NeoScrum

Team JS

NeoSOFT Technology

Mumbai, Maharashtra

Overview

We have created a project through which users(developers) can give feedback to users(developers). User can see feedback given to them but they will not be able to see the name of the developer who has given feedback to them, that means the sender name is anonymous. In this, the first party can give feedback to the second party and the second party can also give feedback to the first party but they will not know the name of the sender. The Purpose of this is particularly helpful because it allows employees to speak their minds as freely as possible.

Goal

The main goal of this project is that the user will come to know their drawbacks and the area of improvement so that they can work on it.

Technologies Used

- ### Front end
    - The entire front is built with Angular 7. For CSS framework Bootstrap is used.

- ### Back end
    - The backend is in Node.js with Express framework by using PostgreSQL as a database.

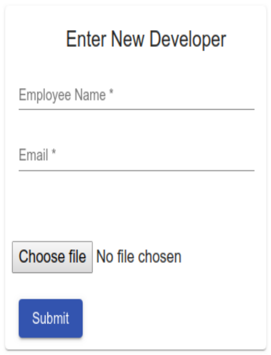
Technical Specification :

The technical specifications components of the NeoSCRUM website are listed and explained below.

1. Registration Page

The Register page has the following features.

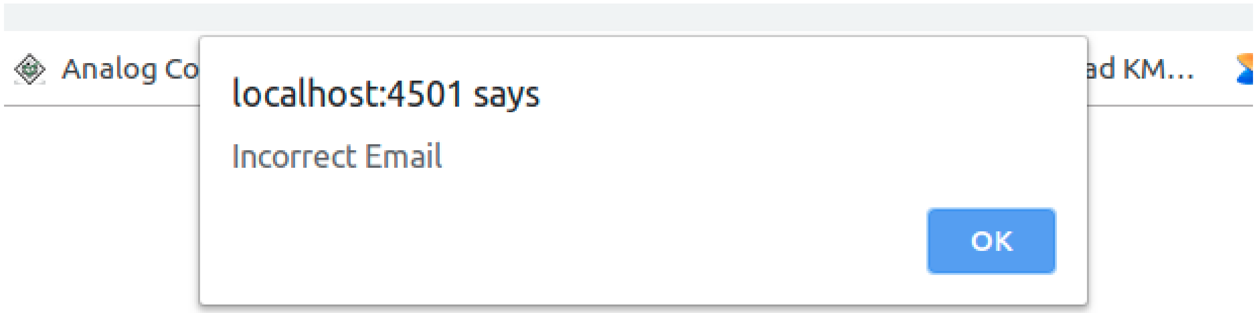
NeoSCRUM



The Login page has the following features.

- Input fields

  There are three input fields name, email and profile image as shown in the figure. Validation is also done if the user enters invalid inputs. It will show error popup.

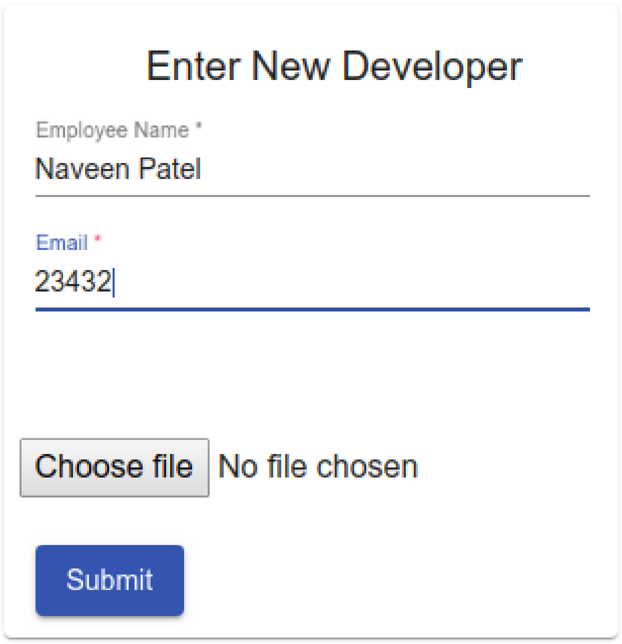


Analog Co

localhost:4501 says

Incorrect Email

ad KM...



