MICRO BLOGGING APPLICATION USING MEAN

Write AIM

ALGORITHM

1) Create a Project Structure

/microblogging-app

/backend

/models /routes server.js /frontend

/src

/app

/components /services angular.json

2)

Set Up Backend

Initialize the Node project mkdir microblogging-app cd microblogging-app mkdir backend cd backend npm init -y npm install express mongoose cors body-parser

3)

Create the Server (server.js): const express = require('express'); const mongoose = require('mongoose'); const cors = require('cors');

const bodyParser = require('body-parser');

const app = express(); const PORT = 3000;

app.use(cors()); app.use(bodyParser.json());

// Connect to MongoDB mongoose.connect('mongodb://localhost:27017/microblog', { useNewUrlParser: true, useUnifiedTopology: true,

}).then(() => console.log('Connected to MongoDB'));

// Define Routes const postsRoutes = require('./routes/posts'); app.use('/api/posts', postsRoutes);

app.listen(PORT, () => { console.log(`Server is running on http://localhost:${PORT}`);

});

4)

Create Mongoose Model (/models/Post.js):

const mongoose = require('mongoose');

const PostSchema = new mongoose.Schema({

content: { type: String, required: true, },

date: { type: Date, default: Date.now,

},

});

module.exports = mongoose.model('Post', PostSchema);

5)

Create Routes for CRUD Operations (/routes/posts.js):

const express = require('express'); const Post = require('../models/Post'); const router = express.Router();

// Get all posts router.get('/', async (req, res) => { const posts = await Post.find(); res.json(posts);

});

// Create a new post router.post('/', async (req, res) => { const newPost = new Post(req.body); await newPost.save(); res.json(newPost);

});

// Edit a post

router.put('/:id', async (req, res) => { const updatedPost = await Post.findByIdAndUpdate(req.params.id, req.body, { new: true

});

res.json(updatedPost);

});

// Delete a post router.delete('/:id', async (req, res) => { await Post.findByIdAndDelete(req.params.id); res.json({ message: 'Post deleted' });

});

module.exports = router;

6)

Frontend (Angular) Set Up Angular Project: cd ..

ng new frontend cd frontend ng serve

7)

Service to Interact with Backend (/src/app/services/post.service.ts):

import { Injectable } from '@angular/core'; import { HttpClient } from '@angular/common/http'; import { Observable } from 'rxjs';

const API\_URL = 'http://localhost:3000/api/posts';

@Injectable({ providedIn: 'root'

})

export class PostService { constructor(private http: HttpClient) {}

getPosts(): Observable<any> { return this.http.get(API\_URL);

}

createPost(content: string): Observable<any> { return this.http.post(API\_URL, { content });

}

updatePost(id: string, content: string): Observable<any> { return this.http.put(`${API\_URL}/${id}`, { content });

}

deletePost(id: string): Observable<any> { return this.http.delete(`${API\_URL}/${id}`);

}

}

8)

Component for Displaying and Managing Posts (/src/app/components/posts/posts.component.ts):

import { Component, OnInit } from '@angular/core'; import { PostService } from '../../services/post.service';

@Component({

selector: 'app-posts', templateUrl: './posts.component.html',

})

export class PostsComponent implements OnInit {

posts: any = []; newPost: string = '';

constructor(private postService: PostService) {}

ngOnInit(): void { this.loadPosts();

}

loadPosts() { this.postService.getPosts().subscribe((data) => { this.posts = data;

});

}

addPost() { this.postService.createPost(this.newPost).subscribe(() => { this.newPost = ''; this.loadPosts();

});

}

deletePost(id: string) { this.postService.deletePost(id).subscribe(() => { this.loadPosts();

});

}

}

9)

HTML for the Component (/src/app/components/posts/posts.component.html):

<div class="container">

<h2>Microblogs</h2>

<input [(ngModel)]="newPost" placeholder="What's on your mind?" />

<button (click)="addPost()">Post</button>

<div \*ngFor="let post of posts" class="post">

<p>{{ post.content }}</p>

<button (click)="deletePost(post.\_id)">Delete</button>

</div>

</div>

10)

Import HttpClient and FormsModule in app.module.ts:

import { HttpClientModule } from '@angular/common/http'; import { FormsModule } from '@angular/forms'; import { PostsComponent } from './components/posts/posts.component';

@NgModule({ declarations: [PostsComponent], imports: [HttpClientModule, FormsModule], providers: [],

bootstrap: [AppComponent],

})

export class AppModule {}

11)

Run the Application

node server.js

Frontend: In the frontend directory, run

ng serve