

Backend Developers Training Plan

1. File sharing app like google drive.

Tech Stack:

- Backend: Node.js with Express.js
- Database: PostgreSQL
- Authentication: JWT (JSON Web Tokens)
- Multer - File handling
- Nodemailer - Use for email sending to verify email, forgot password reset and access sharing invitation emails.

What Candidates will Learn:

- Setting up a Node.js and Express.js project.
- Configuring and connecting to a PostgreSQL database.
- Implementing JWT-based authentication for user registration and login, forgot password and reset password.
- Handling file uploads using Multer and storing files on server storage.
- Implementing email functionality using nodemailer to share files via email.
- Creating public links for file sharing and securing public access and limits as per user plans.
- Indexing and searching files using database queries and filters.

Features of the File Sharing App:

Frontend Tasks:

1. User Authentication:

- Implement user registration and login forms.
- Create authentication routes for user registration, login, and logout.
- Set up JWT token management for user sessions.

2. Dashboard and User Interface:

- Design a user dashboard to display user-specific information.
- Create a user-friendly interface for file management and sharing.
- Display user profile details and account settings.

3. File Upload Component:

- Design a file upload interface with drag-and-drop functionality.
- Implement client-side validation for allowed file types and sizes.
- Display file upload progress and success messages.

4. File Listing and Management:

- Create a component to list user's uploaded files.
- Implement features to view, download, and delete files.
- Include sorting and filtering options for file management.

5. File Sharing and Permissions:

- Design a sharing component to manage file sharing options.
- Allow users to set permissions for shared files (public, private, specific users).
- Display shared files and permissions in the user dashboard.

6. Search:

Implement a search bar to search for files by name or keyword. Display search results in real-time as the user types.

8. Styling and Responsiveness:

- Apply responsive design principles to ensure the app works well on various devices.

- Create a visually appealing and consistent design using CSS.

Backend Tasks:

1. User Authentication and Registration:

- Implement user registration and login endpoints.
- Set up password hashing and user authentication using JWT.

2. File Upload and Storage:

- Create API endpoints to handle file uploads.
- Implement file storage and management on the server.

3. File Metadata and Database Integration:

- Design database schemas to store user profiles, files, and sharing permissions.
- Implement database queries to manage file metadata and permissions.

4. File Sharing and Permissions Logic:

- Implement business logic to handle file sharing and permissions.
- Validate user permissions when accessing shared files.

5. Search and Indexing Logic:

- Develop search functionality to query files based on user input.
- Integrate search with the database and return relevant results.

6. Real-Time Notifications:

- Implement real-time notifications using WebSockets (Socket.io).
- Set up notification triggers for file sharing and updates.

7. API Documentation:

- Document the API endpoints using tools like Swagger or Postman.
- Provide clear instructions on how the frontend can interact with the backend.

8. Security and Error Handling:

- Implement security measures to prevent unauthorized access and data leaks.
- Set up error handling for different scenarios and provide meaningful error messages.

Basic Ecommerce Project

Frontend Tasks:

User Authentication:

- Implement a simple user registration form.
- Set up a login form to allow users to log in.
- Use mock data or hard-coded credentials for demonstration.

Product Listing:

- Display a list of mock products on the homepage.
- Include product names, images, and brief descriptions.

Product Details Page:

- Create a basic product details page for demonstration purposes.
- Display product information such as name, image, and price.

Shopping Cart:

- Implement a minimal shopping cart interface.
- Allow users to add products to the cart and display a basic cart summary.

Checkout Process:

- Design a simplified checkout process with basic steps.
- Include placeholder forms for shipping and payment information.

Order History:

- Create a basic order history page to showcase user order data.

Backend Tasks:

User Authentication and Registration:

- Set up basic API endpoints for user registration and login.
- Generate mock JWT tokens for demonstration.

Product Management:

- Implement API endpoints to fetch mock product data.
- Create a basic product schema in the database (for future enhancements).

Product Search and filters and pagination in data:

- Implement API endpoints to fetch mock product data.
- Create a basic product schema in the database (for future enhancements).

Shopping Cart Logic:

- Create basic API endpoints to simulate cart functionality.
- Store cart data in-memory or use mock data.

Checkout Process and Payment Simulation:

- Set up API endpoints to simulate order creation and payment.
- Generate mock order data for demonstration.

Order Management and Status:

- Implement API endpoints to showcase order history.
- Include mock order data with status updates.

User Reviews and Ratings:

- Implement basic endpoints to showcase user reviews and ratings (without actual input).
- Store and display mock review data.

Mock Interviews

- Candidates will use chat GPT to mock interviews daily on node js, javascript, typescript, Postgresql and Type ORM fundamentals.

- We will conduct a weekly mock interview of candidates with a backend developer. He will make sure the candidate really knows what he has made on project and his fundamental knowledge.

Daily Algorithm solving

1. **Reverse Words in a String:**

Reverse the order of words in a given string.

2. **Valid Parentheses:**

Check if a given string of parentheses is valid.

3. **Array Intersection:**

Find the intersection of two arrays.

4. **Two Sum:**

Find two numbers in an array that add up to a target sum.

5. **Remove Duplicates from Sorted Array:**

Remove duplicates from a sorted array in-place.

6. **Group Anagrams:**

Group anagrams together from an array of strings.

7. **Minimum Window Substring:**

Find the minimum window in a string that contains all characters of another string.

8. **Longest Substring Without Repeating Characters:**

Find the length of the longest substring without repeating characters.

9. **Merge Intervals:**

Merge overlapping intervals in an array.

10. ****Container With Most Water:****

Determine the maximum area that can be formed by connecting two vertical lines with the x-axis

11. ****Next Permutation:****

Find the next lexicographically greater permutation of numbers.

12. ****Subsets:****

Generate all possible subsets of a set.

13. ****Combination Sum:****

Find all unique combinations of numbers that sum up to a given target.

14. ****Word Break:****

Determine if a given string can be segmented into space-separated words in a dictionary.

15. ****Jump Game:****

Determine if you can reach the last index of an array by jumping from element to element.

16. ****Valid Sudoku:****

Check if a given Sudoku board is valid.

17. ****Longest Increasing Subsequence:****

Find the length of the longest increasing subsequence in an array.

18. ****Search in Rotated Sorted Array:****

Search for a target value in a rotated sorted array.

19. ****Three Sum:****

Find all unique triplets in an array that sum up to a target.

20. ****Permutations II:****

Find all unique permutations of an array with duplicates.

21. ****Minimum Path Sum:****

Find the minimum path sum from the top-left to the bottom-right corner of a grid.

22. ****Kth Largest Element in an Array:****

Find the kth largest element in an unsorted array.

23. ****Sort Colors:****

Sort an array containing 0s, 1s, and 2s (Dutch National Flag problem).

24. ****Longest Common Subsequence:****

Find the length of the longest common subsequence between two strings.

25. ****Find Peak Element:****

Find a peak element in an array, which is greater than or equal to its neighbors.

26. ****Evaluate Reverse Polish Notation:****

Evaluate an expression given in Reverse Polish Notation.

27. ****Trapping Rain Water:****

Calculate how much rainwater can be trapped between bars in an elevation map.

28. ****Word Search:****

Determine if a word exists in a 2D board of characters.

29. ****Coin Change:****

Find the minimum number of coins needed to make up a target amount.

30. ****Decode Ways:****

Count the number of ways to decode a string containing digits.

