MAY 6, 2017

CSCE-590 PROJECT

WORKFLOW MANAGEMENT

IPPILI AASHISH
C00220927/axi1334@louisiana.edu

Introduction

In this project, I developed application regarding user work flow management using angular 2, a user will be submitting his files and upload it to server. Here I am using nodeJS as the server to store files. I am using mongo DB to store user credentials and the files he has submitted. Using expressJS I generated rest URLs such that user can log in, signup and upload files.

Once a user submits his files, there exists an approver who will verify the files and provide the required comments to the submitted files by the user. I have added support excel files where I face issues while downloading the file. But later achieved it without any errors in the downloaded file.

Frameworks

- Angular 2
- Mongoose (JavaScript framework for accessing MongoDB)
- Node JS
- Express JS

Angular 2 comes with NPM package so ‘**`npm install -g angular-cli`**’ is the command that is used to setup an angular2 environment in the system.

To create a new angular 2 project, we use ‘**`ng new project_name`**’

Using ‘**`ng serve`**’ angular 2 app will be started.

Angular 2:

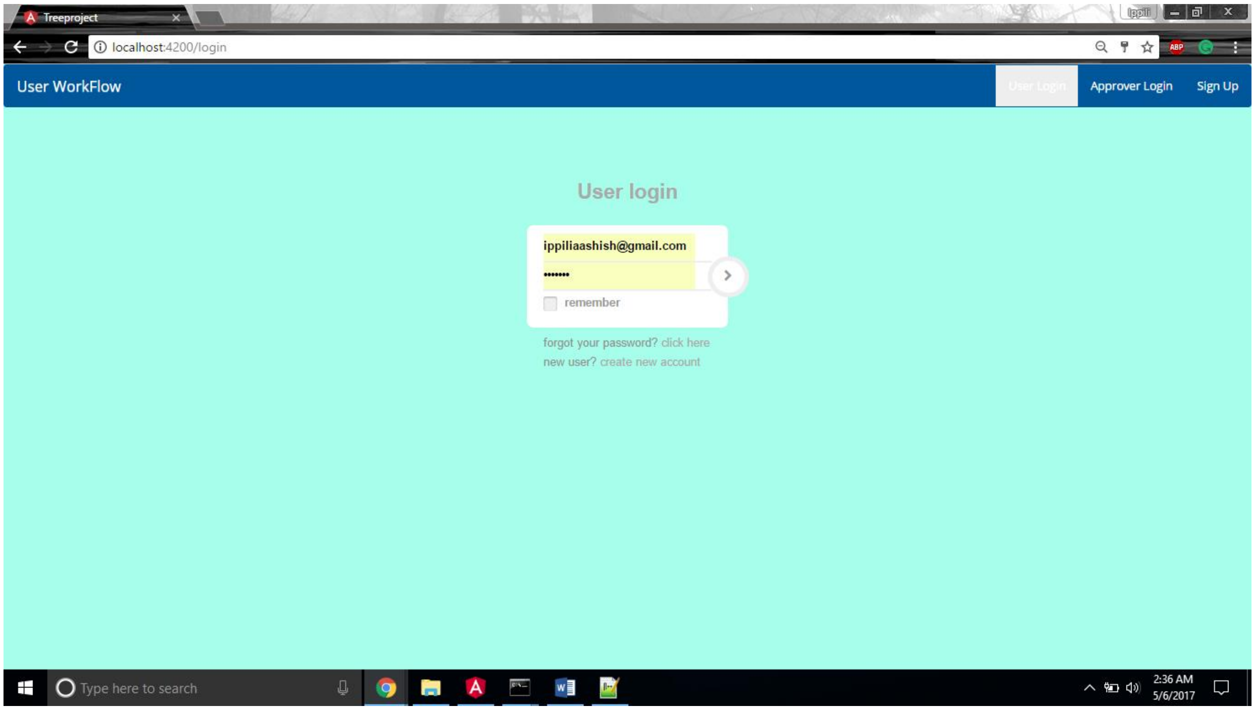
Angular applications are made up of components. A component is the combination of an HTML template and a component class that controls a portion of the screen. Every component begins with an @Component decorator function that takes a metadata object. The metadata object describes how the HTML template and component class work together.

```
@Component({
  selector: 'my-app',
  template: `<h1>Hello {{name}}</h1>`
})
```

The selector property tells Angular to display the component inside a custom <my-app> tag in the index.html.

