Angular CRUD Application with PrimeNG

This Angular application is built to perform CRUD (Create, Read, Update, Delete) operations using PrimeNG components. It fetches data from a sample API and displays it in a sortable, filterable, and paginated PrimeNG table. Users can view, update, and add new records through the user interface.

Installation

Before running the application, make sure you have Node.js and npm installed on your system. Then, follow these steps:

1. Clone the repository to your local machine:

   ```
   git clone https://github.com/your-username/angular-crud-app.git
   ```

2. Navigate to the project directory:

   ```
   cd angular-crud-app
   ```

3. Install dependencies using npm:

   ```
   npm install
   ```

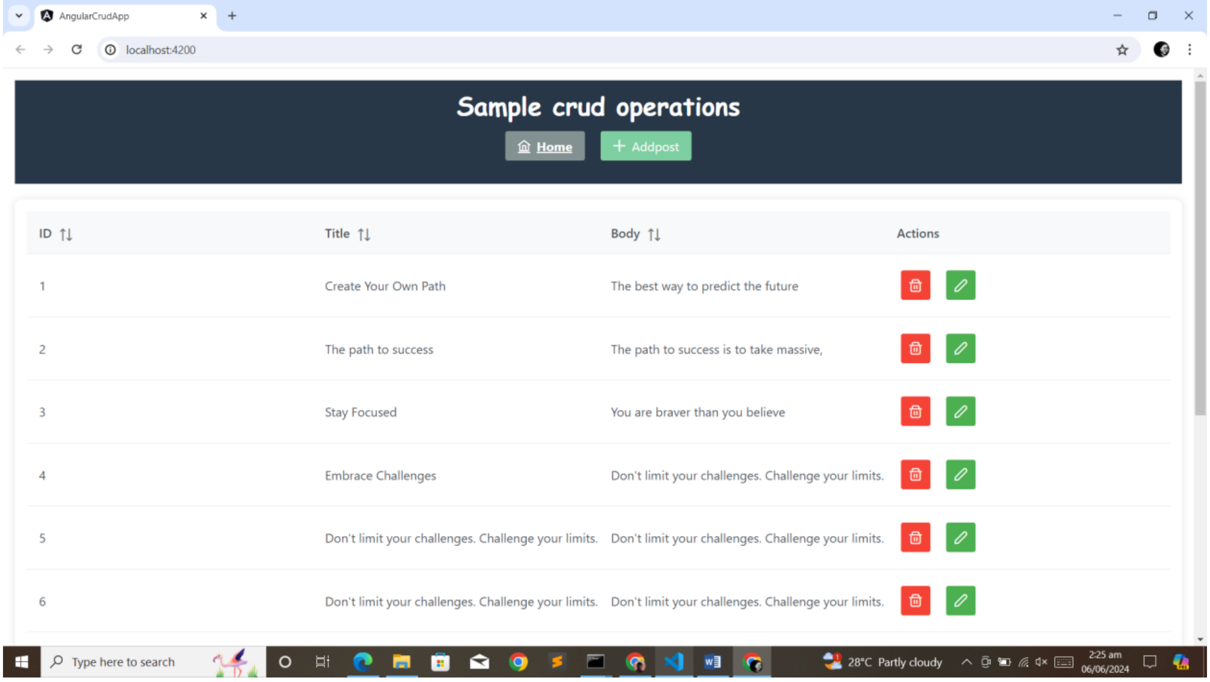
Running the Application

Once the dependencies are installed, you can run the application using Angular CLI:

```
ng serve
```

This command will start the development server. Once the server is running, you can access the application in your web browser at http://localhost:4200