Enter New Developer

Employee Name *

Naveen Patel

Email *

23432

- API to hit

  POST api/register For user registration

- Work-Flow

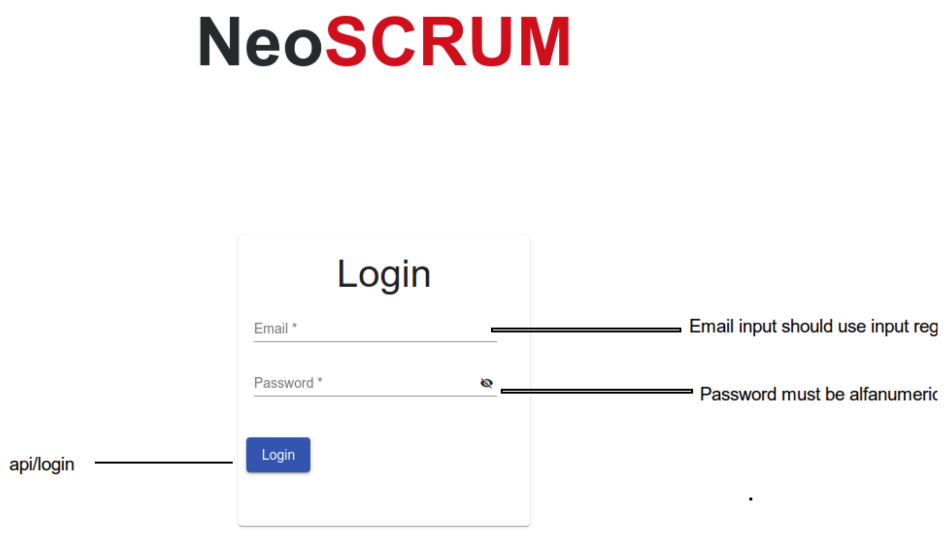
  As we know we have three input field user_name, user_email, and profile_image for user registration. Only the admin can register a new user. So whenever admin will insert the user data and press submit button, we will validate all the field that it is formatted or not, like user_name should be string, email id should be of proper format and profile_image should be of JPEG, JPG and PNG file format.

  If any of the above fields is not correct, an error popup will be shown. But if all field filled correctly then we will hit registration API and at the backend side we are also validating all field by using JOI (npm package) and If the data is correct then we will get a success response from API. But if it is not correct then we will get a failure(error) response.

  The unique thing is that we are not taking password while registering new user, we are generating it and sending the password to the email id of the user. For security reason, we are converting the password into the hashed format with the help of package bcrypt in backend and storing the hashed form in the database.

  After successful registration, welcome mail automatically sent to the user (by using Sendmail npm package in the backend). It contains login credentials and login page's link.

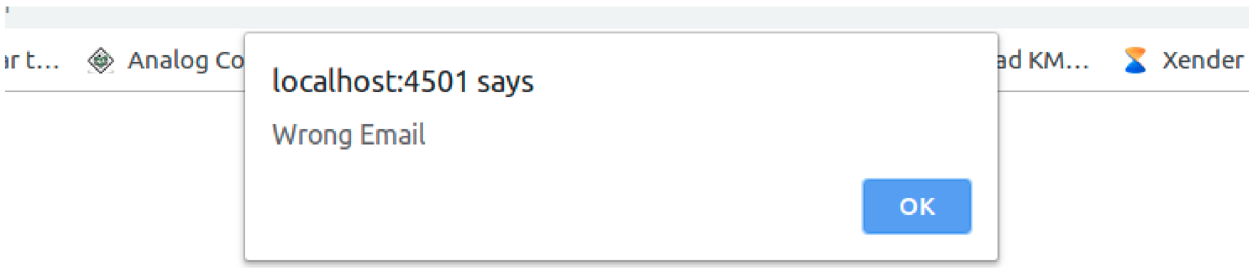
2. Login Page


NeoSCRUM
Login
Email *
Password *
Login
Email input should use input reg
Password must be alfanumeric
api/login

The Login page has the following features.

