Sprint-2 Plan

Employee Detail Management

1. Establishing Project Folder Architecture
2. Server App Initialization
3. **Sprint-2 Execution** [2(a), **2(b)**, 2(c), 2(d), 2(e), 2(f)]
4. Completing Sprint-1 Objectives

2(a) Employee Module - Analysis, Design & DB-Preparation
Data Design, UI Design,   Sample Data, Database Preparation

2(b) Employee View

1. Controller Design
2. Coding-Server App
   Version-1 (Client Pagination)
      Entity Coding, DAO Coding, Controller Coding, Testing
   Version-2 (Server Pagination)
      Controller Modification, Testing
3. Coding-Client App
   Version-1 (Client Pagination)
      UI Preparation, Entity Coding, Service Coding, Controller Coding, Testing
   Version-2 (Server Pagination)
      Service Modification, Controller Modification, Testing

2(c) Employee Search
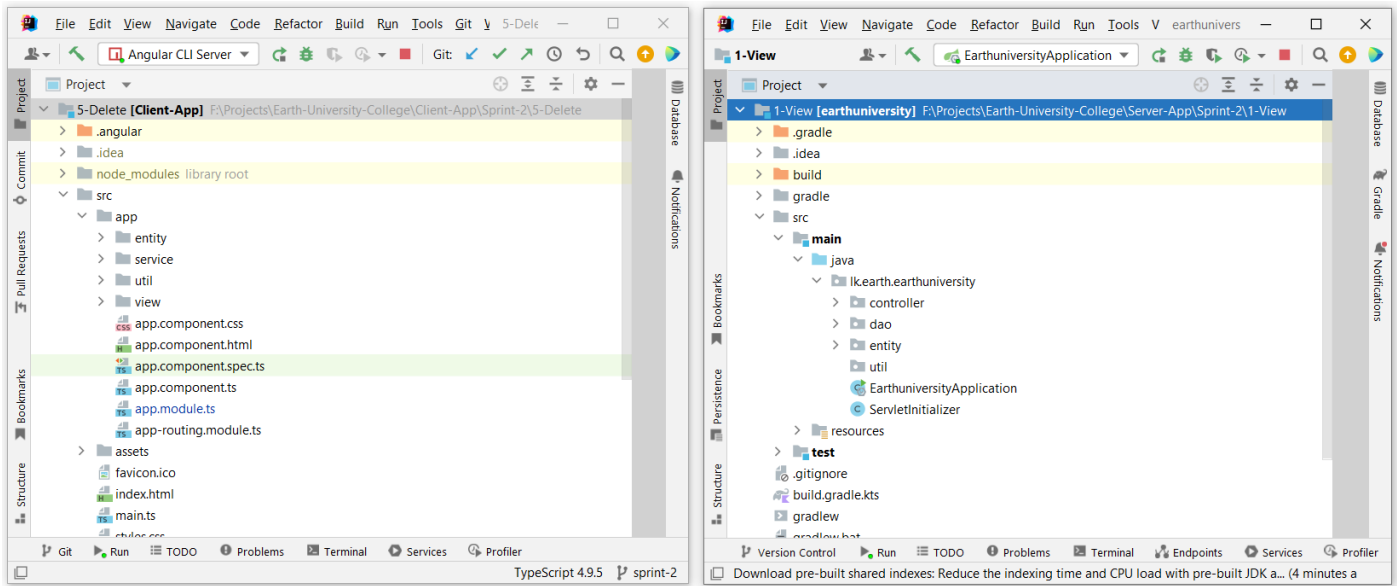**2(d) Employee Insert**
2(e) Employee Update
2(f) Employee Delete

