ALGORITHM FOR PHASE END PROJECT EVENT MANAGEMENT APPLICATION

1. **Import the necessary dependencies and modules.**
2. **Define the SignupComponent class, which is responsible for handling the signup functionality.**
3. **Define the UserService class, which will handle user-related operations.**
4. **Define the NavbarComponent class, which represents the navigation bar component.**
5. **Define the EmployeeCreateComponent class, which is responsible for creating a new employee.**
6. **Define the EmployeeDetailsComponent class, which displays the details of an employee.**
7. **Define the EmployeeEditComponent class, which allows editing employee details.**
8. **Define the EmployeeListComponent class, which displays a list of employees.**
9. **Define the HomeComponent class, which represents the home page component.**
10. **Define the LoginComponent class, which handles the login functionality.**
11. **Define the PasswordChangeComponent class, which allows changing the user's password.**
12. **Create the necessary Angular components with their corresponding HTML templates.**
13. **Implement the signup functionality in the SignupComponent:**
    - ❖ **Import the required dependencies.**
    - ❖ **Define the signupForm FormGroup and errorMessage variables.**
    - ❖ **Initialize the signupForm in the constructor using the FormBuilder.**
    - ❖ **Implement the signup() method, which validates the form data and creates a new user.**

- ❖ **Use the UserService methods to check for duplicate users and store the user data.**

14. Implement the user-related operations in the UserService class:

- ❖ **Define the users array and passwordChangeRequired variable.**
- ❖ **Implement methods like isUniqueUser(), addUser(), isPasswordChangeRequired(), validateUser(), changePassword(), and isEmailExists().**

15. Implement the navigation bar functionality in the NavbarComponent:

- ❖ **Create a navbar with a dark background.**
- ❖ **Include the navbar brand and a collapsible navbar toggle button.**
- ❖ **Add navbar items for home, employees, and logout.**
- ❖ **Add a search form with an input field and a search button.**
- ❖ **Include modals for displaying employee details and employee not found messages.**

16. Implement the employee creation functionality in the EmployeeCreateComponent:

- ❖ **Create a form for creating a new employee.**
- ❖ **Include fields like ID, First Name, Last Name, and Email.**
- ❖ **Add an "Add Employee" button to submit the form.**
- ❖ **Display success and error message divs for feedback.**

17. Implement the employee details display in the EmployeeDetailsComponent:

- ❖ **Display the employee's ID, first name, last name, and email.**
- ❖ **Include buttons for actions like viewing the employee list, deleting the employee, and updating the employee.**

18. Implement the employee editing functionality in the EmployeeEditComponent:

- ❖ **Create a form for updating employee details.**
- ❖ **Include fields like First Name, Last Name, and Email.**
- ❖ **Add an "Update" button to submit the form.**
- ❖ **Include a "Cancel" button to navigate back to the employee details.**

19. Implement the employee list display in the EmployeeListComponent:

- ❖ **Display a table with employee information like ID, First Name, Last Name, and Email.**
- ❖ **Include buttons for viewing employee details and adding a new employee.**

20.Implement the home page display in the HomeComponent:

- ❖ **Display content in a jumbotron or container.**

21.Implement the login functionality in the LoginComponent:

- ❖ **Create a form for user login with fields for username and password.**
- ❖ **Add a "Login" button to submit the form.**
- ❖ **Include links for forgot password and sign up.**

22.Implement the password change functionality in the PasswordChangeComponent:

- ❖ **Create a form for changing the user's password.**
- ❖ **Include fields for the old password, new password, and confirm password.**
- ❖ **Add a "Change Password" button to submit the form.**