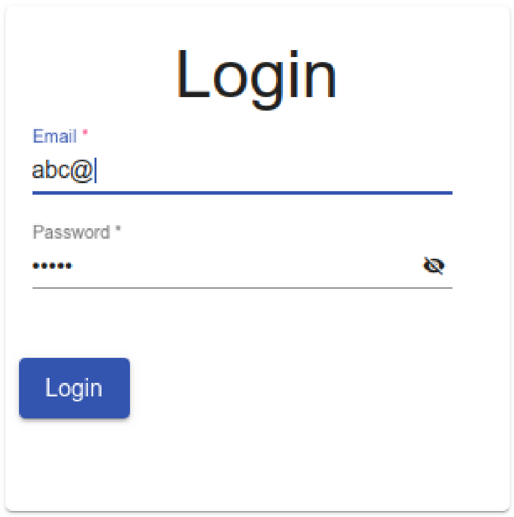
- Input fields

    a. There are two input fields email and password as shown in the figure. Validation is also done if the user enters invalid inputs. It will show error popup.



localhost:4501 says

Wrong Email



- API to hit
    a. POST api/login For user login(also generate jwt token)

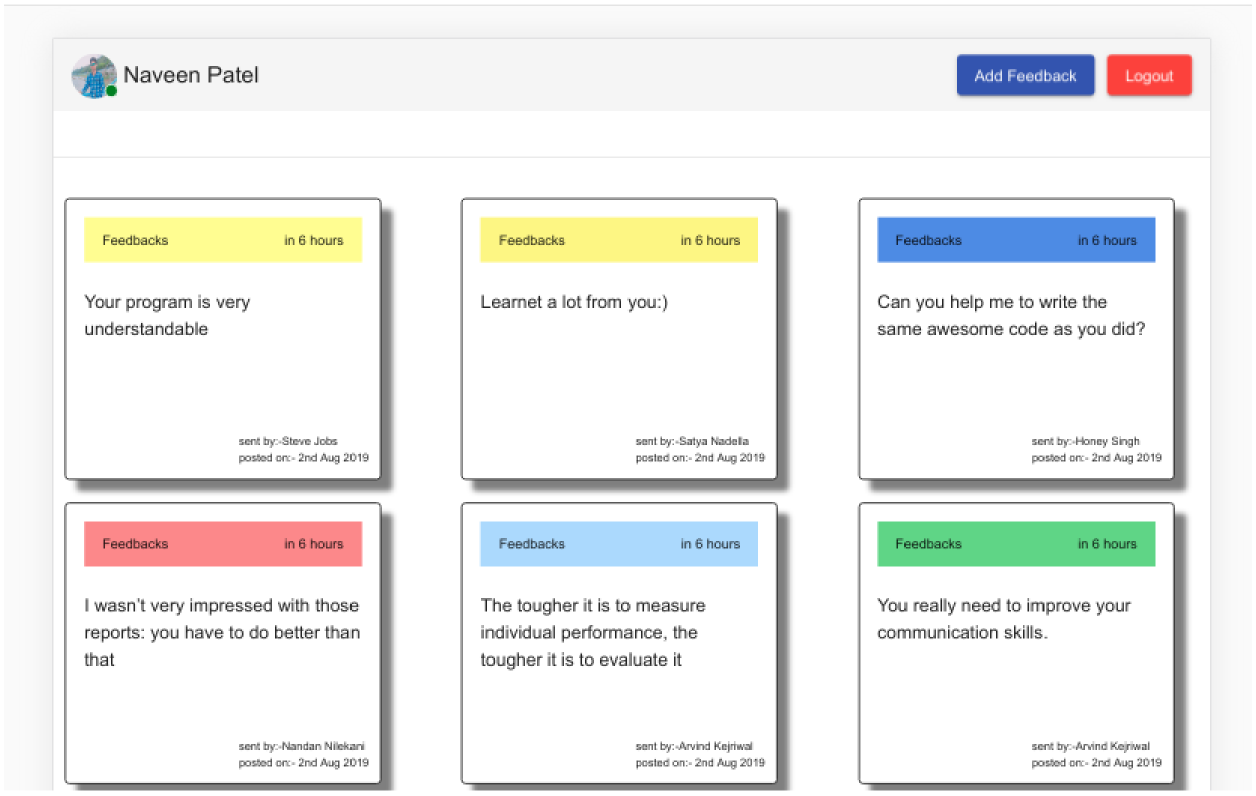
- Work-Flow

  As we have two input field user_email, and password for user login. Where the user can log in by inserting their credentials and when user will press login button, first we will validate all fields and if all fields filled correctly then we will hit login API and at the backend side we are also validating all field by using JOI (npm package) and If the data is correct then we will get a success response with JWT token from API which is generated at backend by using JWT(npm package).

  And we store the token in our browser's local storage so that we can use it in the next user-related API, but if the user credentials are not correct then we will get a failure(error) response form API and the error message will be related to validation error or verification error like if the user is not registered user or password not match then an error popup will be shown related to error.

