Project: - PPL

Requirement Specification:

- Register Page
    - Ask for email, first name, last name(OPTIONAL), password, confirm password
    - On submit user is registered and a welcome email is sent to user
    - Use nodemailer for sending emails
- Sign in Page
    - Ask for email and password
    - Forgot password link
- Forgot password page
    - Ask for email id to send a mail to reset password
- Reset password
- Home Page
    - Header
        - Left Area: Name of site
        - Middle Area : Home,ECoupons,E-Brand,Reuse Market Button (dummy links for now)
        - Right Area :
            - Language switcher (leave for now)
            - Login / First name with drop down menu-
                - My profile
                - Change password
                - Logout (stay where user was but in case of my profile/change password take user to home page)
    - Middle Portion

      Left Area
        - Friends checkbox - show post by all the friends of current user.(leave for now)
        - Flagged checkbox-show all posts with flagged status true
        - Latest First,Oldest First,Most Pet,Most Clicks,Most Commented
        - Post listing : All those posts which are created by different user recently
        - Upload Post(click to post, move to post details page)
            - Title
            - Post creator image,name

            - Time and date
            - Image
            - Share,flag/unflag,Like/unlike,comment(Comes left),Category name(comes right)

        Right Area (Sections)
        - Upload post popup,if user not login then show message first for login
        - Categories List
        - Featured List
    - <u>Footer</u>
        - Terms and conditions,social login buttons(leave for now)

- Detail Page

  <u>Left Area</u>
    - Post
    - All comments view(top to bottom,maximum 5 at one time)
    - View more(to show next comments)
    - Add comment:-open input box to add comment

  <u>Right Area</u>
    - Similar to home page

- Admin user:-
    - Delete button(come bottom right of post):-delete post
    - Delete / unflag posts / mark featured
    - Create, edit, delete categories, featured
    - Seeding data for categories

- Backend:-
    - DB design
    - Create Server with Koa and Nodejs
    - Create Mongoose Schema
    - Create Endpoints for
        - POST /register
        - GET /verifyuserbyemail
        - POST /login
        - POST /forgotpassword/:emailid // forget password(send mail to user and set token)
        - POST /resetpassword/:emailid - Load : token and new password //set new password

        - POST /post
        - GET/post/:postid
        - GET /posts   // flagged,category,sort_by also pagination
        - PUT/post/:postid //delayed until edit feature
        - PUT /post/like/:postid
        - PUT /post/unlike/:postid
        - POST /comment
        - GET /comments with limit skip
        - CRUD / categories
        - featured (PUT /posts).
        - GET/myprofile   //get user profile details
        - PUT/editprofile   //edit user profile
        - POST/changepassword/:userid
        - GET/logout/:userid

DAY WISE BREAKUP

Day 1

- DB design
- Create Server with Express and NodeJs
- Create mongoose schemas for

Day 2

- Use passportjs for session management
- Create endpoints for
    - POST /register   //User register
    - GET /verifyuser/email/verification_code   //Verify user
    - POST /login   //Login user
    - POST /forgotpassword/:emailid   //forget password
    - POST /resetpassword/:emailid   //set new password

Day 3

- Create endpoints for
    - POST /post   //create a post
- Frontend setup using Angularjs i.e. Setup Angular seed to route home page

- Login/Register page HTML
- Bind Register with backend endpoint to save in db and show appropriate message on UI
- Bind login with end point and redirect to homepage UI and Save user information in localStorage

Day 4

- Forget password template and use link on login page to open forget password page
- Forget password page should be bind to backend endpoint
  - generate token
  - create html mail template
  - send mail using nodemailer
- Reset password page template and bind it to its endpoint for resetting password

Day 5

- Create endpoints for
  - GET/logout
- Home page html and check on page load if user is logged in
- Home Page Header
  - Login (in case user is not logged in)
  - username /dropdown(change password,my profile,Logout)
- Bind logout to backend endpoint
- GET categories and set in scope so that it can be used for upload post form.

Day 6

- Create new post html template and bind to POST /post endpoint include the validations i.e. title and image are mandatory.
- Create endpoint for GET /posts // flagged,category,sort_by also pagination
- Use directive at frontend to show posts.
- Category list on right bar and link it to filter the posts

Day 7

- Bind get all posts(flagged,category,sort_by also pagination) with backend endpoint
- Create endpoints for
  - PUT /post/like/:postid
  - PUT /post/unlike/:postid
- Bind like/unlike to backend endpoint.
- Create endpoint for
  - GET/post/:postid
- Detail page HTML
- Redirected to detail page

Day 8

- Create endpoint
  - POST /comment
  - GET /comments with limit skip
- Bind add comment to backend endpoint
- Bind view more (comments) to backend endpoint

Day 9

- MyProfile html
- Create endpoints for
  - GET /myprofile   //get user profile details
  - PUT /editprofile   //edit user profile
- Bind to get profile details backend endpoint and redirected to My profile page

Day 10

- Changepassword html
- Create endpoints for
  - POST/changepassword/:userid   //change user password
- Link to change password page
- Delete post available checking admin
- Create endpoints
  - PUT /post/:postid for marking post deleted.

Day 11/12

- CRUD /categories

- featured (PUT /posts)
- Category crud UI for admin
- Thorough testing

DB Schema

User

- username-string   // first name +last name
- email-string
- password - string
- verification_code -string
- verified - boolean
- reset_pass_token -string,
- role - string   //user or admin

Post

- postedBy -fk
- title - string
- creatorImage - string
- creatorName - string
- createdOn - Date
- catType - string
- comments - array [] //field - creatorId, creatorName, comment, commentedOn
- commentcount - number
- likeby - [ ] users
- likecount - number
- flagcount - number
- flag by - [ ] users

Role

- name -string
- actions - [] permissions