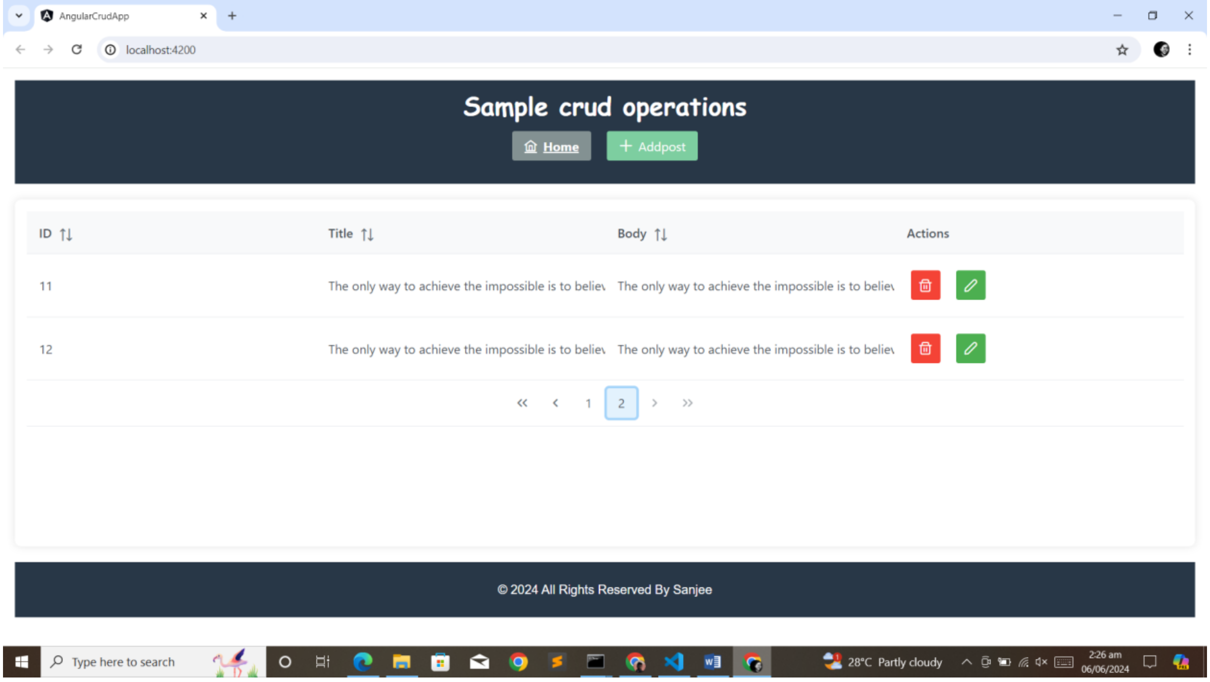
Some Screenshots



Sample crud operations

ID ⇅	Title ⇅	Body ⇅	Actions
1	Create Your Own Path	The best way to predict the future	🗑 ✎
2	The path to success	The path to success is to take massive,	🗑 ✎
3	Stay Focused	You are braver than you believe	🗑 ✎
4	Embrace Challenges	Don't limit your challenges. Challenge your limits.	🗑 ✎
5	Don't limit your challenges. Challenge your limits.	Don't limit your challenges. Challenge your limits.	🗑 ✎
6	Don't limit your challenges. Challenge your limits.	Don't limit your challenges. Challenge your limits.	🗑 ✎

Pagination

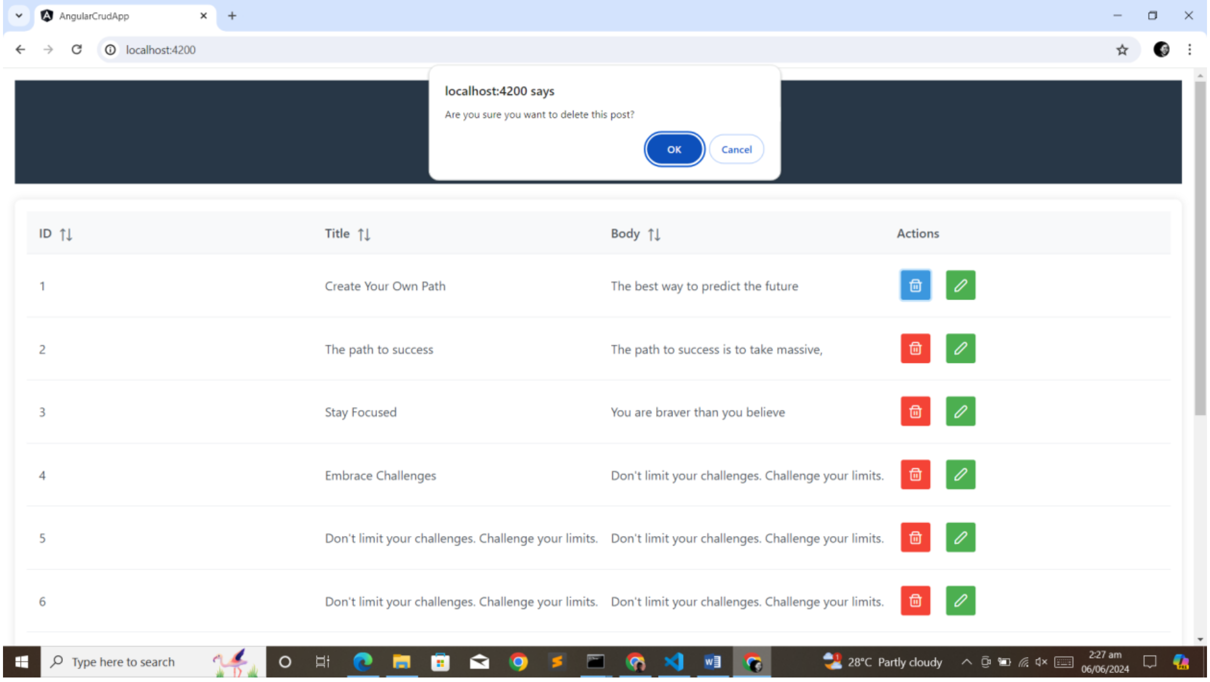


Sample crud operations

ID ⇅	Title ⇅	Body ⇅	Actions
11	The only way to achieve the impossible is to belie	The only way to achieve the impossible is to belie	🗑 ✎
12	The only way to achieve the impossible is to belie	The only way to achieve the impossible is to belie	🗑 ✎

