



Frontend Documentation

Navigation Bar

The navigation bar located on the top of all pages on Code Next.

```
const Links = () => {
  if (token) {
    // Logged in links
    return (
      <>
        <Link to={'/dashboard'} className='nav-link'>Dashboard</Link>
        <Link to={'/problems'} className="nav-link">Browse</Link>
        <Link to={'/behavioral'} className="nav-link">Behavioral</Link>
        <Link to="/mockinterview" className="nav-link active">Mock Interview</Link>
        <Link to={'/profile'} className="nav-link active">Profile</Link>
        <Link to={'/'} className="nav-link" onClick={() => {removeToken(); onLogout();}}>Logout</Link>
      </>
    )
  } else {
    // Logged out links
    return (
      <>
        <Link to={'/'} className="nav-link active">Home</Link>
        <Link to={'/about'} className="nav-link active">About</Link>
        <Link to={'/login'} className="nav-link active">Login</Link>
        <Link to={'/register'} className="nav-link active">Register</Link>
      </>
    )
  }
}

return (
  <div>
    <nav className="navbar navbar-expand-lg navbar-light ">
      <div className="container">
        <a className="navbar-brand">CodeNext</a>
        <button className="navbar-toggler" type="button" data-bs-toggle="collapse"
          data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup"
          aria-expanded="false" aria-label="Toggle navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse justify-content-end" id="navbarNavAltMarkup">
          <div className="navbar-nav">
            <Links />
          </div>
        </div>
      </div>
    </nav>
  </div>
);
```

CodeNext

[Dashboard](#) [Browse](#) [Behaviorals](#) [Mocks](#) [Stats](#) [Profile](#) [Logout](#)

Leetcode Question Box

The container with the name of the question assigned, and a button to begin the question

If a question is already completed, the button will become disabled

```
<Card className="mt-4 QuestionBackground">
  <Card.Body>
    <Row className="d-flex align-items-center">
      <Col sm={10}>
```

```

<Card.Title className="text-light fw-bold mb-3 text-capitalize">
  <h2>{attempted && '✅ '} <Link className="problem-link" to={` /problems/${id}`}>{name}</Link></h2>
</Card.Title>
<span
  className={`px-3 py-2 rounded-pill ${difficultyTagColor()}`}>
  {difficulty}
</span>

{categories.map((category) => (
  <span
    key={category.id}
    className="px-3 py-2 rounded-pill bg-secondary bg-gradient text-white ms-2"
  >
    {category.name}
  </span>
))}

<div className="mb-2"></div>
</Col>
{viewOnly ? (
  <Col sm={2}>
    <a target="_blank" href={leetcode_url} rel="noreferrer">
      <Button
        className="QuestionButton fw-bold"
        onClick={handleClick}>
        Attempt
      </Button>
    </a>
  </Col>) : (
  <Col sm={2}>
    <Link className="problem-link" to={` /problems/${id}`}>
      <Button
        className="QuestionButton fw-bold"
      >
        Code Next
      </Button>
    </Link>
  </Col>
)
}
</Row>
</Card.Body>
</Card>

```

Single Number

EASY

arrays

Code Next

Main Dashboard

The dashboard containing all the Leetcode questions assigned for the day. Also includes a user greeting and a navigation bar

```

<div>
  <Container className="pt-5 pb-5">
    {quizCompleted ? (
      <div>
        <h1 className="">Welcome back! 🌞</h1>
        <p className="fs-3">Our data shows that your weakest data structure is <strong>{categoryName}</strong>,
          so we recommend the following {categoryName} problems to you.
        </p>
        <Problems />
      </div>
    ) : (
      <div>
        <h1 className="">Welcome to CodeNext!</h1>
        <h3>To get started, you are required to complete the <Link className="text-success" to="/skill-assessment">skill as
        </div>
      </div>
    )
  </Container>
</div>

```

Welcome back! 🙌

Our data shows that your weakest data structure is **trees**, so we recommend the following trees problems to you.

Binary Tree Maximum Path Sum

HARD

trees

[Code Next](#)

Serialize And Deserialize Binary Tree

HARD

trees

[Code Next](#)

