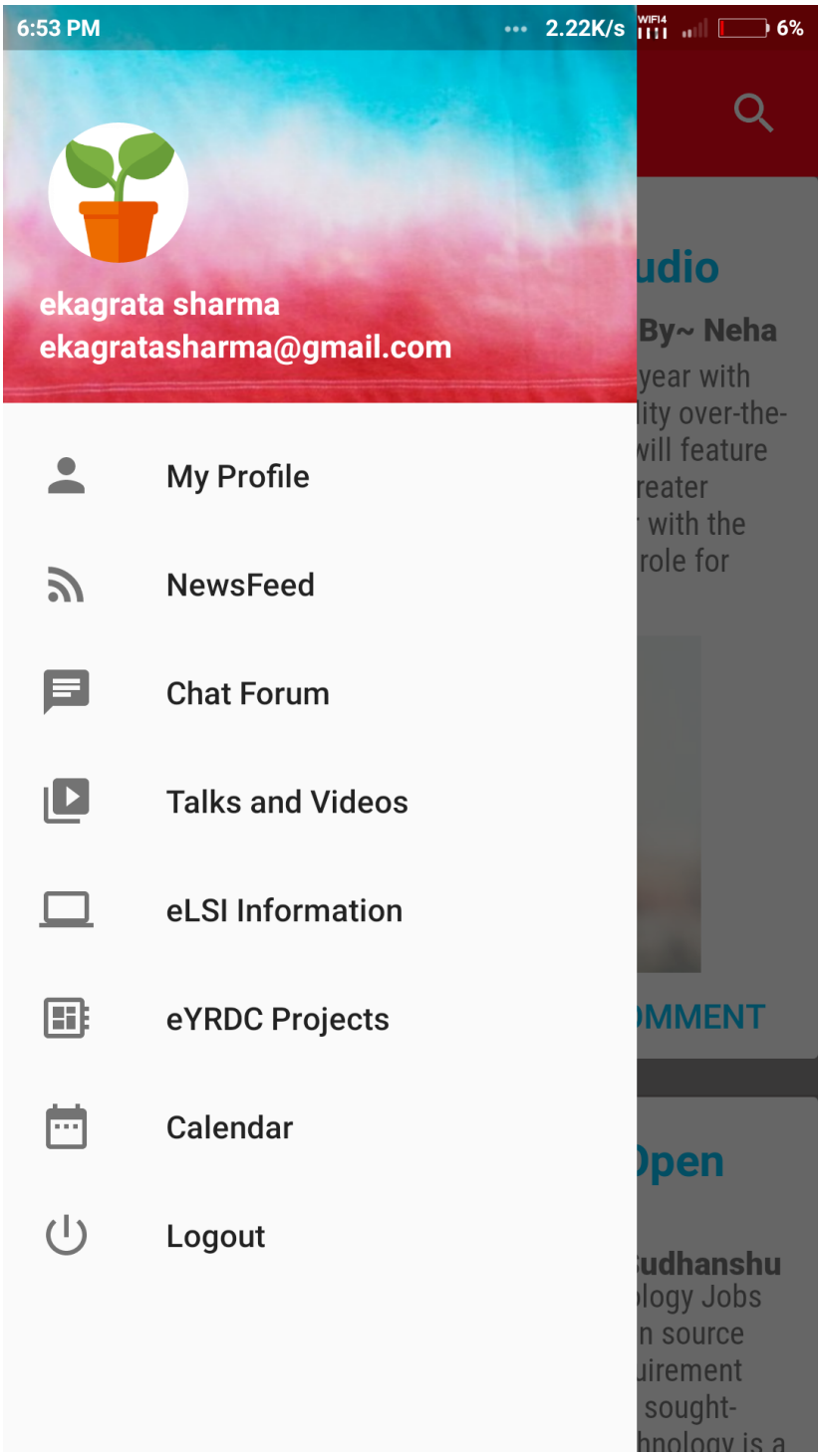

Figure 1.4: This drawer makes it easy for the user to access all the different areas of the app.