© 2024 All Rights Reserved By Sanjee

Alert box

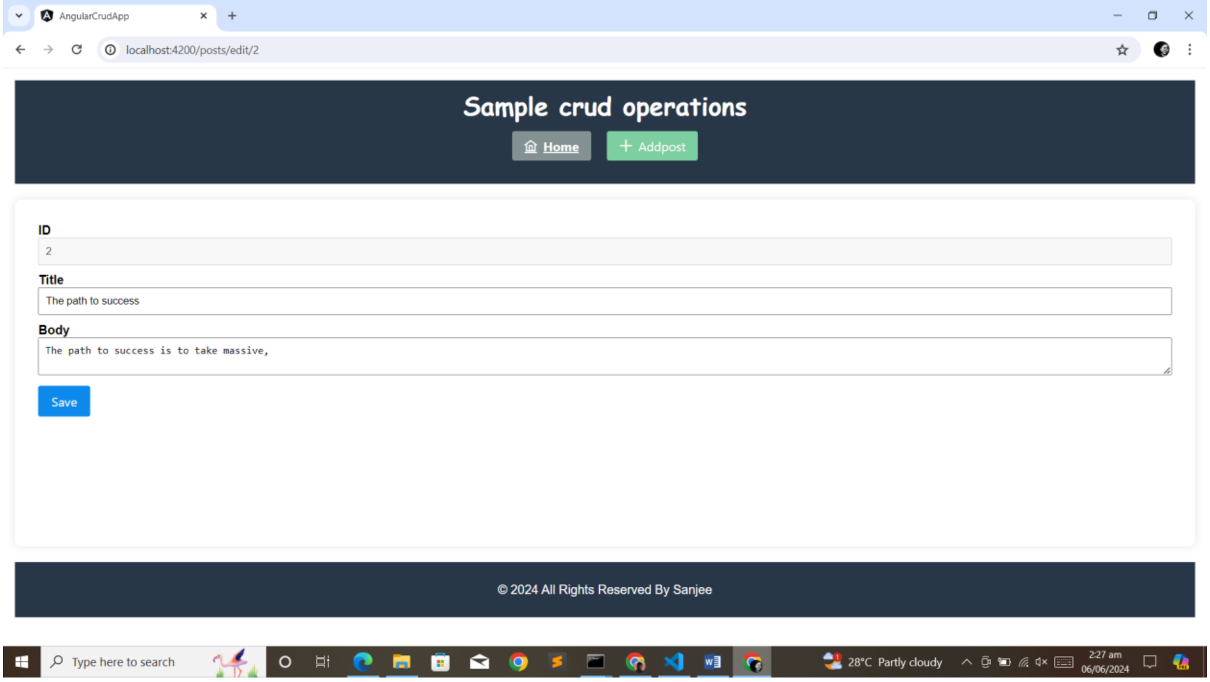


localhost:4200 says

Are you sure you want to delete this post?

ID	Title	Body	Actions
1	Create Your Own Path	The best way to predict the future	
2	The path to success	The path to success is to take massive,	
3	Stay Focused	You are braver than you believe	
4	Embrace Challenges	Don't limit your challenges. Challenge your limits.	
5	Don't limit your challenges. Challenge your limits.	Don't limit your challenges. Challenge your limits.	
6	Don't limit your challenges. Challenge your limits.	Don't limit your challenges. Challenge your limits.	