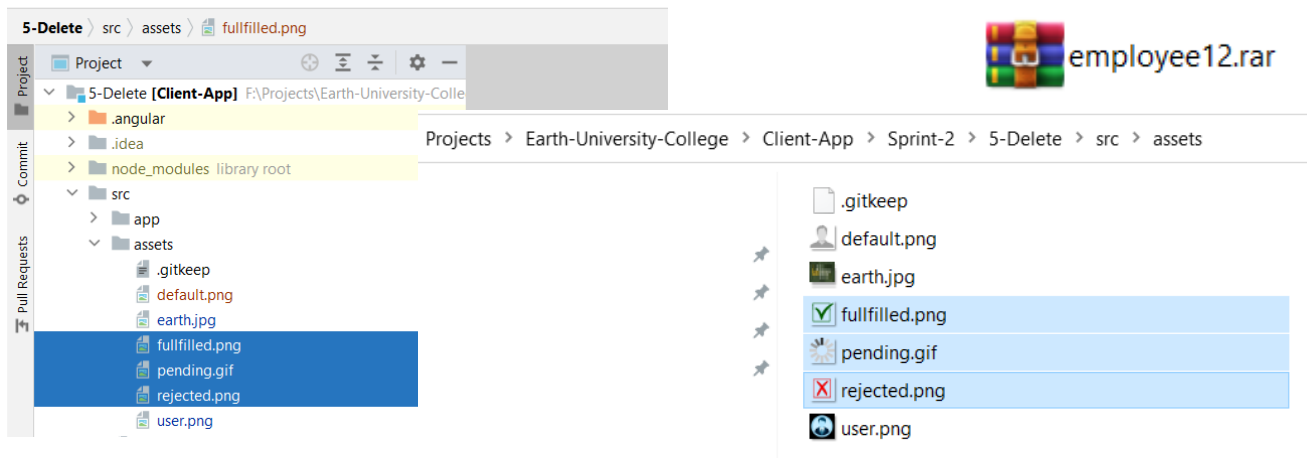===

2(b) Employee View
1. Controller Design
2. Coding Server-App
3. Coding Client-App
   Version-1 (Client Pagination)
      1. Client App Preparation for Sprint-2
      2. UI Preparation with Static Data & Testing
      3. Coding Entity Classes
      4. Data Loading with Dynamic Data & Testing
      5. Coding Service Classes for Dynamic Data & Testing (With & Without DI)
      6. Error Correction – Server App – Employee Data Service
      7. Data loading from Server & Testing, **Promise-Pending/Resolve/Reject Status**
      8. Client Pagination
      9. Client Search
   Version-2 (Server Pagination)