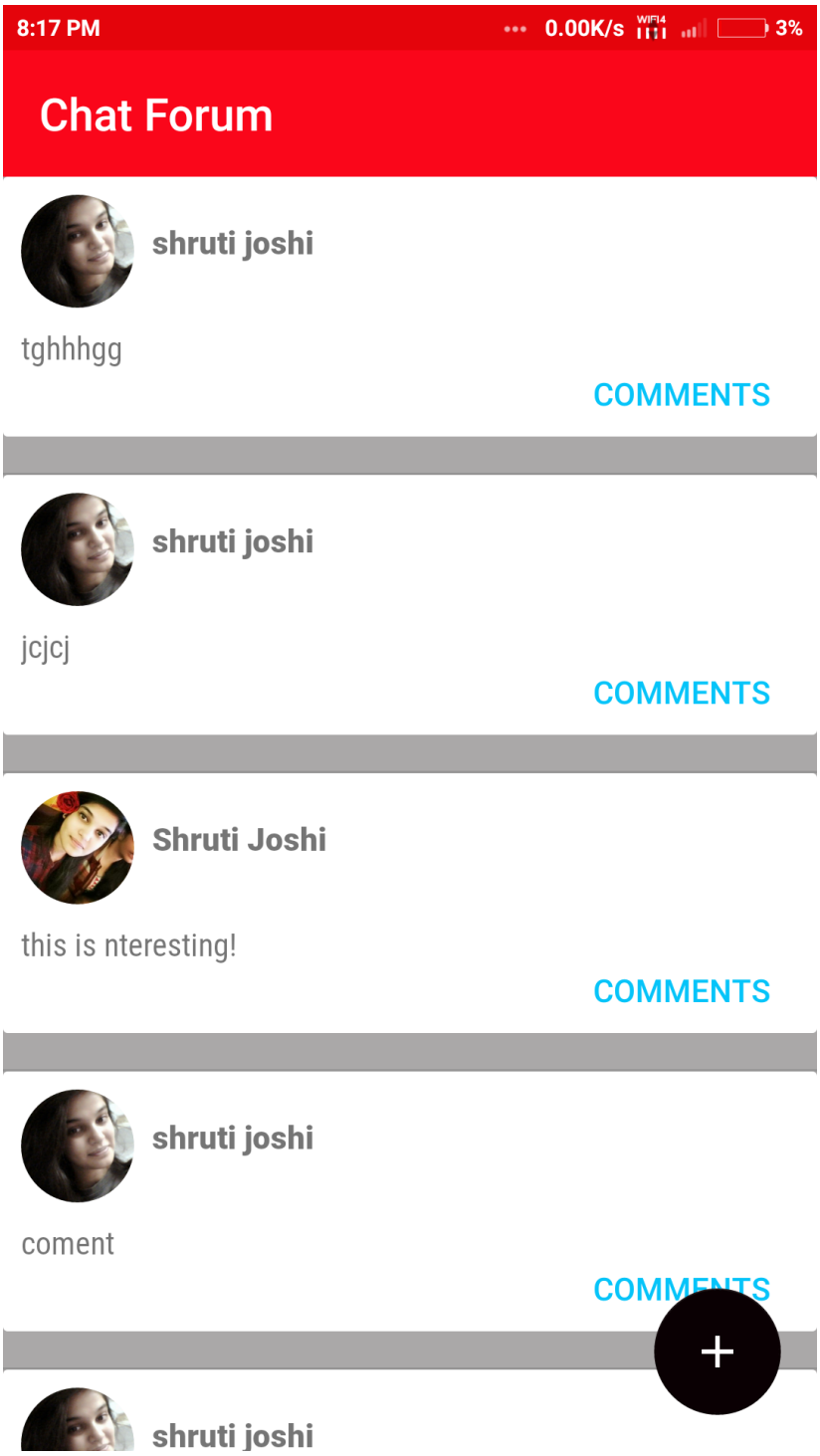

Figure 1.5: The app has an inbuilt chat forum which can be used for discussions and questions answers between the e-yantra community.

