

1. Project Setup with Angular CLI

Run:

```
ng new angular-quiz-app --style=scss --routing
cd angular-quiz-app
npm install bootstrap
```

In `angular.json`, add Bootstrap:

```
"styles": [
  "node_modules/bootstrap/dist/css/bootstrap.min.css",
  "src/styles.scss"
]
```

2. Convert Each Page to Angular Components

Use the CLI to generate components:

```
ng generate component pages/home
ng generate component pages/contact
ng generate component pages/frameworks
ng generate component pages/quiz
ng generate component pages/admin
```

Update `app-routing.module.ts`:

```
const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'contact', component: ContactComponent },
  { path: 'frameworks', component: FrameworksComponent },
  { path: 'quiz', component: QuizComponent },
  { path: 'admin', component: AdminComponent },
];
```

3. Convert Scripts to Angular Services + Logic

We will extract the JS logic (`contactBK.js`, `filterBK.js`, `quizquestionsBK.js`, `adminBK.js`) into Angular services or component methods:

Contact (RxJS Form)

Use `ReactiveFormsModule` and RxJS operators (`debounceTime`, etc.)

Create form controls for phone, email, zip.

Quiz (Dynamic Questions)

Create a `QuizService` to fetch/add questions (`assets/quiz.json` or API).

Render quiz in `quiz.component.ts`.

Admin (Manage Questions)

Use FormGroup to create/edit questions.

Two-way bind using [(ngModel)] or FormControl.

Frameworks (Searchable Table)

Use @ViewChild and an RxJS fromEvent for the input.

Filter table using filter + map.

4. Convert HTML to Angular Templates

Take your HTML and refactor each page to be its corresponding .component.html. Reuse shared layout via app.component.html:

```
<router-outlet></router-outlet>
```

Convert adminBK.js to Angular

```
<!-- home.component.html -->
```

```
<div class="container mt-4">
  <h1>Welcome to My Node Site</h1>
  <p>This is the homepage.</p>
</div>
```

```
<!-- contact.component.html -->
```

```
<div class="container mt-5">
  <h1>Contact Form</h1>
  <form [formGroup]="contactForm" (ngSubmit)="onSubmit()">
    <div class="mb-3">
      <label for="phone" class="form-label">Phone</label>
      <input type="text" id="phone" formControlName="phone"
class="form-control" />
    </div>
    <div class="mb-3">
      <label for="email" class="form-label">Email</label>
      <input type="email" id="email" formControlName="email"
class="form-control" />
    </div>
    <div class="mb-3">
      <label for="zip" class="form-label">Zip Code</label>
      <input type="text" id="zip" formControlName="zip" class="form-
control" />
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
  <div id="response" class="mt-3">{{ responseMessage }}</div>
```

</div>

//contact.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'app-contact',
  templateUrl: './contact.component.html'
})
export class ContactComponent {
  contactForm: FormGroup;
  responseMessage = '';

  constructor(private fb: FormBuilder) {
    this.contactForm = this.fb.group({
      phone: ['', [Validators.required, Validators.pattern(/
^\\d{10}$/) ]],
      email: ['', [Validators.required, Validators.email]],
      zip: ['', [Validators.required]]
    });
  }

  onSubmit(): void {
    if (this.contactForm.valid) {
      this.responseMessage = 'Form submitted successfully!';
    } else {
      this.responseMessage = 'Please correct the errors in the form.';
    }
  }
}
```

<!-- frameworks.component.html -->

```
<div class="container mt-4">
  <h1>Popular JavaScript Tools</h1>
  <input type="text" [(ngModel)]="searchTerm" class="form-control
my-3" placeholder="Search..." />
  <table class="table">
    <thead>
      <tr><th>Name</th><th>Year</th><th>Description</th></tr>
    </thead>
    <tbody>
      <tr *ngFor="let fw of filteredFrameworks()">
        <td>{{ fw.name }}</td>
        <td>{{ fw.year }}</td>
        <td>{{ fw.description }}</td>
      </tr>
    </tbody>
  </table>
</div>
```

```
</table>
</div>
```

```
//frameworks.component.ts
```

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-frameworks',
  templateUrl: './frameworks.component.html'
})
export class FrameworksComponent {
  searchTerm = '';
  frameworks = [
    { name: 'React', year: 2013, description: 'UI library' },
    { name: 'Angular', year: 2010, description: 'Full-featured
framework' },
    { name: 'Vue', year: 2014, description: 'Progressive framework' }
  ];

  filteredFrameworks() {
    return this.frameworks.filter(f =>
      f.name.toLowerCase().includes(this.searchTerm.toLowerCase())
    );
  }
}
```

```
<!-- quiz.component.html -->
```

```
<div class="container mt-4 fs-6">
  <h1>Quiz</h1>
  <div *ngFor="let question of quizData; let i = index">
    <h5>{{ question.question }}</h5>
    <div *ngFor="let choice of question.choices">
      <label>
        <input type="radio" name="q{{ i }}" (change)="selectAnswer(i,
choice)" />
        {{ choice }}
      </label>
    </div>
  </div>
  <button class="btn btn-primary mt-3" (click)="submitQuiz()">Submit</
button>
  <div id="result" class="mt-3">{{ resultMessage }}</div>
</div>
```

```
//quiz.component.ts
```

```
import { Component } from '@angular/core';
```

```

@Component({
  selector: 'app-quiz',
  templateUrl: './quiz.component.html'
})
export class QuizComponent {
  quizData = [
    { question: 'What is Angular?', choices: ['Library', 'Framework'],
      answer: 'Framework' }
  ];
  userAnswers: string[] = [];
  resultMessage = '';

  selectAnswer(index: number, choice: string) {
    this.userAnswers[index] = choice;
  }

  submitQuiz() {
    let score = 0;
    this.quizData.forEach((q, i) => {
      if (this.userAnswers[i] === q.answer) score++;
    });
    this.resultMessage = `You got ${score} out of ${
      this.quizData.length}`;
  }
}

```

<!-- admin.component.html -->

```

<div class="container mt-4">
  <h1>Quiz Admin Panel</h1>
  <div *ngFor="let q of quizData; let i = index" class="border p-3
mb-3">
    <input [(ngModel)]="q.question" class="form-control mb-2" />
    <input [(ngModel)]="q.choicesString" class="form-control mb-2"
placeholder="Choices comma-separated" />
    <input [(ngModel)]="q.answer" class="form-control" />
  </div>

  <form #f="ngForm" (ngSubmit)="addQuestion(f)">
    <h3>Add New Question</h3>
    <input name="newQuestion" ngModel required class="form-control
mb-2" placeholder="Question" />
    <input name="newChoices" ngModel required class="form-control
mb-2" placeholder="Choices (comma-separated)" />
    <input name="newAnswer" ngModel required class="form-control mb-2"
placeholder="Answer" />
    <button class="btn btn-success mt-2">Add Question</button>
  </form>
</div>

```

```

//admin.component.ts

import { Component } from '@angular/core';

@Component({
  selector: 'app-admin',
  templateUrl: './admin.component.html'
})
export class AdminComponent {
  quizData: { question: string; choicesString: string; answer:
string }[] = [];

  addQuestion(form: any): void {
    const newQuestion = form.value.newQuestion;
    const newChoices = form.value.newChoices;
    const newAnswer = form.value.newAnswer;
    this.quizData.push({
      question: newQuestion,
      choicesString: newChoices,
      answer: newAnswer
    });
    form.resetForm();
  }
}

```