===

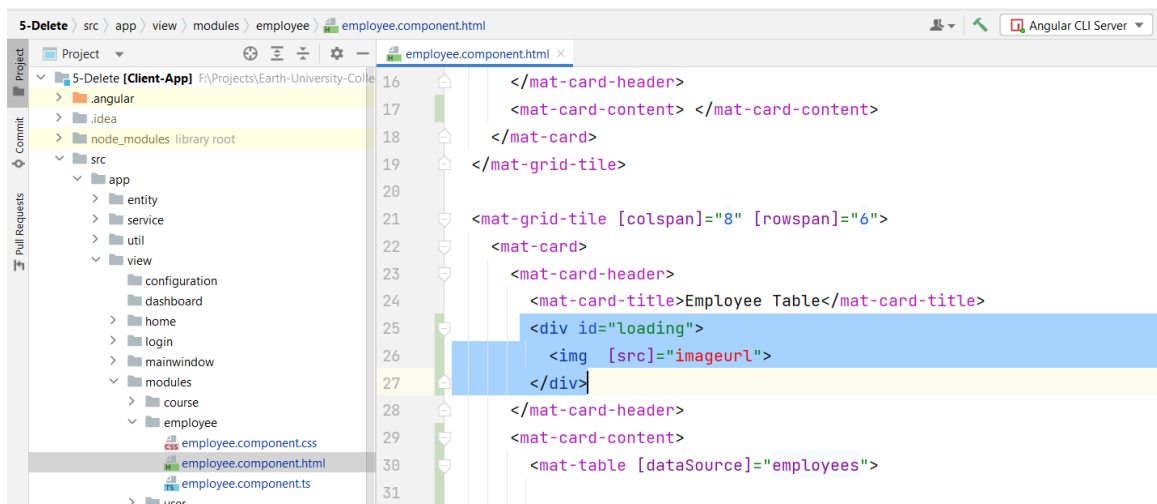# Promise-Pending/Resolve/Reject Status

1. Open and Run both the Server-App & Client-App in separate IntelliJ-IDEA windows.
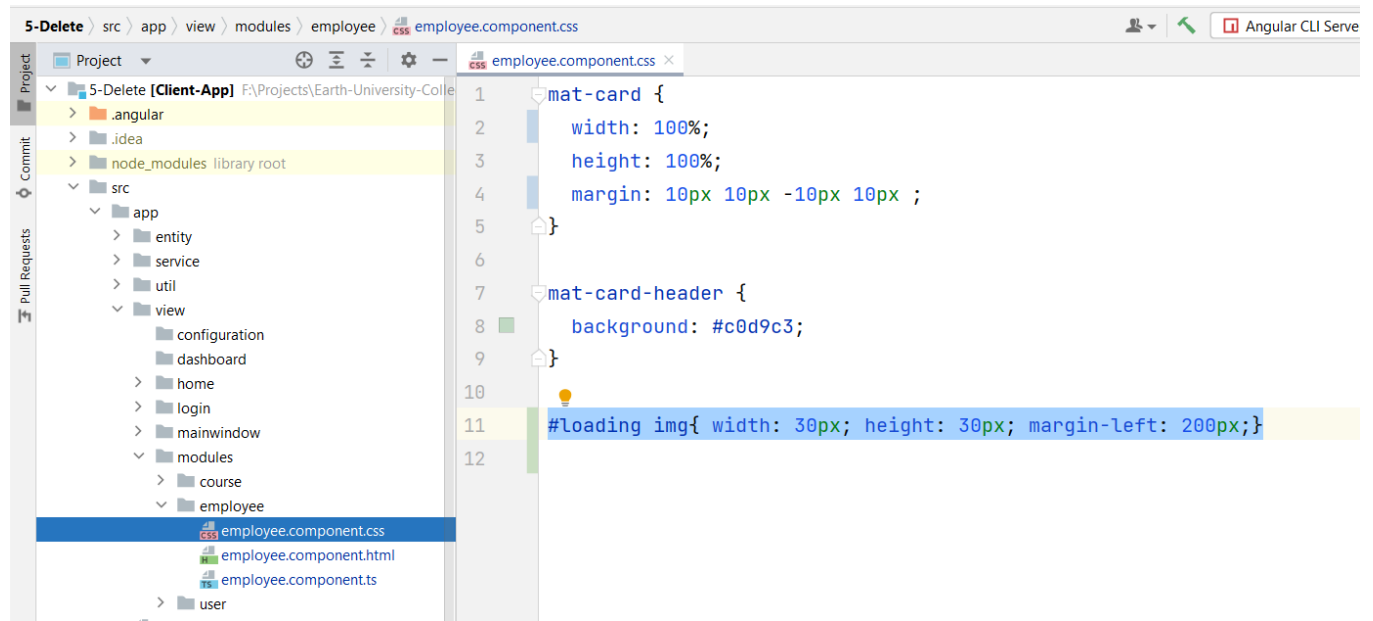   Server App – 8080, Client App – 4200
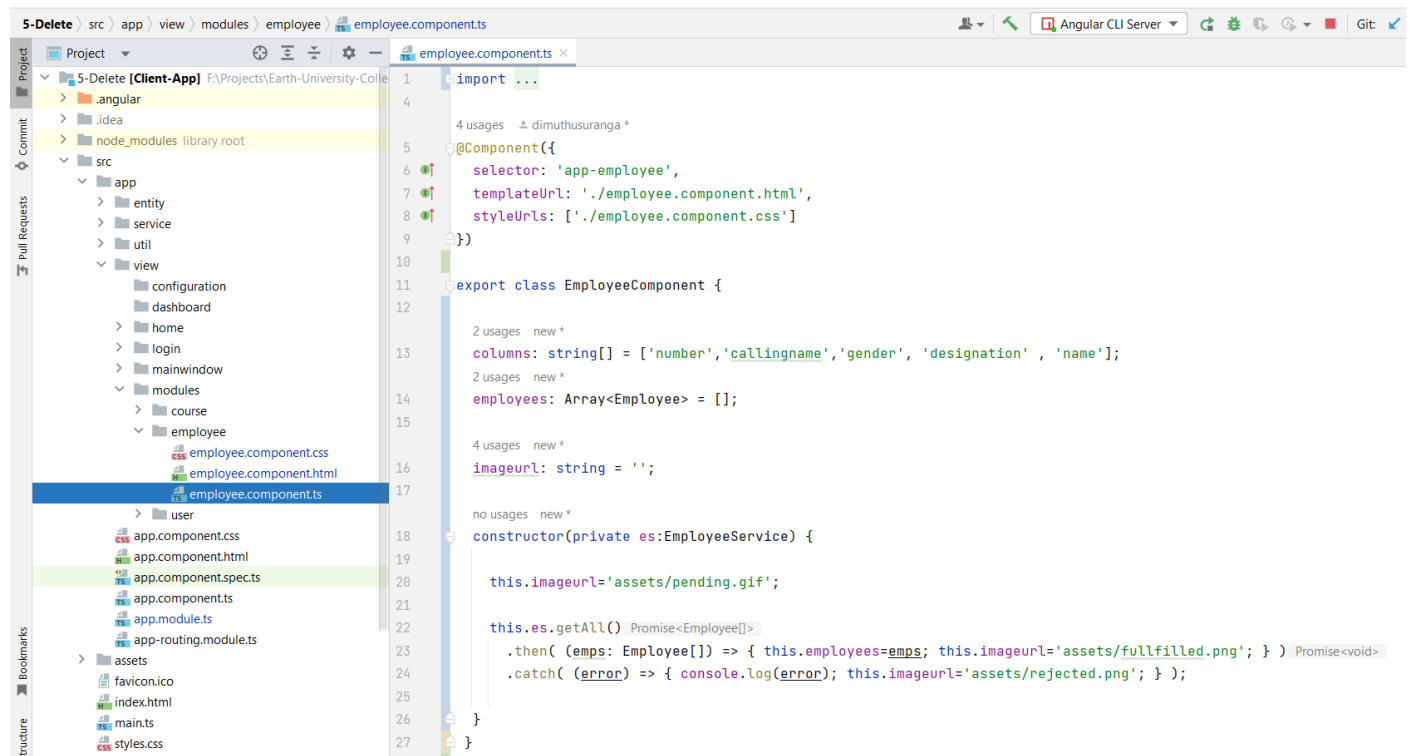