Landing Page

The landing page where user visits the base url of Code Next. This page displays general information about the site.

```
const cardInfo = [
  {title: "Tailored Plan", text:
    <ul style={{ "font-size": "1rem", "line-height": "2rem"}} className="mx-auto">
      <li>Assessment to dictate concepts you need a better understanding.</li>
      <li>Personalized road map following the assessment.</li>
      <li>No need to wonder where to start.</li>
    </ul>
  },
  {title: "Distributed Practice", text:
    <ul style={{ "font-size": "1rem", "line-height": "2rem"}} className="mx-auto">
      <li>Precision practice of selected questions.</li>
      <li>Distributed learning strategy to enhance your understanding of the concepts.</li>
      <li>Better Retention.</li>
    </ul>
  },
  {title: "Track Your Progress", text:
    <ul style={{ "font-size": "1rem", "line-height": "2rem"}} className="mx-auto">
      <li>Manage and track your progress as you use our platform.</li>
      <li>See a visualization of your progress.</li>
      <li>Record key metrics like time taken and perceived difficulty.</li>
    </ul>
  },
];

const renderCard = (card, index) => {
  return (
    <Card style={{ width: '18rem', height: '20rem', margin: 'auto'}} key={index} className="box">
      <Card.Body>
        <Card.Title>{card.title}</Card.Title>
        <Card.Text className="mx-auto">
          {card.text}
        </Card.Text>
      </Card.Body>
    </Card>
  );
};

return(
  <div className="grid d-flex align-items-center">
    <h1 style={{textAlign: "center", "font-size": '48px'}}>CODE NEXT</h1>
    <h3 style={{textAlign: "center"}}>EVERYTHING YOU NEED TO EXCEL IN YOUR NEXT SOFTWARE INTERVIEW
      IS NOW MADE AVAILABLE THROUGH CODE-NEXT.
    </h3>

    {cardInfo.map(renderCard)}
  </div>
);
```

```

<div className="mx-auto">
  <Link to="/register" style={{'padding-right': '8rem'}}>
    <Button variant="info" className="align-middle">Register Now!</Button>
  </Link>

  <Link to="/login">
    <Button variant="info" className="align-middle">Sign in!</Button>
  </Link>
</div>

</div>

)

```

CodeNext

[Home](#) [About](#) [Login](#) [Register](#)

Code Next

Everything you need to excel in your next software interview is now made available through code-next.

Tailored Plan

- Assessment to dictate concepts you need a better understanding.
- Personalized road map following the assessment.
- No need to wonder where to start.

Distributed Practice

- Precision practice of selected questions.
- Distributed learning strategy to enhance your understanding of the concepts.
- Better Retention.

Track Your Progress

- Manage and track your progress as you use our platform.
- See a visualization of your progress.
- Record key metrics like time taken and perceived difficulty.

Register Now!

Login Page

The login page allows previous users to login to their account.

```

<div className="auth-form-container">
  <h2>Log in</h2>

  <form className="login-form" onSubmit={handleSubmit}>
    <label htmlFor="username">username</label>

    <input value={username} onChange={(e) => setUsername(e.target.value)}
      type="text"
      placeholder="enter your username here"
      id="email"
      name="email">
    </input>

    <label htmlFor="password">password</label>
    <input value={password} onChange={(e) => setPassword(e.target.value)}
      type="password"
      placeholder="enter your password here"
      id="password"
      name="password">
    </input>

    <button
      id="login-button"
      type="button"
      onClick={handleSubmit}>log in</button>
  </form>

```

```

    <Link to="/Register" className="link-button" onClick={() =>props.onFormSwitch('register')}}>
      don't have an account? register here.
    </Link>

  </div>

```

CodeNext

[About](#) [Login](#) [Register](#)

Log in

Username

Password

Log in

Register Page

The register page allows new users to create an account with Code Next

```

<div className="auth-form-container">
  <h2>Register</h2>

  <form className="register-form" onSubmit={handleSubmit}>

    <label>username</label>
    <input value={username} onChange={(e) => setUsername(e.target.value)}
      type="name"
      name="name"
      id="name"
      placeholder="enter your username here">
    </input>

    <label htmlFor="email">email</label>
    <input value={email} onChange={(e) => setEmail(e.target.value)}
      type="email"
      placeholder="enter your email here"
      id="email"
      name="email">
    </input>

    <label htmlFor="password">password</label>

    <input value={pass} onChange={(e) => setPass(e.target.value)}
      type="password"
      placeholder="enter your password here"
      id="password"
      name="password">
    </input>
    <button id="register-button">register</button>
  </form>

  <Link to="/login" className="link-button" onClick={() =>props.onFormSwitch('login')}}>
    already have an account? login here.
  </Link>
</div>