3. Dashboard

The Dashboard page has the following features :



Feedbacks in 6 hours
Your program is very understandable
sent by:-Steve Jobs
posted on:- 2nd Aug 2019

Feedbacks in 6 hours
Learnet a lot from you:)
sent by:-Satya Nadella
posted on:- 2nd Aug 2019

Feedbacks in 6 hours
Can you help me to write the same awesome code as you did?
sent by:-Honey Singh
posted on:- 2nd Aug 2019

Feedbacks in 6 hours
I wasn't very impressed with those reports: you have to do better than that
sent by:-Nandan Nilekani
posted on:- 2nd Aug 2019

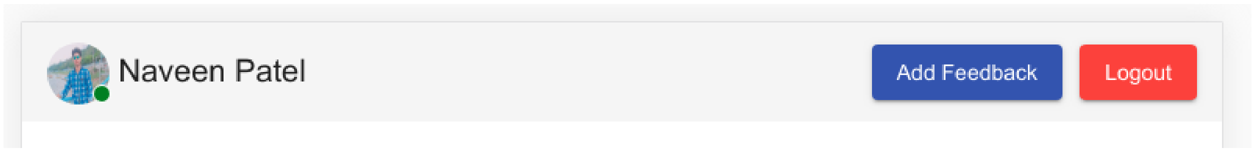
Feedbacks in 6 hours
The tougher it is to measure individual performance, the tougher it is to evaluate it
sent by:-Arvind Kejriwal
posted on:- 2nd Aug 2019

Feedbacks in 6 hours
You really need to improve your communication skills.
sent by:-Arvind Kejriwal
posted on:- 2nd Aug 2019

- Feedback list

  It contains a list of feedback which is given by other users. When feedback load it will animate the entire list. Angular Animation is also used on the loading of feedback.

- Navbar



The Navbar contains the following :

1. User information- User profile image and user name
2. Buttons (Add feedback) for adding feedback for other users and (logout) for account logout.

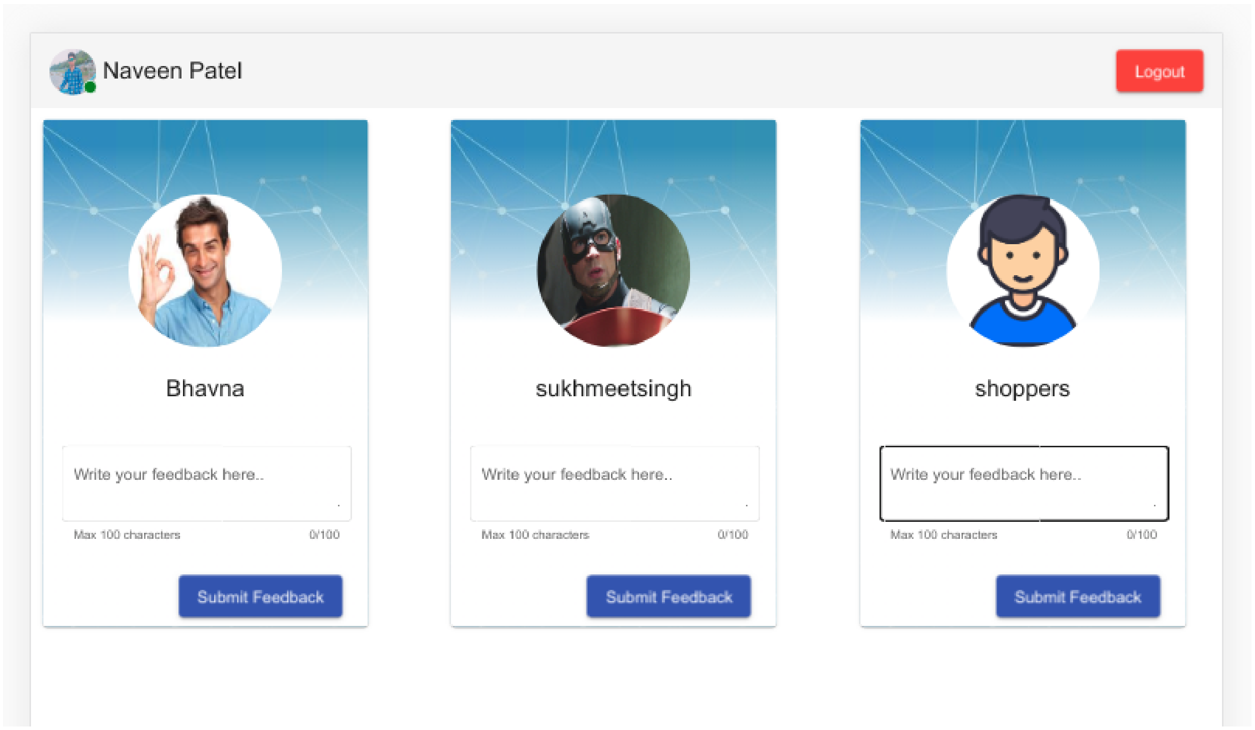
- API to hit
  - GET api/dashboard For getting dashboard data(on oninit/on load)