The template property defines a message inside an <h1> header. At runtime, Angular replaces **{{name}}** with the value of the component's name property. Interpolation binding is one of many Angular features you'll discover in this documentation.

Screen shots of the application



User login

ippiliaashish@gmail.com

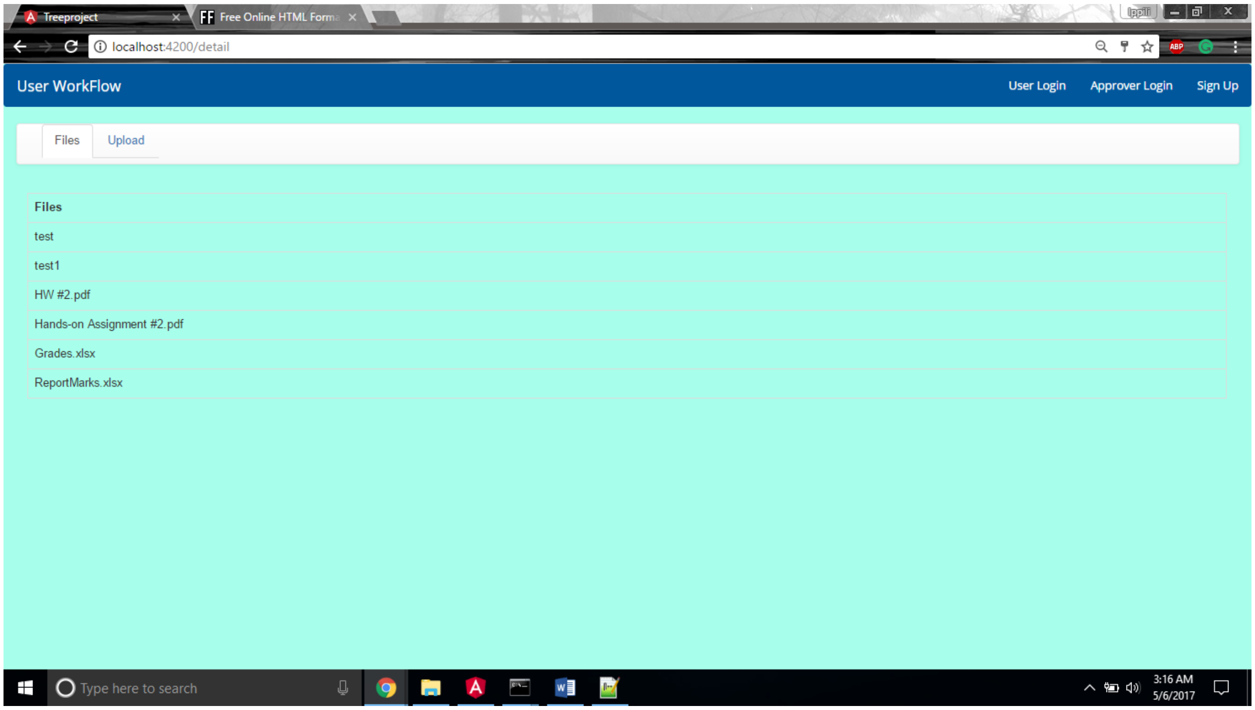
•••••••

□ remember