Update box



Sample crud operations

ID

2

Title

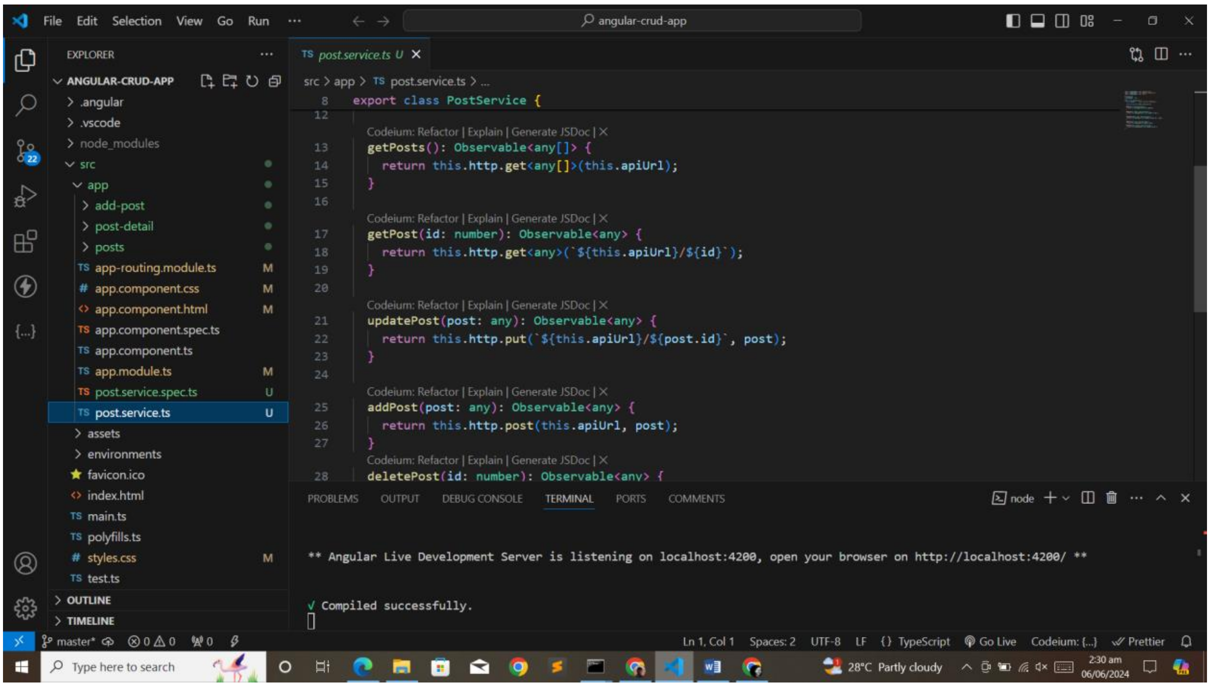
The path to success

Body

The path to success is to take massive,

© 2024 All Rights Reserved By Sanjee

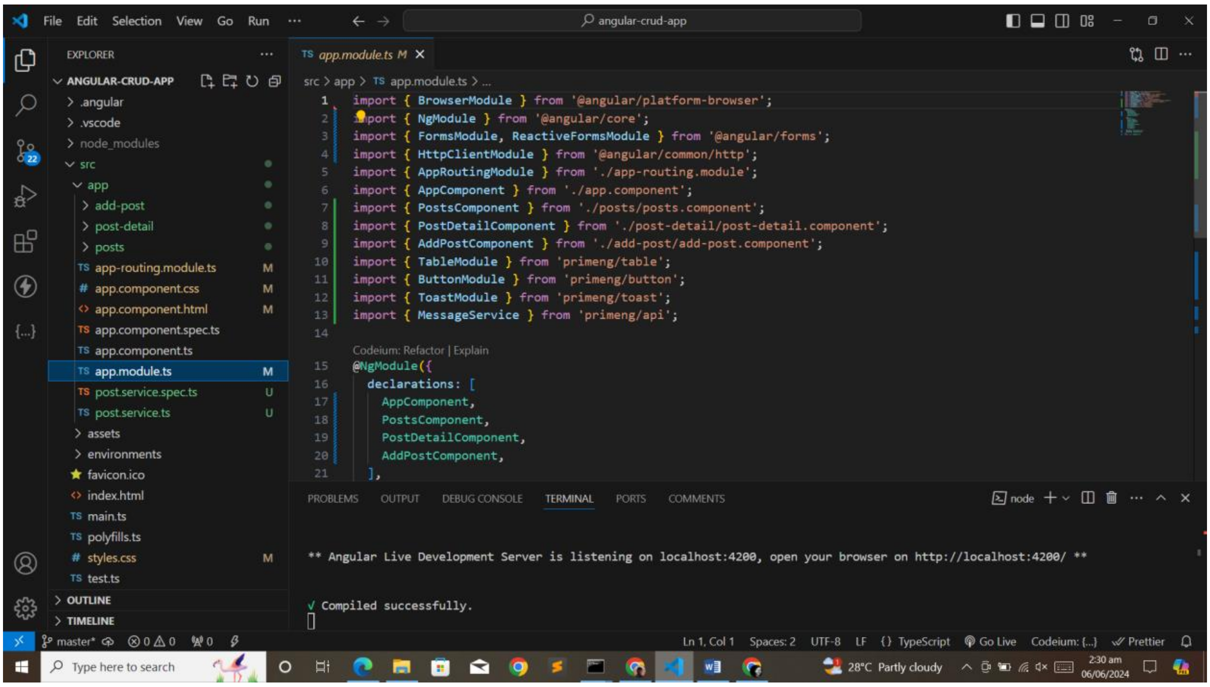
Service ts page



```
export class PostService {

  getPosts(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl);
  }

  getPost(id: number): Observable<any> {
    return this.http.get<any>(`${this.apiUrl}/${id}`);
  }

  updatePost(post: any): Observable<any> {
    return this.http.put(`${this.apiUrl}/${post.id}`, post);
  }

  addPost(post: any): Observable<any> {
    return this.http.post(this.apiUrl, post);
  }
  deletePost(id: number): Observable<any> {
```

