
# Basic-ish AF blog webshite

06.08.2021

—

**Devleet d.o.o**
Obirska ulica 1,
1000 Ljubljana

## Summary

You are tasked with the creation of a web application from start to finish. The project described below is split into several smaller tasks. It is advisable to firstly get a good idea of the project's architectural setup, before proceeding with the implementation. You are encouraged to complete all of the tasks, but it is not strictly required. The tasks in bold are harder and thus much more valuable than others, but cannot be implemented independently.

Good luck.

## Project description

### Architecture

The project must consist of a client-side application and a backend service. The client side application must be built in Angular version 12, all libraries and scripts used in the implementation should be installed or packaged along with the application (avoid CDN links).

The backend must be built using Amazon's Amplify service. Manual "tweaking" of services deployed via the amplify cli is allowed if no suitable solution can be found through the use of amplify cli. The backend will need to have a functional REST API, several lambda functions to facilitate the business logic and a DynamoDB table, which must follow a single table design pattern. For authorization and authentication, the Amazon Cognito service must be used, communication between client and REST api should be via a JWT, which is received from cognito after a successful login.
At the end of the development, the application should be published through the amplify console.

Bonus points for using a git solution with reasonable branch management.

### Application

The website allows the user to browse different blog posts created by registered users. The blog post itself consists of text and an optional image.
An unregistered user can only browse over blog posts and read comments, but not perform any other action aside from starting the process of registration or login.
After completing the registration, the user can then create comments on blog posts. The

user can then update or delete their own comments. Aside from creating comments on blog posts, the user should have the ability to comment upon another person's comment (creating a comment chain). Registered users have the ability to upvote or downvote a comment. A user can only leave 1 vote per comment, but can be changed between like and dislike. Registered users can also remove a previously given vote, for example by clicking the upvote button twice.
Registered users can create, update and delete blog posts. These actions should only be allowed to the authors of the blog post in question. When a blog post is deleted, all of the comments and comment chains associated with the blog should be deleted as well.
Registered users must have the ability to logout.

## Tasks

- Development setup
    - The goal of this task is to setup a blank Angular application and initialize a new Amplify project. This step is necessary, since all additional tasks will be built upon both technologies.
        - ☐ Setup
- Create a working authentication service for the application
    - The goal of this task is to implement the functionality, which allows users to register, login and logout. Optionally, the user's email should be verified.
        - ☐ Client side Cognito integration (registration, login, logout)
        - ☐ Amplify Cognito setup (AWS)
        - ☐ (Optional) Verify user's email
- Blog post creation
    - The goal of this task is to implement the functionalities of the blog post, with regard to the authors. After this task is complete, the registered user will have the ability to create a new blog post, edit an existing blog post or delete an existing blog post.
        - ☐ Create new blog post - Client side
        - ☐ Update blog post - Client side
        - ☐ Delete blog post - Client side
        - ☐ Create new blog post - Amplify implementation

        - [ ] Update blog post - Amplify implementation
        - [ ] Delete blog post - Amplify implementation
        - [ ] **Restrict update and delete view to authors only - Client side**
        - [ ] **Restrict update and delete to authors only - Server side**
        - [ ] **(Optional) Implement adding and displaying an image in a blog post.**
        - [ ] **(Optional) Set image to be publicly accessible to unauthorized users.**
- Blog post listing
    - The goal of this task is to create a sort of gallery or a list, where all or a portion of random blog posts are displayed to the users. (Facebook / Twitter - like for example). Remember that user experience is not important, only functionality.
        - [ ] Display blog posts on Client side
        - [ ] **Rest API call to retrieve all / some of the blog posts for display**
- Comments
    - The goal of this task is to expand the displayed blog posts with the ability for the registered users to comment upon them. The comments can also be created upon other comments, creating a comment chain. If a comment is in the middle or start of a comment chain, it should not be possible to remove it.
      The authors of the blog post and comment should have the ability to delete the comment. Only the author of the comment should have the ability to update a comment.
      Registered users should have the ability to vote on a comment (upvote / downvote). The sum of the votes should then be displayed somewhere in the comment.
        - [ ] Comment creation on blog post
        - [ ] Comment creation on comment ( comment chain)
        - [ ] Update / delete comment
        - [ ] **(Optional) Restrict deletion to only tail comments (end of comment chain)**
        - [ ] Restrict update only to author
        - [ ] **Restrict delete to author of comment and blog**

        - [ ] **Comment upvote system**
            - [ ] Upvote / downvote
            - [ ] **Restrict users to 1 vote per comment**
        - [ ] **Retrieve and display the sum of the votes.**
        - [ ] **(optional) Display comments sorted by upvotes.** (votes of the first comment in a chain or a solitary comment)
- Hosting
    - If all went well, the application can be easily hosted via Amplify by adding a hosting. Give it a try.
        - [ ] It's hosted!

## Documentation and other material

- Amplify
    - https://docs.amplify.aws/cli
        - Auth, Storage, Api, Function, Hosting
- Cognito
    - https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html
- AWS in general
    - https://www.youtube.com/c/BeABetterDev/playlists
- DynamoDB single table pattern
    - https://www.youtube.com/watch?v=8Ww1YW3AChE
- Angular
    - Udemy course, info in email

If there are any other Udemy courses you would like to watch, let us know. If we find it expediently we will add it to our collection at once, so you can watch them.