Mahesh IT Services

LAB ASSIGNMENT PROBLEM STATEMENTS FOR MEAN AND MERN

Mahesh Sabnis

MAHESH IT SERVICES

B-604, Prestene Fontana, LMD Chowk, Behind Maratha Mandir, Bavdhan Khurd Pune 411021

Assignment 1

Create a MEAN application which will contains following features

1. The PersonInformation database in MongoDB that will be containing collection of Person Information as like aadhar information (except any binary data)
    a. The Collection in MongoDB should capture information line
        i. PersonalUniqueueID
        ii. Full Name (3 Separate field First Name, Middle Name and Last Name).
        iii. Gender
        iv. Date-Of-Birth
        v. Age
        vi. Address
            1. Flat/Bungalow Number
            2. Society Name
            3. Street Name/Area Name
        vii. City
        viii. State
        ix. Pin Code
        x. Phone No
        xi. Mobile No
        xii. Physical Disability If Any
        xiii. Marital Status
            1. Married, Unmarried, Divorced, Widow, Widower, etc
        xiv. Education Status, e.g. Masters, Phd, Graduate, Under-Graduate, HSC, SSC, Illiterate, etc.
        xv. Birth Sign if Any
    b. The Mongodb Must have following collections
        i. Roles
            1. RoleID
            2. Role Name
                a. Roles will be
                    i. Administrator
                    ii. Operator
                    iii. AccessUser
        ii. Users
            1. User ID
            2. User Name
            3. Email Address
            4. Password
            5. Role Id
        iii. LoginStatus
            1. LoginStatusId
            2. UserName
            3. LoginForm

4. DateTime
5. IPAddress

2. Create a REST APIs using Express and Node to Access MongoDB data. These REST APIs will be performing following operations
    a. Create REST API for User Management as explain below
        i. Create Roles
        ii. Create Users
            1. User Must be assigned to Role while creating User
            2. If User assignment to role fails, then user creation should also be failed
                a. The default role is AccessUser
    b. The REST APIs, must manage login process and application access must be controlled using Token (Json Web Tokens)

    c. CRUD operations on Database for Personal Information
        i. The REST API should Connect to MongoDB Collection to Perform CRUD Operations.
            1. The Information of the Persons can be created by only by Administrator or Operator.
            2. The PersonUniqueueId can be generator, only when the Administrator Approves information.
            3. Similarly, the Information of the person can be Updated by AccessUser/Operator/Administrator. These modifications must be saved only when the Administrator approves it.
        ii. The REST APIs, must validate the posted data from the user while performing POST and PUT operations. If any invalid data is posted then the REST API must respond exact error to client.
    d. Search operations based on criteria. Note there should be single method in REST api which will perform search operation any criteria e.g. PersonName=value, City=Value, Age>value, Age<value etc.
        i. The search is accessible to only Operator and Administrator.
        ii. The AccessUser can see only his/her information after log-in.
3. The REST APIs must use Token Based Authentication using JWT
4. Create a Angular Front-End application which will perform following
    a. Perform User Authentication
        i. The Role Creation Form. This must be accessed by Administrator
        ii. User Registration Form
            1. This will be created only for registering Operators
        iii. The AccessUsers can Login only when their profile is created by Operator and Approved by Administrator
    b. The Login View
        i. This should create a token
    c. Provide Search View to search record making calls to REST APIs
        i. The Search View should provide search criteria (use radio button/dropdown)
            1. By City

2. By Pin
3. By State
4. By Kast Name
5. By Gender
6. By Age Condition

ii. The Search Result must be updated for each criteria's **on change** event

d. View for CRUD Operations by making calls to REST APIs
e. Perform Validations as per the needs (Developer must define scale of validations)
    i. Create and Update view must have validations displayed near the UI element.
f. A common Error-Page must be created for all types of errors.
g. LogOff View
    i. This should clear the token

Assignment 2

Create a MERN Application for performing following

This application is designed for the providing an information for various insurance companies and their insurance services. The site like http://www.Policybazaar.com

1. The MongoDB Database must be created for storing Insurance information for to be provided to customers
    a. Create Collection call Insurance Provider
        i. Provider Name
        ii. Address
            1. Office Location
            2. Building Name
            3. Area
            4. Pin Code
        iii. WebSite of the provider
        iv. Contact Information
            1. Phone number
            2. Fax number
        v. Working Days
        vi. Number of Branches
        vii. Branch Locations
            1. Address of Each branch by City
    b. Collection for Insurance information
        i. Type of Insurance
            1. Home
            2. Personal
            3. Accidental
            4. Children
            5. Family
            6. Vehicle
        ii. Insurance details

1. Insurance Type
2. Min Time Period → In Months
3. Max Time Period → In Months
4. Premium Amount

c. Create a collection for Insurance Provider Login Information
   i. User Name
   ii. Password
   iii. Contact Email
   iv. Contact Number (Must not mobile number)
   v. Insurance Company Name
   vi. Head office Address
   vii. Web-Site

d. Create a collection for Customer Information
   i. CustomerId (Must be Generated)
   ii. Customer Name
   iii. Selected User Name
   iv. Password
   v. Email Address
   vi. Contact Number
   vii. Mobile Number
   viii. Date-of-Birth
   ix. Address
   x. City

2. One Customer can select multiple insurance types. The Database must store information line Insurance Types e.g. Personal, Home, Vehicle, Accidental. (Please refer insurance provider site for further information about)
3. Create a REST APIs using Express and Node to Access MongoDB data. These REST APIs will be performing following operations
   a. Insurance Information Creator
      i. This API will need to Create Insurance Providers Login Information
      ii. The API will allow Insurance Provider to upload/edit Insurance Information
   b. Customer Information Creator
      i. This API will allow customer to create his profile to access insurance information. This will create Customer Login.
   c. API to receive insurance information based on Insurance Type
      i. Here Customer can select insurance type or Insurance Provider
         1. The Insurance Details either based on selected insurance type will be displayed or based on selected Insurance Provider.
   d. API for managing user authentication based on Passport.js
   e. API for applying for insurance
4. Create React Front-End for accessing REST APIs for following
   a. Authentication
      i. View for Insurance Provider Registration
      ii. View for Customer Registration

    iii. View for Logic as Insurance Provider or Customer
b. View for CRUD operations using REST APIs
    i. CRUD Operation views for Insurance Provider
c. Insurance Information View for Customer must be created to show Information about selected insurance type.
d. Validations (Developers can decide scope for validations)
    i. Create and Update View must have validation
e. Error Handling
    i. A Common View for all Error Information
f. LogOff
    i. This will create Authentication information