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

√ Compiled successfully.
```

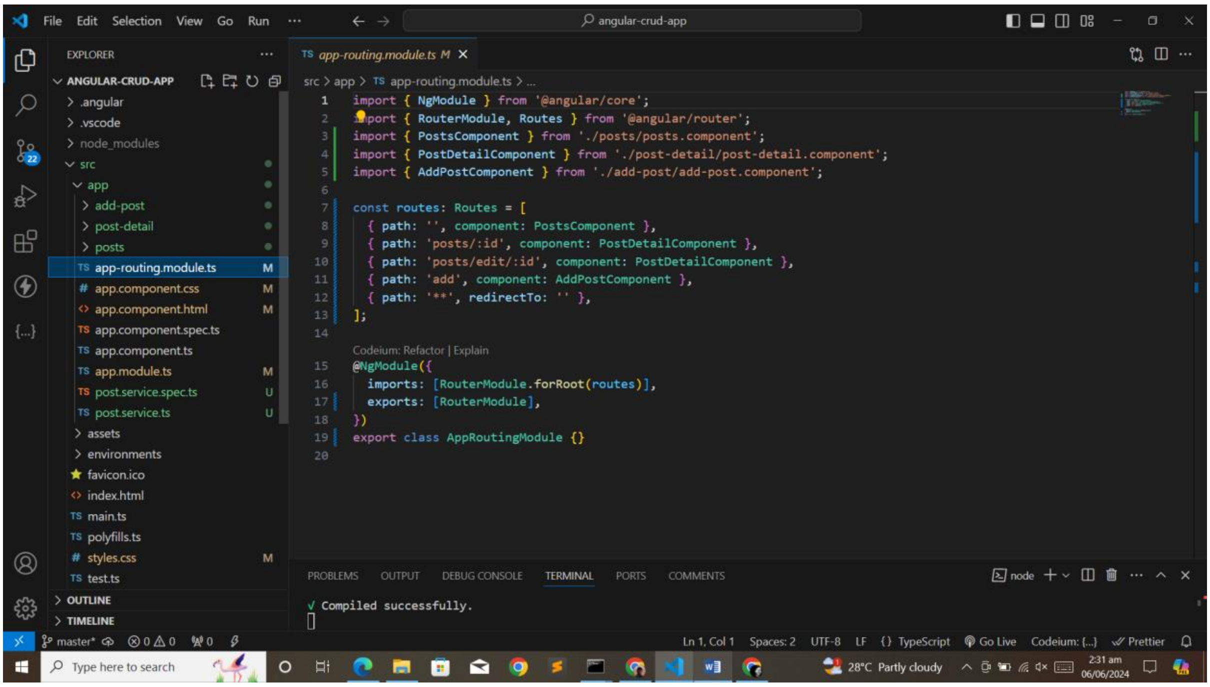
Appmodule.ts



```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { PostsComponent } from './posts/posts.component';
import { PostDetailComponent } from './post-detail/post-detail.component';
import { AddPostComponent } from './add-post/add-post.component';
import { TableModule } from 'primeng/table';
import { ButtonModule } from 'primeng/button';
import { ToastModule } from 'primeng/toast';
import { MessageService } from 'primeng/api';

@NgModule({
  declarations: [
    AppComponent,
    PostsComponent,
    PostDetailComponent,
    AddPostComponent,
  ],
```

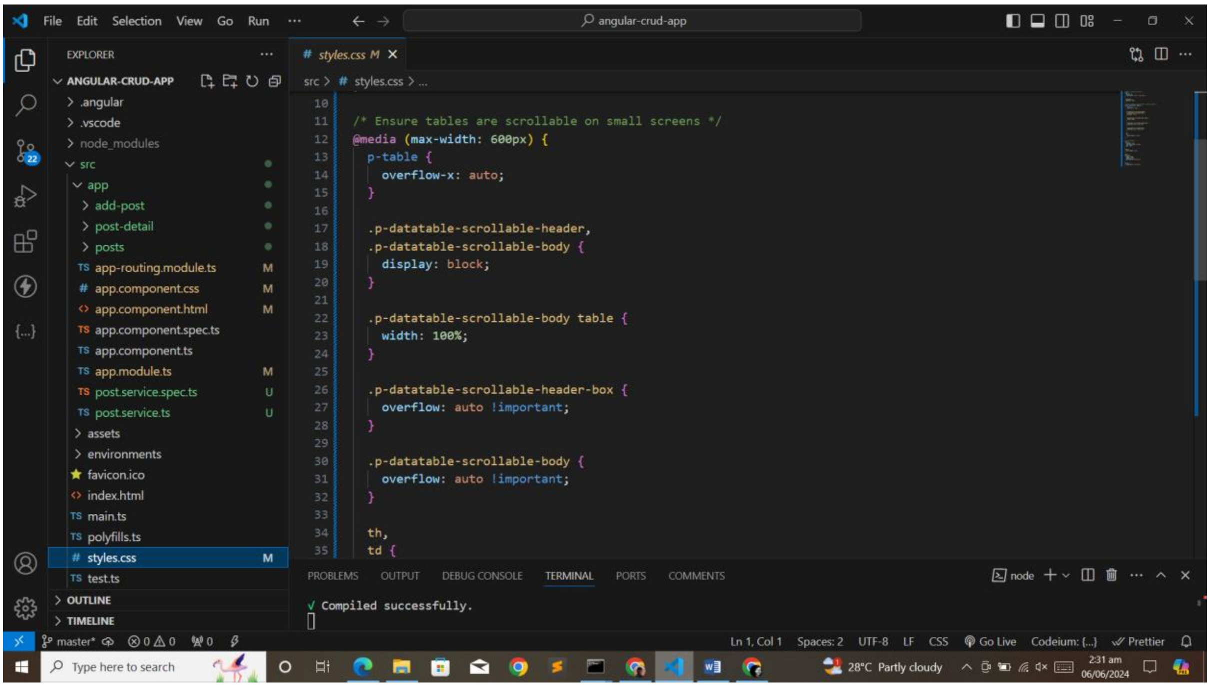
```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

