**Node.js Tasks**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Description of the Task** |
|  | Basic CRUD | Create a basic CRUD application :  - Insert atleast 6-8 fields into the DB tables  - Get all the records from the table  - Get a particular based on criteria (I.e Id or name fields)  - Update particular Record  - Delete (Hard Delete) from the table |
|  | Joi Validations | Add a request Joi Validations to the above Task-1(CRUD):  - validate all the request validations such that it can be anything ( query parameters , path parameters , body parameters )  - also validate for different types like : Phone numbers , Emails , can allow empty , regular expressions |
|  | Loggers | - Make use of winston npm module in the above task such that use levels like (info,debug,error) and use these loggers instead of console messages  - for simple debugging and testing purposes use info,debug levels at concole  - for erorr message make sure to store those messages in separate file like (logs/errors.log) |
|  | Erorr/Exception Handling | Make sure that we are handling erorrs and exceptions at all places  - DB connections, DB column names error , DB Authentication  - Joi Validations  - 400,404,500,,,etc |
|  | User Management CRUD | Perform CRUD operation (Registration , Login , Update , Remove ):  - **Registration API**: Create an API to store request data in db table such that except all fields the password should be auto generated and that autogenerated passsword must be Encrypted/Hashed while storing in database and the auto generated password must be sent to particular mail id.  - **Login API**: Create an API to check for the stored encrypted password with the plain text password while login  - Get a list of all users (make sure passwords should not come in response object)  - Update user details based on some ID or email (unique field)  - Delete a user information from table (Soft Delete instead of Hard Delete ) also while getting list of users , it should not get soft deleted record in which the record is deleted |
|  | JWT | **JSON Web Tokens :**  - Generate a JWT token while login was successful and that token should have some expiration time. |
|  | Secure API's | - Provide Security to all API's with login JWT Token  by passing JWT token in Header of the request payload and validate the Token while hitting to specific API. |
|  | Email Service | - usage of nodemailer npm module.  - For Registration API make sure that autogenerated password to be sent to particular email ID |
|  | Joins | Association among tables (joins)  - Create 3 tables in database to have a relationship among them  - Using Sequelize methods for Joins(Assiciations) , Create an API to Get the data from all the 3 tables with  inner join / left join  - Build a response in simple JSON's rather than nested JSON format joins (default Join result) |
|  | Filters | Filter the response Data  - Insert more number of records in of the tables (users). Atleast 30+ records  - Create an API to get the list of users with filters like :  OrderBy , Sorting , Nested Sorting , Regular Expression Search , pagination ( limit, offset )(page size , page number). |
|  | File Systems | Work on File systems :  - Perform CRUD operations on File rather than DB. |
|  | Testing | Unit Testing :  - Add a test cases using MOCHA , CHAI  - Test all the API's (GET,PUT,POST,DELETE).  - Perform the Test cases on Service layer (on top of functions)  Mock Testing :  - While testing for Adding,Update,Delete Record the test case should not affect database tables.(it should not modify or add to the db ) |
|  | Code Coverage | Check the Code Coverage using istanbul npm module  - try to test cases for the code as much as possible until all the statements, lines are covered in code coverage reoprt  - the code coverage report should be GREEN color in all the files in the project  Note : Code coverage report should not indicate RED color such that the application need to have tested more (So write more test cases ) |
|  | TDD Testing | Test Driven Development  - Make sure that the test cases are written first and then write the code to make the test case pass. |
|  | Swagger | API representation (Documentation):  - Add all the API's in the Swagger such that it should represent along with JWT authentication for the api's in the Swagger  - Integarte the Swagger inside the Developed Application to run on some Endpoint |
|  | Git,PCF | Push the Application to Git Hub  - maintain 2 branches (master , develop)  Deploy Application to PCF and run the application in it. |