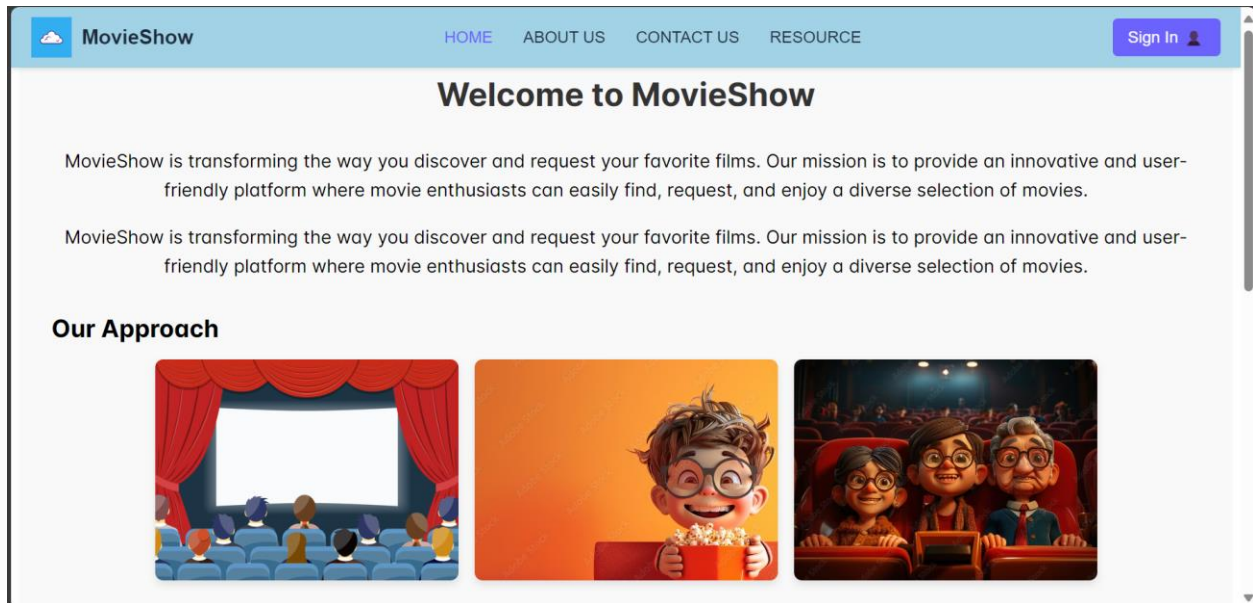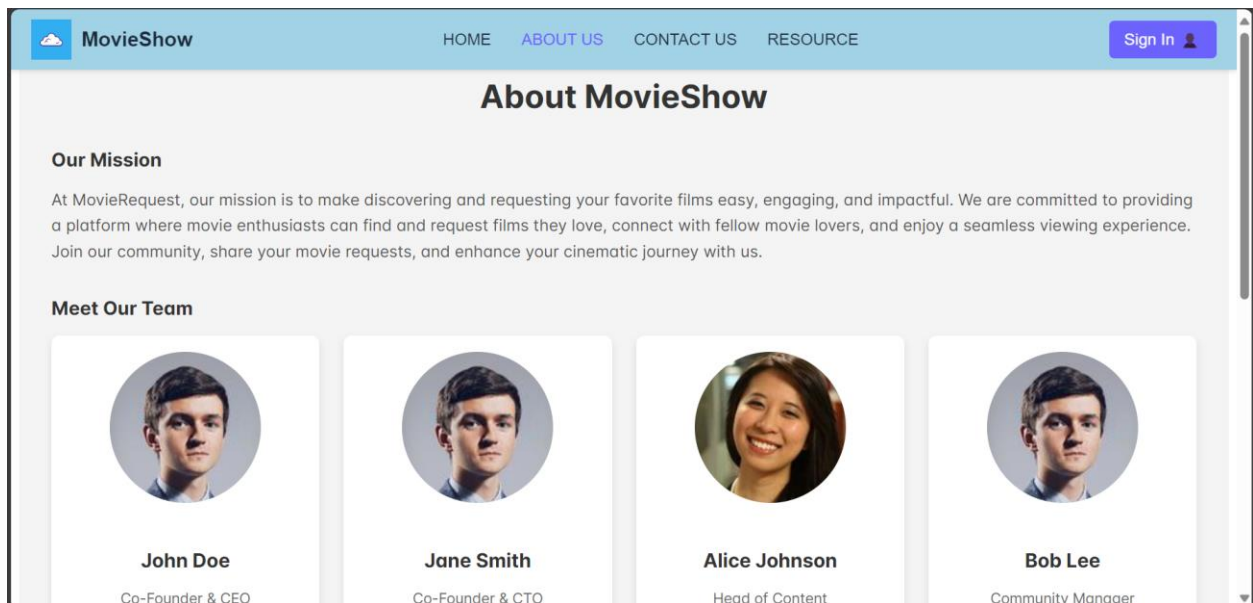# SCREEN SHOTS OF APP
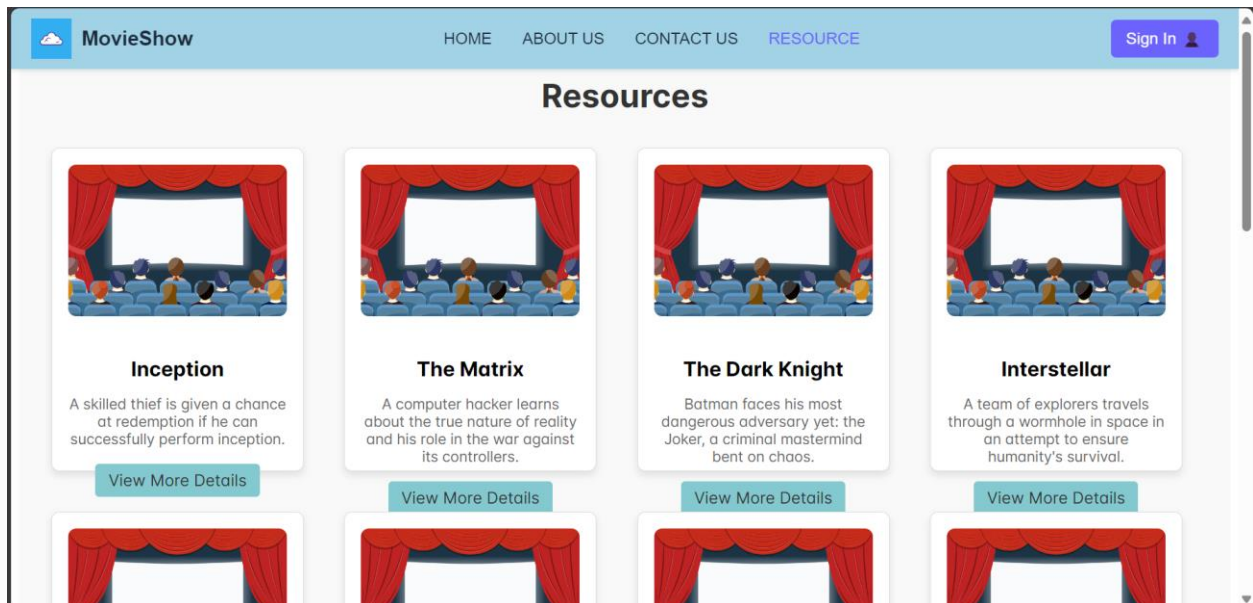
1 . Home