√ Compiled successfully.
```

Routing module



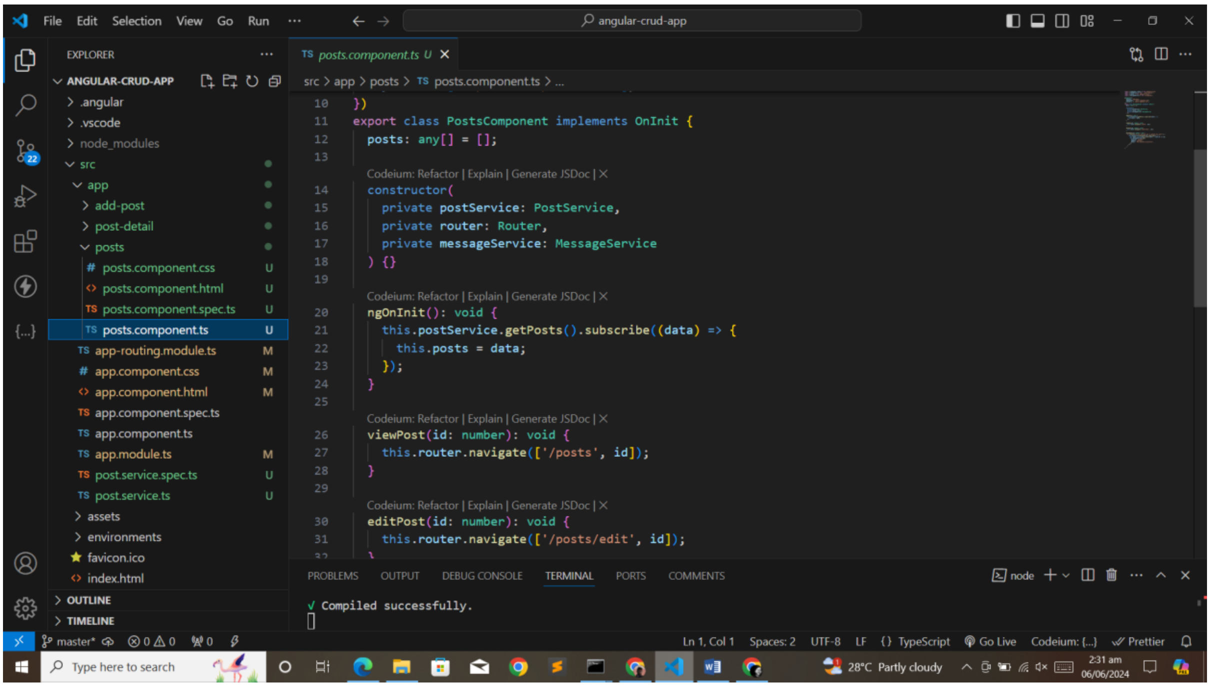
```typescript
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { PostsComponent } from './posts/posts.component';
import { PostDetailComponent } from './post-detail/post-detail.component';
import { AddPostComponent } from './add-post/add-post.component';

const routes: Routes = [
  { path: '', component: PostsComponent },
  { path: 'posts/:id', component: PostDetailComponent },
  { path: 'posts/edit/:id', component: PostDetailComponent },
  { path: 'add', component: AddPostComponent },
  { path: '**', redirectTo: '' },
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

```
√ Compiled successfully.
```



