

To create the web application using Angular, you can follow these steps:

Step 1: Set up the Angular Project

1. Install Angular CLI (Command Line Interface) globally on your system.
2. Open a command prompt and navigate to the desired directory where you want to create the project.
3. Run the following command to create a new Angular project:

```
ng new tu-young-professionals-form
```

4. Change to the project directory:

```
cd tu-young-professionals-form
```

5. Launch the project in a development server:

```
ng serve
```

Step 2: Create Components and Services

1. Create a new component for the request info page:

```
ng generate component request-info
```

2. Create a new component for the confirmation page:

```
ng generate component confirmation
```

3. Create a new service for handling data and API requests:

```
ng generate service data
```

Step 3: Design and Implement the Request Form

1. Open the `request-info.component.html` file and add the form elements according to the provided requirements.
2. Use Angular's reactive forms approach to handle form validation and submission. Create a FormGroup with the required form controls and validators.
3. Implement the form submission method in the `request-info.component.ts` file. This method should handle the form submission, validate the data, and send a POST request to the API to save the data.
4. Implement the email validation for the email address field using Angular's built-in email validator.

Step 4: Implement Confirmation Page and Email Sending

1. Open the `confirmation.component.html` file and design the

confirmation page layout.

2. In the ``request-info.component.ts`` file, after successfully submitting the form and receiving a response from the API, navigate to the confirmation page using Angular's router.
3. In the ``confirmation.component.ts`` file, you can access the submitted data from the API response and display it on the confirmation page.
4. Implement email sending using a suitable email service or library. You can use the nodemailer package in a Node.js environment to send the confirmation email upon form submission.

Step 5: Implement List of Requests and Marking as Complete

1. Create a new component for displaying the list of requests and a component for marking a request as complete.
2. Create a new route for the list of requests and the mark as complete functionality.
3. In the list component, fetch the list of requests from the API using the ``data`` service and display them.
4. Implement the functionality to mark a request as complete using the ``data`` service and the API.

Step 6: Connect to MongoDB via API

1. Install the required dependencies for Express.js and MongoDB.
2. Create an Express.js server in the project and set up the necessary routes and endpoints for handling API requests related to the form data.
3. Use the MongoDB driver or an ODM (Object-Document Mapper) like Mongoose to connect to the MongoDB database.
4. Implement the API endpoints to handle data persistence, retrieval, and marking requests as complete.

Step 7: Styling and UI Enhancement

1. Use CSS or a UI library/framework like Bootstrap to enhance the styling and layout of the form, confirmation page, and request list.
2. Add appropriate styles and visual feedback for form validation errors and success messages.

These are the general steps to implement the web application form using Angular. You can adapt and modify the code based on your specific requirements and preferences.