forgot your password? click here

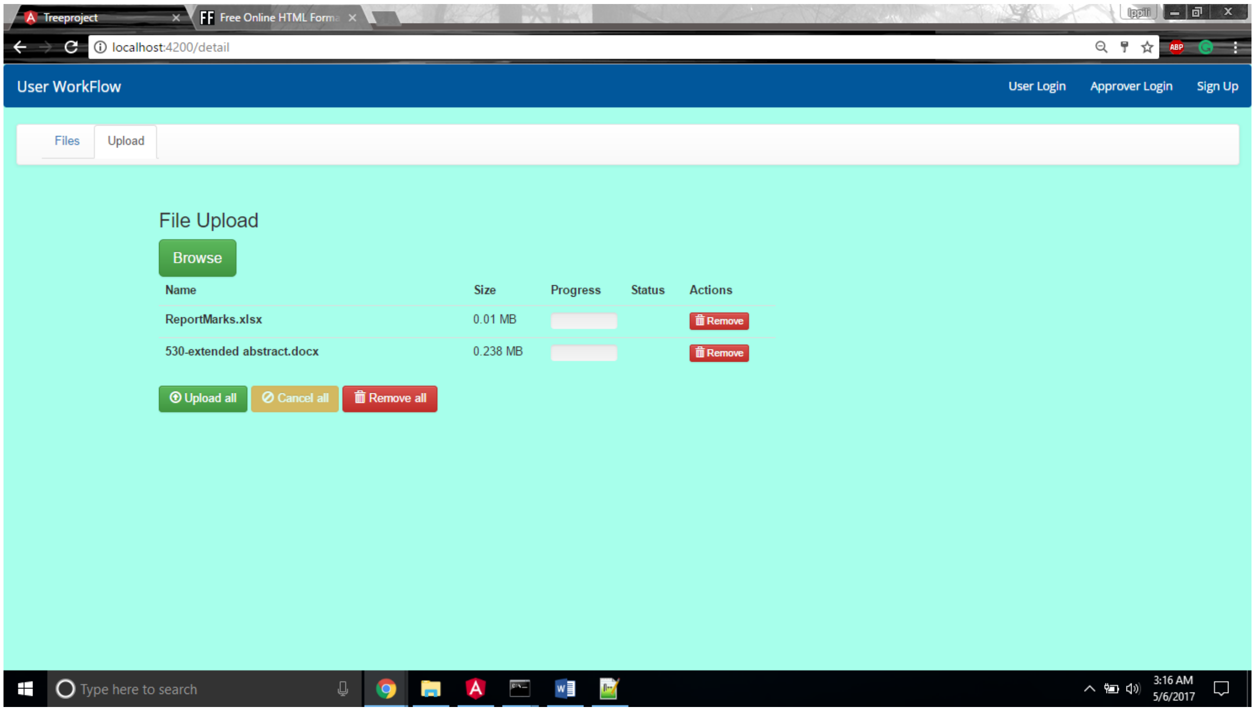
new user? create new account

I have provided screen for user login and approver login two separate screens.



Files
test
test1
HW #2.pdf
Hands-on Assignment #2.pdf
Grades.xlsx
ReportMarks.xlsx

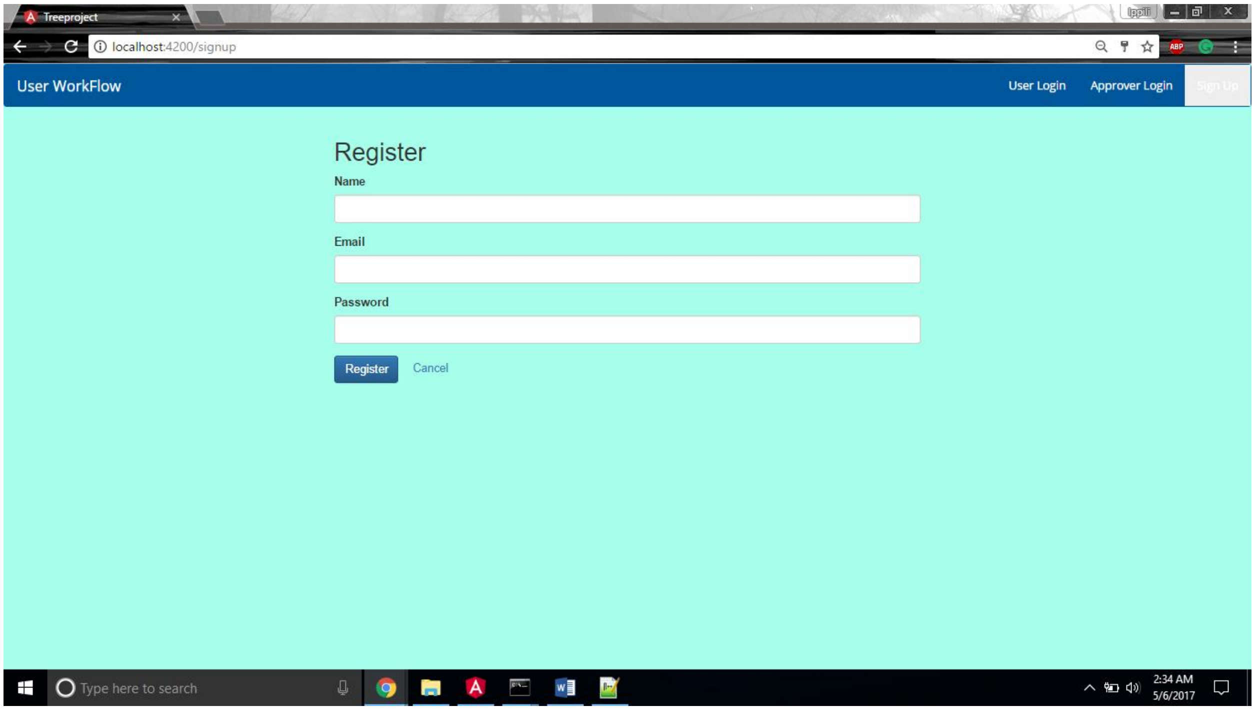
After user logs in he will see the list of files he uploaded