- Work-Flow

  As we stored JWT token in our browser’s local storage after successful login. Now we have to call dashboard API on on-init or on-load of dashboard page and where we need to send JWT token in the header because we are using JWT at the backend side for security purpose, so we verify the token at the backend side and if the token is valid then we will get a success response with dashboard data of user who is logged in.

  The user’s feedback is shown in the card and the user’s details are shown in the navbar. Navbar contains Name, profile image of the user and two buttons in which one is for logout by which the user can logout and another one is for redirecting to add feedback page where the user can give feedback to other users but here we also need to send a token with API.

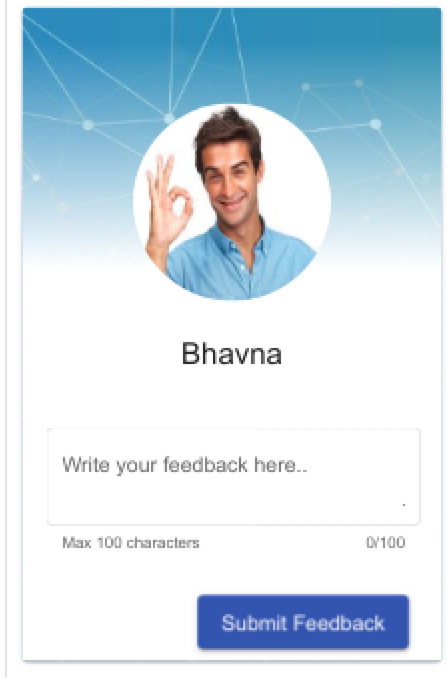
4. Add feedback page



The Add feedback page contains the following :

- Receiver list

  It contains a list of receivers(also input fields) in which logged in user can give feedback for the receivers and list automatically update every Friday.



- Card

  It contains information - the name of the receiver and profile image and input field for feedback.

- Submit Feedback

  Whenever the user hits the submit button it calls API and saves feedback in the database and then the submit button will be disabled. And card automatically removed from the list.

- API to hit
  - GET api/Feedback For getting a list of receiver
  - POST api/Feedback For sending feedback

- Work-Flow

On this page, first we verify the token at the backend side and if the token is valid then we will get a success response with data. The data contain a list of user data to whom the user has to give feedback. It contains the name and profile image of users to whom feedback is to be sent.

Here we use cron job at the backend side. The cron job is a npm package that schedules a command or script on your server to run automatically at a specified time and date. So using this npm package we are generating a list of sender and receiver in such a way that no sender is repeated and the sender can't be a receiver for itself and also the same receiver can't be repeated twice in any sender's list.

Once sender (user) has given feedback to any receiver (user) so its card will be removed from screen as well update in the database by API so that sender can't give feedback again to the same user.

The unique thing is that after specified time cron job runs the server and hit api and mail automatically sent to the user. It contains receiver's name and login page's link. It also a type of reminder that reminds the user that you have to give feedback for the mentioned receiver.

Template

The zip of the website template is available here

Packages used for Angular 7 Project

- Ngx-bootstrap

  ngx-bootstrap contains all core (and not only) Bootstrap components powered by Angular. So you don't need to include original JS components, but we are using markup and CSS provided by Bootstrap. Its official documentation is here

- Moment.js

  Moment.js is a free and open-source **javascript library** that removes the need to use the native javascript Date object directly

- Ngx-spinner

  An animated loading **spinner** for Angular 4+ is intended to inform the user that an operation is in progress.

- angular material

  Angular Material is a UI component library for **Angular** JS developers. **Angular Material** components help in constructing attractive, consistent, and functional web pages and web applications while adhering to modern web design principles like browser portability, device independence, and graceful degradation.