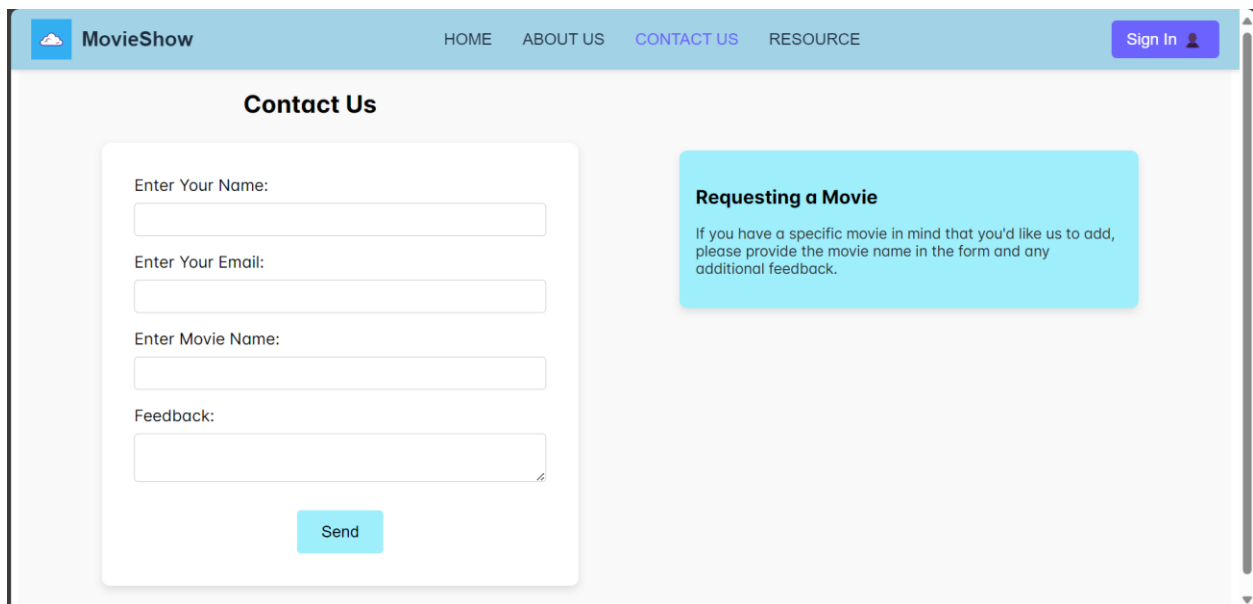

2. About us

3. Resource

4. Contact Us

5. Contact Us after submitting form

5. Footer

6. Test Cases

ContactusComponent
- should have initial formData values as empty strings
- should create the component
- SPEC HAS NO EXPECTATIONS should set error message on failed email send
- SPEC HAS NO EXPECTATIONS should reset formData and set success message on successful email send
- should call sendEmail method when form is submitted
- should display the message when message property is set