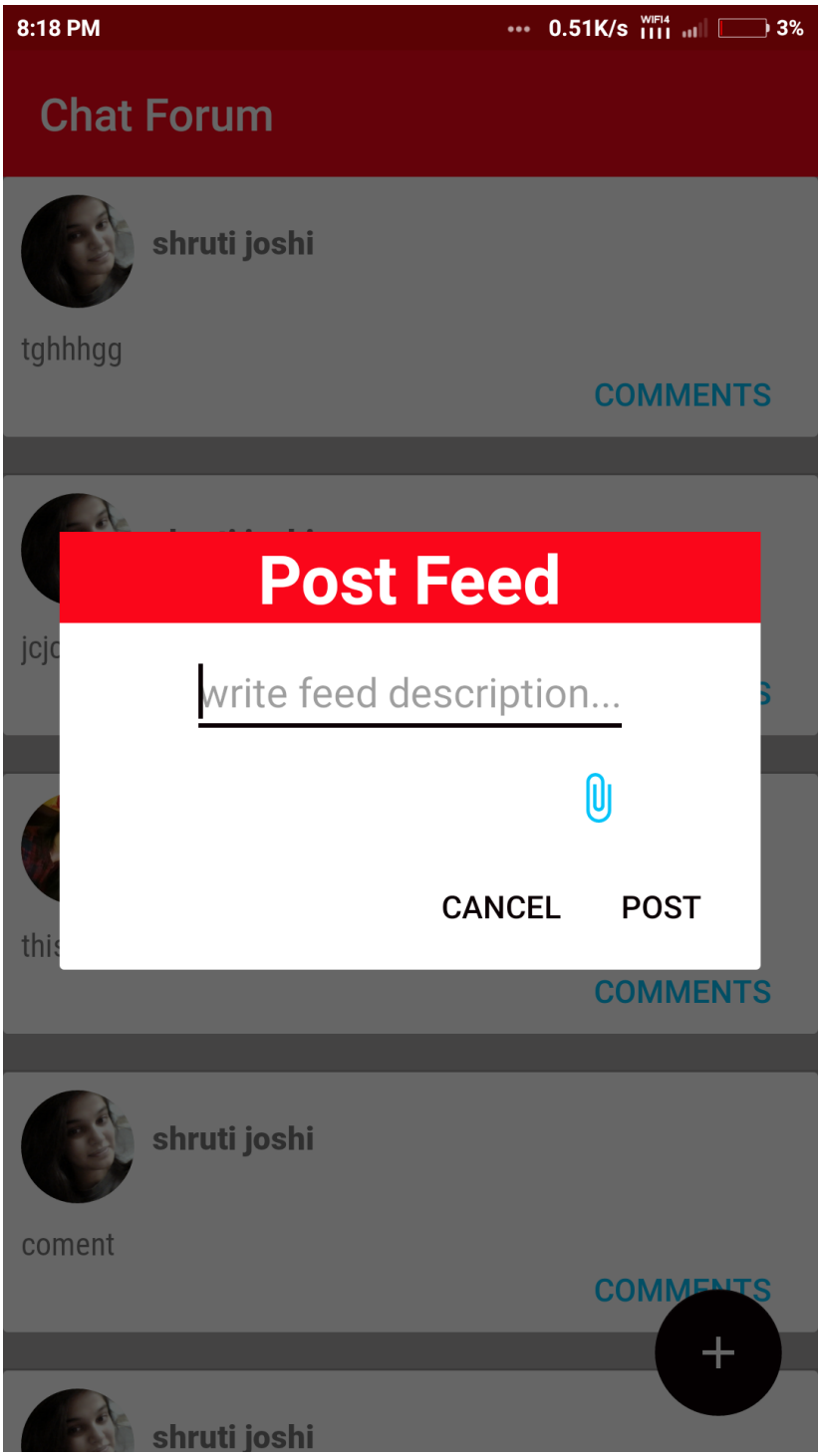

Figure 1.6: A dialog box enables the user to enter in his text and the post the chat forum.