```

Register

Username

Email

Password

register

Pre-Assessment Multiple Choice Page

The Pre Quiz page allows users to answer questions in order for Code Now to understand their progress and skill level as coders.

```

<NavBar/>
<div style={{ display: "flex", justifyContent: "center", alignItems: "center" }}>
  <FontAwesomeIcon icon={faClipboardCheck} size="10x" />
</div>
<br><br>
<Container className="d-flex justify-content-center align-items-center flex-column pt-5 pb-5" style={{ backgroundColor: '#F5F5F5', padding: '10px' }}>
  <h1 className="display-1 text-center mb-4">Welcome to the Pre-Quiz! </h1>
  <h3 className="text-center mb-4">We will use this quiz to identify which problems to assign to you!</h3>
  { /* <Button variant="primary" onClick={handleStart}>Start</Button> */ }
  { !isStarted && (
    <div className="start-button-container" style={{ display: "grid", justifyContent: "center", alignItems: "center" }}>
      <p className="text-center mb-4">To begin the quiz, simply click on the Start button. Please answer these questions to the best of your ability.</p>
      <Button
        variant="primary"
        onClick={() => {
          handleStart();
          setShowModal(true);
        }}
        className={submitted ? "hidden" : ""}
      />
      Start Prequiz
    </div>
  ) }
</Container>

```



Welcome to the Pre-Assessment!

We will use this assessment to identify which problems to assign to you!

To begin the assessment, simply click on the Start button. Please answer these questions to the best of your ability.

Start Assessment

Pre-Assessment Modal Popups

The modals containing the questions and a slider to give the experience level

```
const renderModalTitle = () => {
  switch (currentQuestion) {
    case 1:
      return 'Question 1';
    case 2:
      return 'Question 2';
    case 3:
      return 'Question 3';
    case 4:
      return 'Question 4';
    case 5:
      return 'Question 5';
    default:
      return '';
    case 6:
      return 'WOOHOO! >.<';
  }
};

const renderModalBody = () => {
  if (submitted) {
    return (
      <div>
        <p>{modalText}</p>
      </div>
    );
  }
  switch (currentQuestion) {
    case 1:
      return (
        <>
          <p>How well do you know Arrays?</p>
          <div className="slider-container">
            <span className="slider-label">No experience </span>
            <input type="range" min="0" max="4" value={sliderValues[0]} className="slider" id={"stack-knowledge-slider-" + currentQuestion}>
            <span className="slider-label"> Experienced</span>
          </div>
        </>
      );
    case 2:
      return (
        <>
          <p>How well do you know Linked Lists?</p>
          <div className="slider-container1">
            <span className="slider-label">No experience </span>
            <input type="range" min="0" max="4" value={sliderValues[1]} className="slider" id={"stack-knowledge-slider-" + currentQuestion}>
            <span className="slider-label"> Experienced</span>
          </div>
        </>
      );
    case 3:
      return (
        <>
          <p>How well do you know Stacks?</p>
          <div className="slider-container2">
            <span className="slider-label">No experience </span>
            <input type="range" min="0" max="4" value={sliderValues[2]} className="slider" id={"stack-knowledge-slider-" + currentQuestion}>
            <span className="slider-label"> Experienced</span>
          </div>
        </>
      );
    case 4:
      return (
        <>
          <p>How well do you know Priority Queues?</p>
          <div className="slider-container3">
            <span className="slider-label">No experience </span>
            <input type="range" min="0" max="4" value={sliderValues[3]} className="slider" id={"stack-knowledge-slider-" + currentQuestion}>
            <span className="slider-label"> Experienced</span>
          </div>
        </>
      );
    case 5:
      return (
        <>
          <p>How well do you know Trees?</p>
          <div className="slider-container4">
```

```


        <span className="slider-label">No experience </span>
        <input type="range" min="0" max="4" value={sliderValues[4]} className="slider" id={"stack-knowledge-slider-" + currentQuestionId} />
        <span className="slider-label"> Experienced</span>
      </div>
    </>
  );
};
case 6:
  return (
    <div>
      <p>You have completed the quiz!</p>
      <Button onClick={() => setShowModal(false)}>Close</Button>
    </div>
  );
};

default:
  return '';
}
};

```

Question 1

How comfortable are you solving Leetcode problems with the **arrays** data structure?

Uncomfortable

Very comfortable

You are a skilled warrior at solving problems arrays problems! 🙌

Save and Continue

Post Question Survey

The survey that users will complete in order for our site to evaluate your skill level

```

<Card>
  <section className='Main'>
    <div>
      <span className='home-page'></span>
    </div>
    <h1>Practice the Question!</h1>
    <p>
      Get ready to tackle some mind-bending questions and unleash your coding potential!
      Before you start, here are a few things you should know:
    </p>
    <ul>
      <li>Read the instructions and problem statement carefully for the question.</li>
      <li>You can use any programming language you are comfortable with to solve the question.</li>
      <li>There is no time limit, so take your time and think through your solution</li>
      <li>Once you have completed a question, come back to this page and stop the timer! You will be prompted with a feedback</li>
    </ul>
  </section>
</Card>
{active === "SecondCard" &&
  <Card>
    <section className='Question'>
      <h1>Are you done?</h1>
      <p>
        If you have successfully attempted the given problem,
        It is now time to go over a few questions to track your progress!
      </p>
    </section>
  </Card>
}