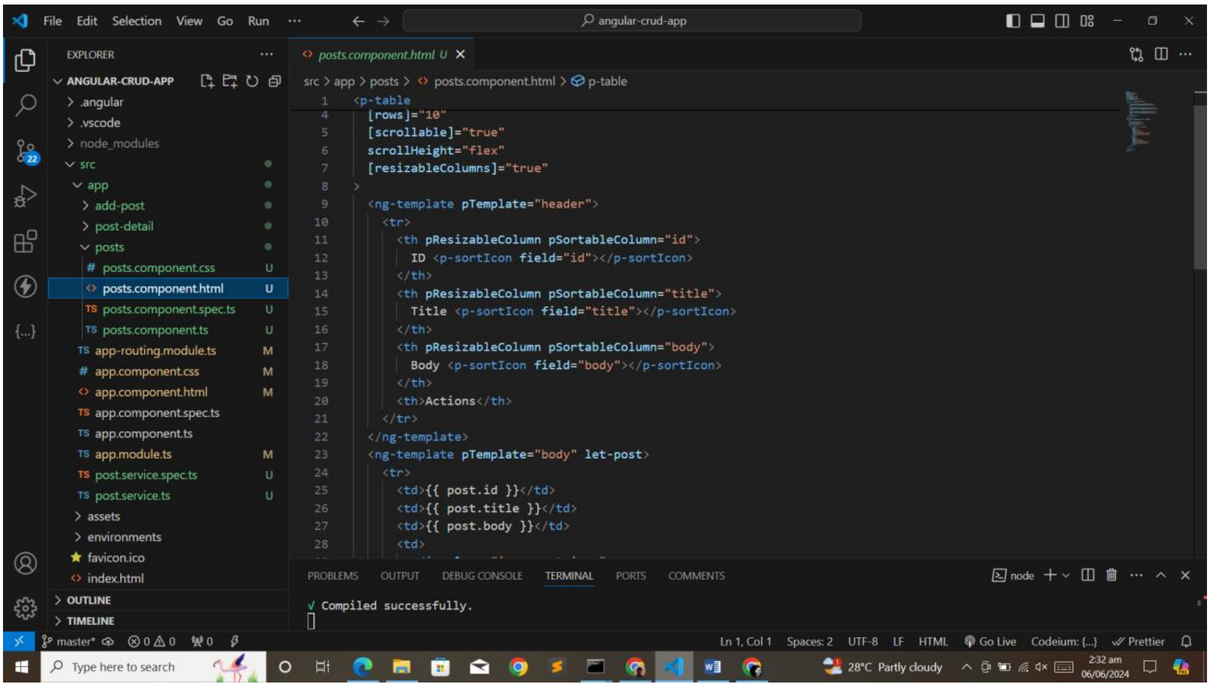
```css

/* Ensure tables are scrollable on small screens */
@media (max-width: 600px) {
  p-table {
    overflow-x: auto;
  }

  .p-datatable-scrollable-header,
  .p-datatable-scrollable-body {
    display: block;
  }

  .p-datatable-scrollable-body table {
    width: 100%;
  }

  .p-datatable-scrollable-header-box {
    overflow: auto !important;
  }

  .p-datatable-scrollable-body {
    overflow: auto !important;
  }

  th,
  td {
```

```
√ Compiled successfully.
```



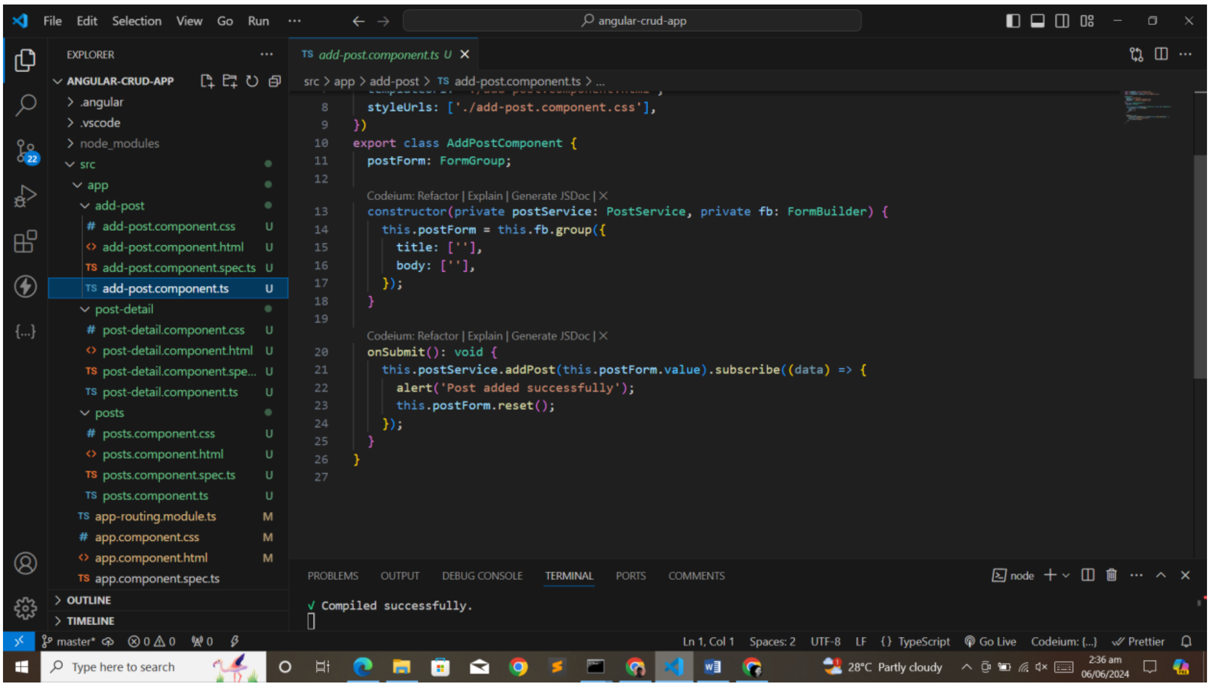
```
})
export class PostsComponent implements OnInit {
  posts: any[] = [];

  Codeium: Refactor | Explain | Generate JSDoc | ×
  constructor(
    private postService: PostService,
    private router: Router,
    private messageService: MessageService
  ) {}

  Codeium: Refactor | Explain | Generate JSDoc | ×
  ngOnInit(): void {
    this.postService.getPosts().subscribe((data) => {
      this.posts = data;
    });
  }

  Codeium: Refactor | Explain | Generate JSDoc | ×
  viewPost(id: number): void {
    this.router.navigate(['/posts', id]);
  }

  Codeium: Refactor | Explain | Generate JSDoc | ×
  editPost(id: number): void {
    this.router.navigate(['/posts/edit', id]);
```

```
√ Compiled successfully.
```