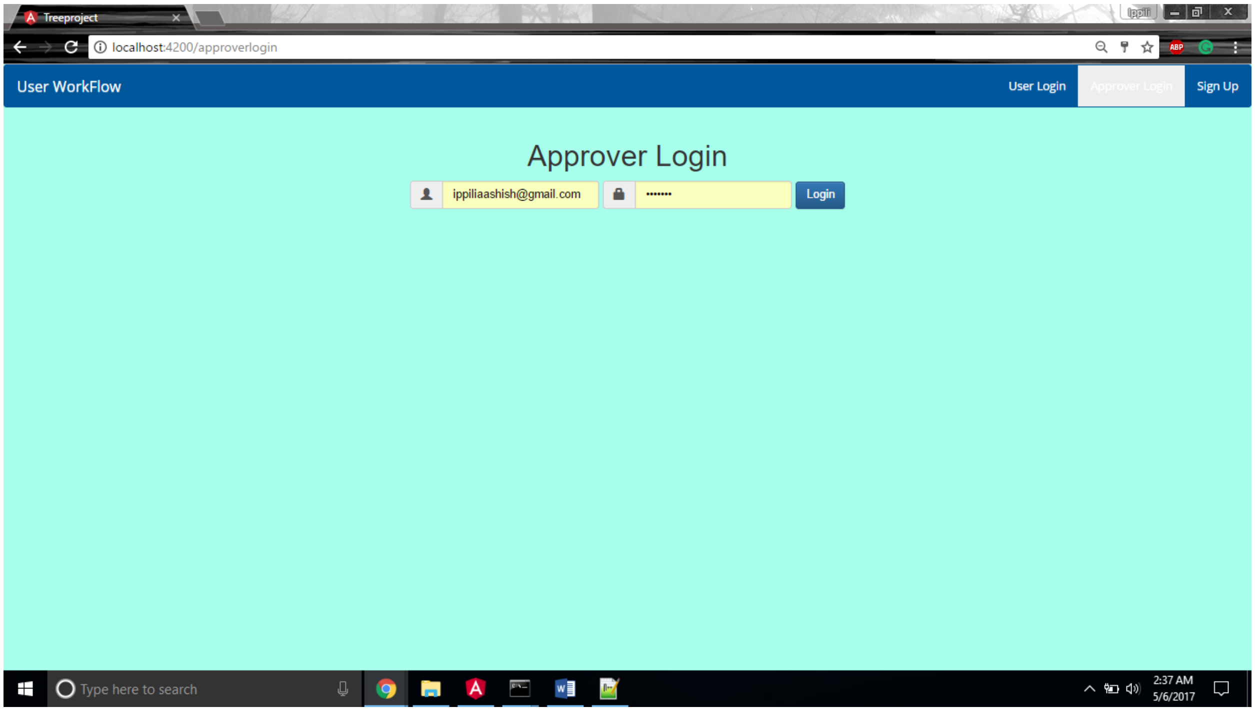
Name	Size	Progress	Status	Actions
ReportMarks.xlsx	0.01 MB			Remove
530-extended abstract.docx	0.238 MB			Remove

User can upload new files to the server by navigating to upload tab.

