Web Operations

Backend

Problem Statement

INSTRUCTIONS:

1. Solve as many questions as possible. Each question tests your skill in one aspect. Solve as many sub-questions as Possible. Do not hesitate to

2. Answer each question separately and name the files as specified. All files must be compressed into one .**zip** file named "RollNumber_Name_SubDepartment.zip"and uploaded in the portal.

3. Deadline for submission of applications is **23:59, Friday, 23rd May**. Interviews will be held for the shortlisted candidates.

4. Try to use a **Linux environment. Ubuntu** is recommended.

5. If you have any doubts or queries, contact us at:
   Deepak Padamata – 9789107938 , deepak@saarang.org

6. **Resources:**
   * W3Schools
   * CodeAcademy
   * CodeSchool
   * If all else fails, try googling.
     Someone has probabaly faced your problem already.
7. Good Luck!

Part 1 - Answer the following questions:

1. What do you think are the drawbacks of the Saarang 2015 website and [app](app)?

2. How do you think you can improve on the present website and [app](app)? You can contribute an improvement to an existing feature or suggest a new feature.

3. Your credentials (any past work you have done as part of a team or any work related to Web/Mobile Operations).

Part 2 - Solve the following problems

For the next questions, you need to know the basics of JavaScript which can be learnt from the links provided.

1. Install nvm from here for Ubuntu and OSX. For Windows, install from here. After this, install node using nvm. You can follow this tutorial. If you face difficulties in installing, don't refrain from contacting me. After that, complete the "learnyounode" tutorial from nodeschool.io. Put a screenshot named "learnyounode.jpg" in the zip file showing your completion. The subsequent problems will be easier to solve if you do this first.

2. Todo App:
    Try solving this question with NodeJS (JavaScript) web framework. ExpressJS is also recommended, but not necessary.

- Your task is to build a todo app using a RESTful Node API. A brief description of its functionalities is given below:
- Create a todo – Provision to create a todo with the following fields:
  - Name – A name for your task
  - Description – A description for your task
  - Status – A boolean with 'true' if the task is completed or 'false' otherwise
  - Updated at – The time when it was last updated. This will be the time of creation if it was just created.
- List all tasks: You should be able to see all todos and their details with this.
- Update: You should be able to edit any field in the task and update. The "Updated At" field should automatically change on every update.
- Delete: You should be able to delete any task at will.

3. Blog App:

      Try solving this question with NodeJS (JavaScript) web framework. ExpressJS is also recommended, but not necessary.

Your task is to build a blog app using a RESTful Node API.

A brief description of its functionalities is given below:

- Registration: A visitor must be able to register. User details must be Name, Email ID and password.
  - Note: A lot of tutorials and templates are available for a 'user' implementation in NodeJS with Express. Make use of those.

The following functionality must be available only if the person is logged in:

- Create a post – Provision to create a post with the following fields:
  - Name – The name of person who posted
  - Content – The content of the post
- List all posts: You should be able to see all posts and their details with this.
- Edit and Delete: You should be able to edit or delete any post at will.
- Create a comment on a post:
  - Comment should also have the name of commenter and content.
  - Should also provide edit and delete functionality.

Note: The solutions to problems 2 and 3 must be in separate folders named 'Todo ' and 'Blog' respectively.

      Bonus points for making a web frontend for question 2 and 3.

      If you do make a frontend, make sure it is presentable – an easy way is to use bootstrap.