COLLEGE CODE :9623

COLLEGE NAME :Amrita College of Engineering and Technology

DEPARTMENT :Computer Science Engineering

STUDENT NM-ID : B1454C10A8B0D939C4A4E1643EC6FDF7

ROLL NO :962323104099

DATE :22-09-2025

Completed the project named as Phase 3

TECHNOLOGY PROJECT NAME : AngularJS with SQL Integration

SUBMITTED BY,

NAME: R.S.SIVA PRIYA

MOBILE NO:9952856585

PHASE 3 : MVP IMPLEMENTATION

1. Project Setup

- **Definition:**

  Set the foundation by defining the project goal, tech stack, and architecture. This ensures a clear roadmap and scalable structure.

- **Basic Code Snippet:**

```
var app = angular.module('app', []);
app.controller('MainCtrl', ($scope) => {
  $scope.msg = "AngularJS + SQL Integration";
});
```

2. Core Features Implementation

- **Definition:**

  Implement essential CRUD operations that connect frontend UI with backend logic and database. AngularJS services communicate with Express routes, which execute SQL queries to manage data.

- **Example Code:**

```
app.service('UserService', ($http) => ({
  getUsers: () => $http.get('/api/users'),
  addUser: user => $http.post('/api/users', user)
}));

app.controller('UserCtrl', ($scope, UserService) => {
```

```
  UserService.getUsers().then(res => $scope.users =
res.data);
  $scope.addUser = user =>
UserService.addUser(user).then(res =>
$scope.users.push(res.data));
});
```

3. Data Storage(local State/Database)

- **Definition:**

  LocalStorage is used for small, quick-access data like tokens, while MySQL handles Manage temporary client data via localStorage and persistent data in a MySQL database. structured and relational data.

- **Example Code:**

```
localStorage.setItem('token', 'abc123');
const token = localStorage.getItem('token');
console.log(token);
```

- **SQL Table:**

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100)
);
```

4. Testing Core Features

- **Definition:**

  Use automated tests to verify frontend services and backend API

functionality. This ensures reliability and helps catch bugs early before deployment.

- **Example Frontend Test:**

```
describe('UserService', () => {
  beforeEach(module('app'));
  var UserService, $httpBackend;

  beforeEach(inject((_UserService_, _$httpBackend_) => {
    UserService = _UserService_;
    $httpBackend = _$httpBackend_;
  }));

  it('fetches users', () => {
    $httpBackend.expectGET('/api/users').respond([{name:
'Test'}]);
    UserService.getUsers().then(res =>
expect(res.data.length).toBe(1));
    $httpBackend.flush();
  });
});
```

Example Backend Test:

```
const request = require('supertest');
const app = require('../server');

describe('GET /api/users', () => {
  it('returns users', done => {
    request(app).get('/api/users').expect(200, done);
  });
});
```

5. Version Control (GitHub)

- **Definition:**

  Track code changes and collaborate using Git and GitHub. Commit changes, push to remote repos, and manage branches for smooth teamwork and project history.

- **Basic Commands:**

```
git init
git add .
git commit -m "Initial commit"
git remote add origin
https://github.com/username/repo.git
git push -u origin main
```

- **.gitignore:**

```
node_modules/
.env
*.log
```