Signup screen



Signup screen for users we can make one of the users as approvers by changing the attribute value in database.

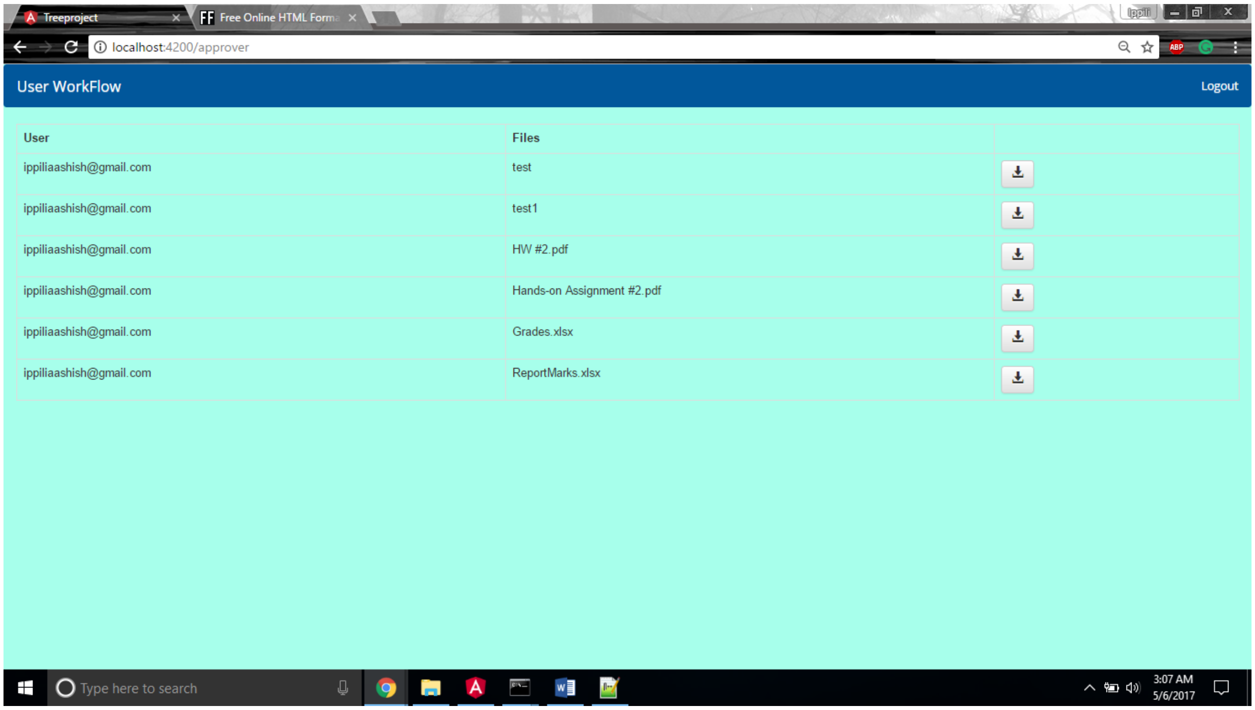


Approver Login

ippiliaashish@gmail.com •••••••

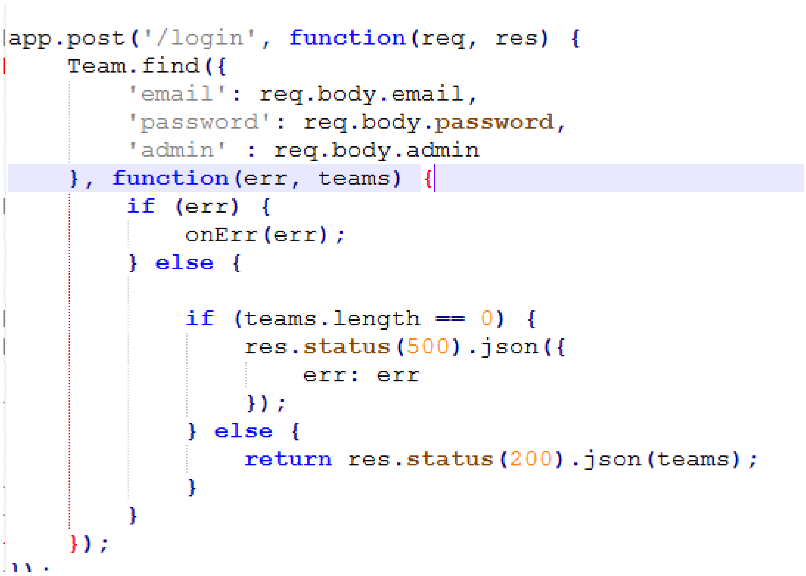
Login

Log in page for approver once the approver logs in he will see list of files submitted by users