AppComponent
- checking for Home Route
- checking for About Us Route
- checking for contact us route
- checking for resources route

AboutusComponent
- should render the mission statement
- should create the AboutusComponent
- should include the FooterComponent
- should display team member names

FooterComponent
- should display the footer bottom content
- should display the subscribe section with correct form
- should display the stats section with correct numbers
- should display the popular categories section
- should display the logo section with correct text
- should render the footer element

HomeComponent
- should render the "Our Approach" section heading
- should render the "About Us" section heading
- should include the FooterComponent

AboutusComponent
- should create the AboutusComponent
- should render the mission statement
- should display team member names
- should include the FooterComponent

ResourcesComponent
- should display correct Title
- should display correct Image URL
- should display the title "Resources"
- should render resource cards
- should create
- should render the footer component
- should display correct Description

ContactusComponent
- should create the component
- should have initial formData values as empty strings
- should call sendEmail method when form is submitted
- SPEC HAS NO EXPECTATIONS should reset formData and set success message on successful email send
- SPEC HAS NO EXPECTATIONS should set error message on failed email send
- should display the message when message property is set

AppComponent
- checking for About Us Route
- checking for Home Route
- checking for resources route
- checking for contact us route

HomeComponent
- should include the FooterComponent
- should render the "About Us" section heading
- should render the "Our Approach" section heading
- should render the "Welcome to MovieShow" heading
- should create

7. Code Coverage

All files

89.65% Statements 26/29   **100%** Branches 1/1   **62.5%** Functions 5/8   **87.5%** Lines 21/24

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter: [   ]

File ▲		Statements ⇕		Branches ⇕		Functions ⇕		Lines ⇕	
app		72.72%	8/11	100%	0/0	25%	1/4	70%	7/10
app/components/aboutus		100%	2/2	100%	0/0	100%	0/0	100%	1/1
app/components/contactus		100%	9/9	100%	1/1	100%	4/4	100%	9/9
app/components/footer		100%	2/2	100%	0/0	100%	0/0	100%	1/1
app/components/home		100%	2/2	100%	0/0	100%	0/0	100%	1/1
app/components/resources		100%	3/3	100%	0/0	100%	0/0	100%	2/2

Code coverage generated by istanbul at 2024-08-10T06:34:47.088Z

PROJECT WORK BREAK DOWN STRUCTURE:

Project: MovieShow Website

1. **Project Management**
   - 1.1 Project Planning
     - 1.1.1 Define Project Scope
     - 1.1.2 Schedule and Budget
     - 1.1.3 Risk Management
2. **Frontend Development**
     - 2.1.1 Design HTML/CSS
     - 2.1.2 Implement Form Validation
     - 2.1.3 User Interface Testing
   - 2.2 Product Cards
     - 2.2.1 Design Layout
     - 2.2.2 Implement Styling

- - 2.2.3 Responsive Design
  - 2.3 Angular Components
    - 2.3.1 HomeComponent
      - 2.3.1.1 Implement Component Logic
      - 2.3.1.2 Design UI
      - 2.3.1.3 Unit Testing
    - 2.3.2 AboutUsComponent
      - 2.3.2.1 Implement Component Logic
      - 2.3.2.2 Design UI
      - 2.3.2.3 Unit Testing
    - 2.3.3 ResearchComponent
      - 2.3.3.1 Implement Component Logic
      - 2.3.3.2 Design UI
      - 2.3.3.3 Unit Testing
    - 2.3.4 FooterComponent
      - 2.3.4.1 Implement Component Logic
      - 2.3.4.2 Design UI
      - 2.3.4.3 Unit Testing
    - 2.3.5 ContactusComponent
      - 2.3.5.1 Implement Component Logic
      - 2.3.5.2 Design UI
      - 2.3.5.3 Unit Testing

3. **Backend Development**
   - 3.1 Database Setup
     - 3.1.1 Configure MongoDB Atlas
     - 3.1.2 Define Database Schema
   - 3.2 API Development

- 3.2.1 User Authentication
- 3.2.2 Product Management
- 3.2.3 Contact Form Handling

4. Testing and Quality Assurance

- 4.1 Unit Testing
  - 4.1.1 Angular Component Tests
  - 4.1.2 API Endpoint Tests
- 4.2 Integration Testing
  - 4.2.1 Frontend-Backend Integration
  - 4.2.2 User Flow Testing
- 4.3 User Acceptance Testing
  - 4.3.1 Test Scenarios
  - 4.3.2 Feedback Collection and Adjustments

5. Maintenance and Support

- 6.1 Bug Fixes
  - 6.1.1 Identify and Resolve Issues
  - 6.1.2 Patch Updates
- 6.2 Feature Enhancements
  - 6.2.1 Collect User Feedback
  - 6.2.2 Implement Improvements