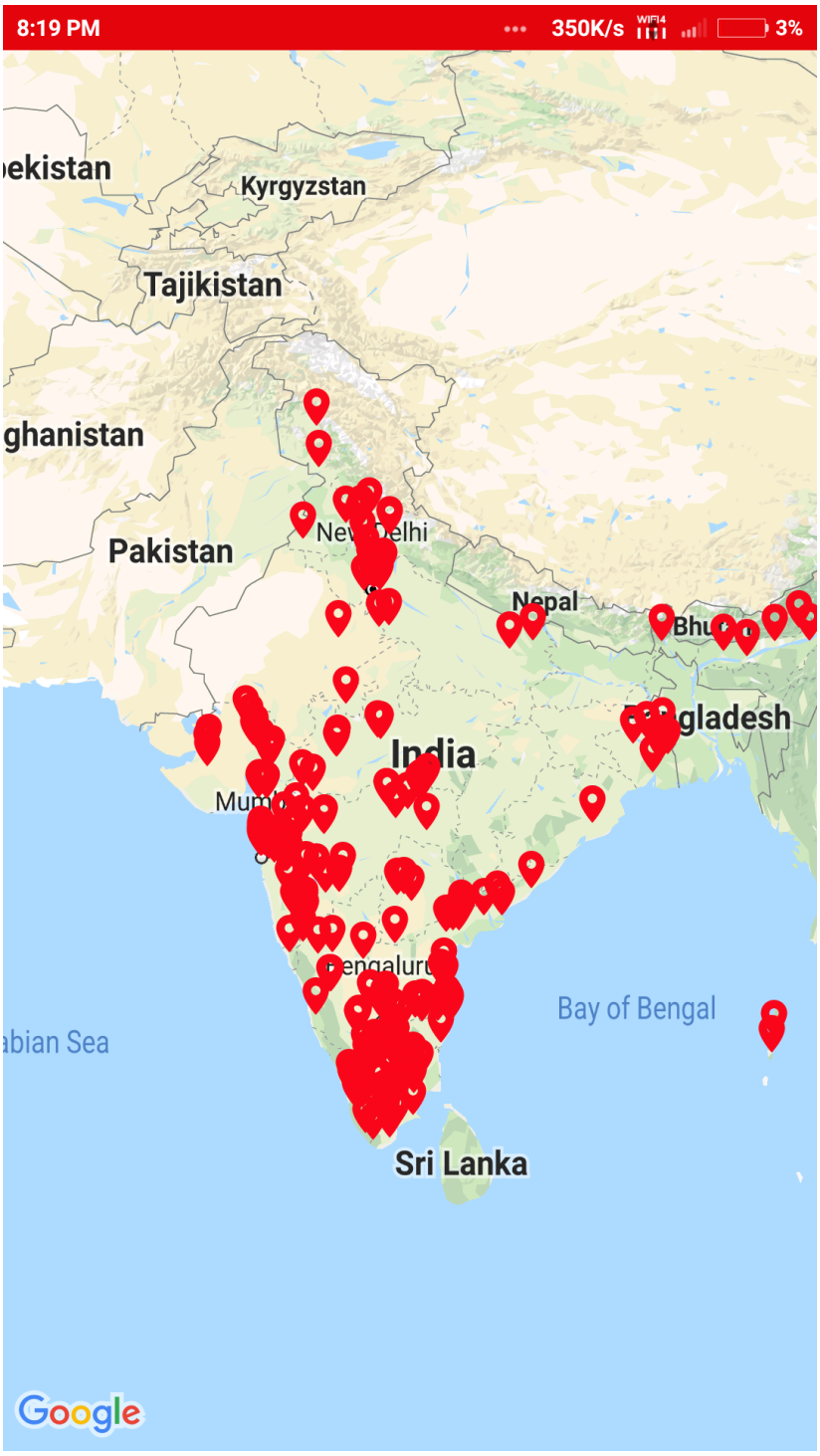

Figure 1.7: All the elsi labs across India can be found on this map integrated into the app.