User	Files	
ippiliaashish@gmail.com	test	⬇
ippiliaashish@gmail.com	test1	⬇
ippiliaashish@gmail.com	HW #2.pdf	⬇
ippiliaashish@gmail.com	Hands-on Assignment #2.pdf	⬇
ippiliaashish@gmail.com	Grades.xlsx	⬇
ippiliaashish@gmail.com	ReportMarks.xlsx	⬇

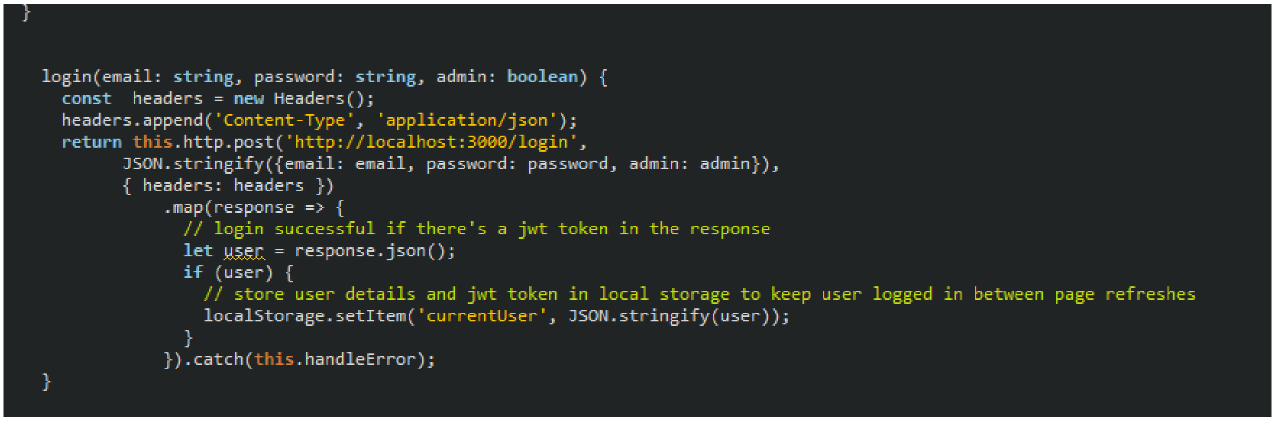
Once the approver logs in he will see the users and the list of files he uploaded.



```
app.post('/login', function(req, res) {
    Team.find({
        'email': req.body.email,
        'password': req.body.password,
        'admin' : req.body.admin
    }, function(err, teams) {
        if (err) {
            onErr(err);
        } else {

            if (teams.length == 0) {
                res.status(500).json({
                    err: err
                });
            } else {
                return res.status(200).json(teams);
            }
        }
    });
});
```

Login post API using express js when we passed the required parameters to this and if credentials are equal to the one that in the database user will be logged in. In the similar fashion, I have written rest calls for remaining actions like download, signup, and file upload.



```
  }

   login(email: string, password: string, admin: boolean) {
     const  headers = new Headers();
     headers.append('Content-Type', 'application/json');
     return this.http.post('http://localhost:3000/login',
            JSON.stringify({email: email, password: password, admin: admin}),
            { headers: headers })
                .map(response => {
                  // login successful if there's a jwt token in the response
                  let user = response.json();
                  if (user) {
                    // store user details and jwt token in local storage to keep user logged in between page refreshes
                    localStorage.setItem('currentUser', JSON.stringify(user));
                  }
                }).catch(this.handleError);
   }
```

This is the login method in service component we can use http.post to pass the login credentials and if it succeeds we will navigate to page.

We navigate to any page using router in angular. Each time code is modified angular server which is ng server refreshes the content in the web browser.