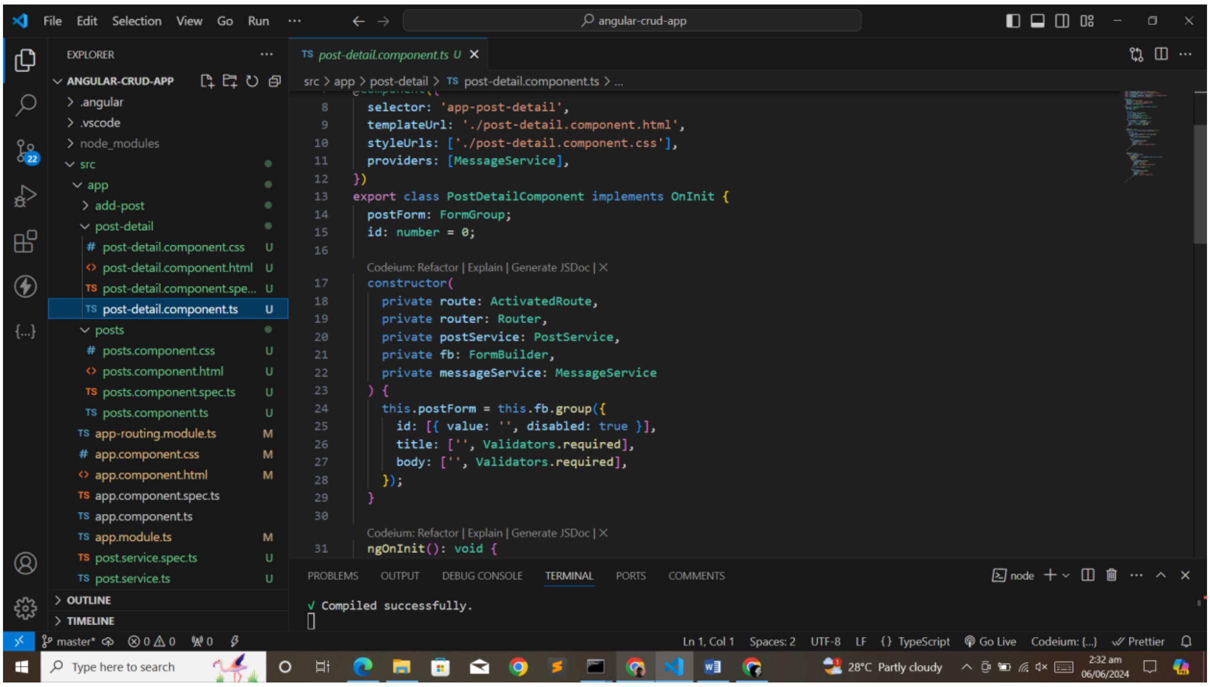
```
<p-table
  [rows]="10"
  [scrollable]="true"
  scrollHeight="flex"
  [resizableColumns]="true"
>
  <ng-template pTemplate="header">
    <tr>
      <th pResizableColumn pSortableColumn="id">
        ID <p-sortIcon field="id"></p-sortIcon>
      </th>
      <th pResizableColumn pSortableColumn="title">
        Title <p-sortIcon field="title"></p-sortIcon>
      </th>
      <th pResizableColumn pSortableColumn="body">
        Body <p-sortIcon field="body"></p-sortIcon>
      </th>
      <th>Actions</th>
    </tr>
  </ng-template>
  <ng-template pTemplate="body" let-post>
    <tr>
      <td>{{ post.id }}</td>
      <td>{{ post.title }}</td>
      <td>{{ post.body }}</td>
      <td>
```

```
√ Compiled successfully.
```



```typescript
  styleUrls: ['./add-post.component.css'],
})
export class AddPostComponent {
  postForm: FormGroup;

  constructor(private postService: PostService, private fb: FormBuilder) {
    this.postForm = this.fb.group({
      title: [''],
      body: [''],
    });
  }

  onSubmit(): void {
    this.postService.addPost(this.postForm.value).subscribe((data) => {
      alert('Post added successfully');
      this.postForm.reset();
    });
  }
}
```

```
√ Compiled successfully.
```



```typescript
  selector: 'app-post-detail',
  templateUrl: './post-detail.component.html',
  styleUrls: ['./post-detail.component.css'],
  providers: [MessageService],
})
export class PostDetailComponent implements OnInit {
  postForm: FormGroup;
  id: number = 0;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private postService: PostService,
    private fb: FormBuilder,
    private messageService: MessageService
  ) {
    this.postForm = this.fb.group({
      id: [{ value: '', disabled: true }],
      title: ['', Validators.required],
      body: ['', Validators.required],
    });
  }

  ngOnInit(): void {
```

```
√ Compiled successfully.
```