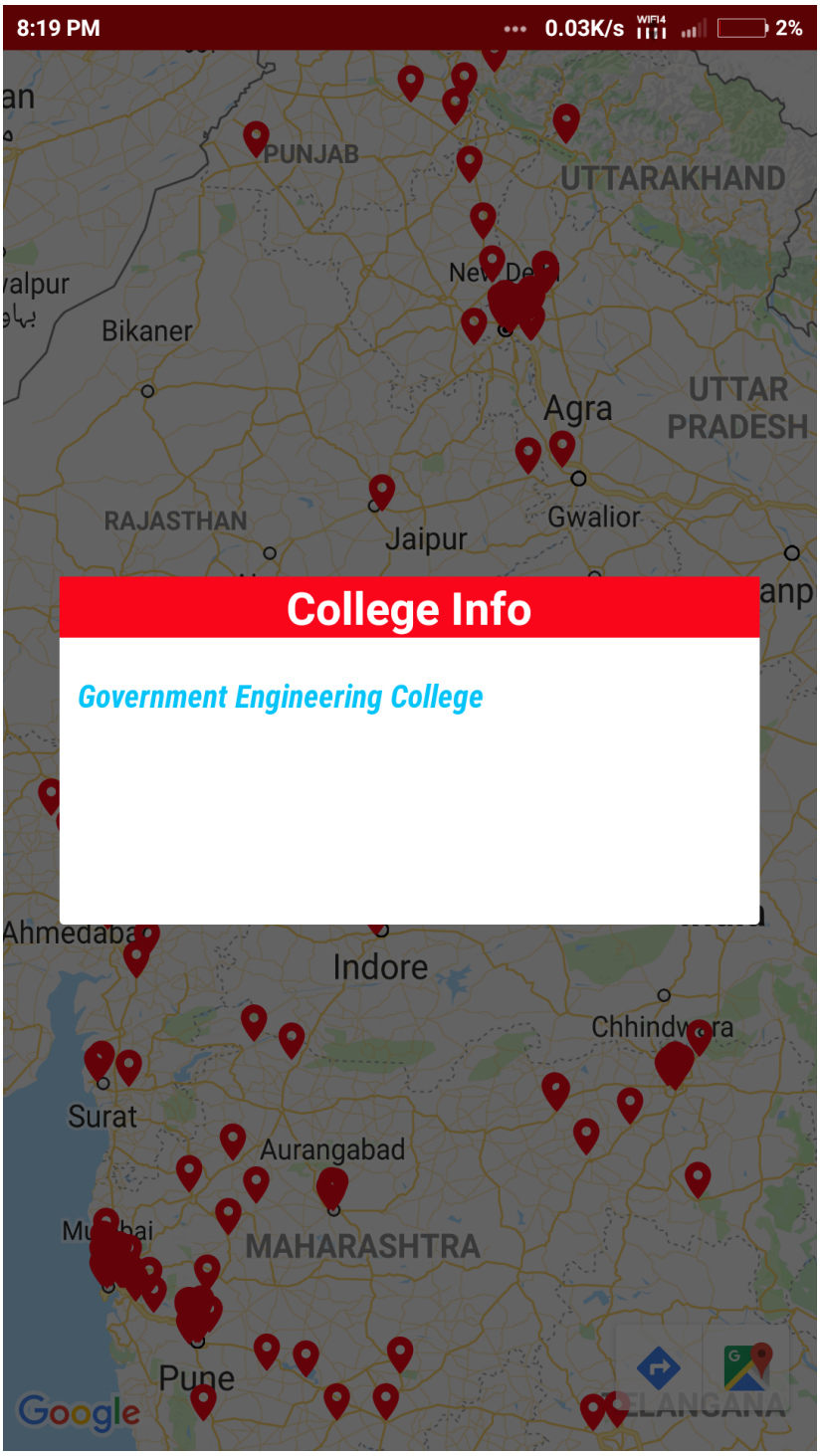

Figure 1.8: User can easily know the details of the elsi college and then can access the resources at the college or participate actively in the lab activities.