2. Add following 3 pictures to the "assets" as follows

3. Change the HTML File as follows

4. Change the CSS File as follows

The CSS file (employee.component.css) contains:

```css
mat-card {
    width: 100%;
    height: 100%;
    margin: 10px 10px -10px 10px ;
}


mat-card-header {
    background: #c0d9c3;
}

#loading img{ width: 30px; height: 30px; margin-left: 200px;}
```

5. Change the TS file as follows

The TS file (employee.component.ts) contains:

```typescript
import ...

@Component({
    selector: 'app-employee',
    templateUrl: './employee.component.html',
    styleUrls: ['./employee.component.css']
})

export class EmployeeComponent {

    columns: string[] = ['number','callingname','gender', 'designation' , 'name'];
    employees: Array<Employee> = [];

    imageurl: string = '';

    constructor(private es:EmployeeService) {

        this.imageurl='assets/pending.gif';

        this.es.getAll() Promise<Employee[]>
        .then( (emps: Employee[]) => { this.employees=emps; this.imageurl='assets/fullfilled.png'; } ) Promise<void>
        .catch( (error) => { console.log(error); this.imageurl='assets/rejected.png'; } );

    }
}
```

6. Testing

(a) Pending & Succeed

Refresh the employee window and observe for a few milliseconds before successes

Number	Calling Name	Gender	Designation	Full Name

Employee Table ✓

Number	Calling Name	Gender	Designation	Full Name
2201	Ashan	Male	Demonstrator L1	Ashan Pathum
2202	Rukmal	Male	Demonstrator L1	Dileesha Rukmal
2203	Imran	Male	Demonstrator L1	Thahir Imran
2301	Lakshan	Male	Demonstrator L1	Lakshan Ruwinda
2302	Arshad	Male	Demonstrator L1	Arshad Ahamad
2303	Imasha	Female	Demonstrator L1	Imasha Gaupadi

(b) Pending & Failed

Stop the Server App and look for Pending and Failed Status

Employee Table

Number	Calling Name	Gender	Designation	Full Name

Employee Table ✗

Number	Calling Name	Gender	Designation	Full Name

Discuss,

What is Promise?

What are the status that a Promise can exists,

How to handle each of the different status,

What is the importance of Promise in "async" functions calls?

Compare Observable over Promise in "async" function calls

Alternative implementations of Service Calls with Promise and Observable
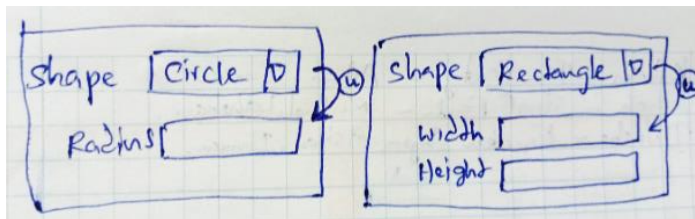
Discuss,

What is rendering UI Components by the Controller?

Template-Driven Rendering, Reactive Rendering, Dynamic Rendering

Renderer2, ViewChild and ElementRef in DOM-Manipulation

`ngOnInit(), ngAfterViewInit() and other Lifecycle hooks`

DOM Manipulation Requirement on updating View dynamically depend on user Inputs

Department of Information Technology
Earth University College