```



```

</section>
<>

<Form className='question-input'>
  <form>
    <div className='form'>
      <label className='Q1'>
        Did you manage to complete this question?
        <select value={selectedOption} onChange={handleChange} required>
          <option value=""> --- </option>
          <option value="option1">Yes</option>
          <option value="option2">No</option>
        </select>
      </label>
      <br></br>
      <label className='Q2'>
        How long did you take to complete the question?
        <input type="text" id="text" name="text" placeholder='ex. 10 mins' required/>
      </label>
      <br></br>
      <label className='Q3'>
        How difficult was the question from the range 1-5?
        <label htmlFor="customRange2" className="form-label"></label>
        <input type="range" className="form-range" min="0" max="5" id="customRange2" required/>
      </label>
      <br></br>
      <button>Done</button>
    </div>
  </form>
</Form>
</>
</Card>

```

Are you done?

If you have successfully attempted the given problem, It is now time to go over a few questions to track your progress!

Did you manage to complete this question?

Select

How long did you take to complete the question? (Minutes)

10

How difficult was this question for you?

Easy Hard

Done

Forgetting Curve Information Card

A card to explain to the user what the Forgetting curve is, and how the website will combat this.

```

<div>
  <Navbar />
  <div className="container d-flex justify-content-center mt-5">
    <Card bg="light">
      <Card.Body>
        <Card.Title className="text-center" style={{ fontSize: '1.4rem', color: 'black'}}>What is Spaced Repetition?</Card.Title>
        <Card.Text className="text-center" style={{ fontSize: '1.2rem', color: 'black' }}>
          Spaced repetition is a learning technique that involves reviewing
          material at increasing intervals of time. By incorporating spaced
          repetition into CodeNext, we aim to help our users retain information
          and develop a deeper understanding of the material. By practicing
          questions multiple times, users will be less likely to forget the information
          they have learned and will avoid the Forgetting Curve phenomenon. This will ultimately lead to improved performance
          and increased confidence in their abilities. Our commitment to incorporating
          this technique into CodeNext highlights our dedication to providing our
          users with a high-quality learning experience that is both effective and efficient.
        </Card.Text>
        <Card.Title className="text-center" style={{ fontSize: '1.4rem', color: 'black'}}>How does CodeNext utilize Spaced Repetition?</Card.Title>
        <Card.Text className="text-center" style={{ fontSize: '1.2rem', color: 'black' }}>
          CodeNext utilizes the spaced repetition strategy by sending notifications to users,
          asking them to repeat the coding questions they have previously attempted.
        </Card.Text>
      </Card.Body>
    </Card>
  </div>
</div>

```

```

        This helps the user to retain knowledge, improve their understanding of coding concepts,
        and ultimately become better coders. By incorporating spaced repetition into our platform,
        we are committed to providing the best possible learning experience for our users.
      </Card.Text>
    </Card.Body>
  </Card>
</div>
<div className="d-flex justify-content-center mt-5">
  <h5>So what are you waiting for?</h5>
</div>
<div className="d-flex justify-content-center">
  <Link to="/register">
    <button className="btn btn-primary pe-4 ps-4"> Register Now! </button>
  </Link>
</div>
</div>
</div>

```

What is Spaced Repetition?

Spaced repetition is a learning technique that involves reviewing material at increasing intervals of time. By incorporating spaced repetition into CodeNext, we aim to help our users retain information and develop a deeper understanding of the material. By practicing questions multiple times, users will be less likely to forget the information they have learned and will avoid the Forgetting Curve phenomenon. This will ultimately lead to improved performance and increased confidence in their abilities. Our commitment to incorporating this technique into CodeNext highlights our dedication to providing our users with a high-quality learning experience that is both effective and efficient.

How does CodeNext utilize Spaced Repetition?

CodeNext utilizes the spaced repetition strategy by sending notifications to users, asking them to repeat the coding questions they have previously attempted. This helps the user to retain knowledge, improve their understanding of coding concepts, and ultimately become better coders. By incorporating spaced repetition into our platform, we are committed to providing the best possible learning experience for our users.

So what are you waiting for?

Register Now!

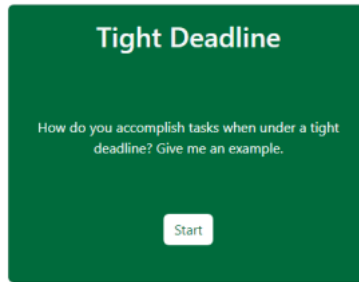
Behavioural Question Card

The container with the name and description of the behavioral question assigned, and a button to begin the question.

```

<Col>
  <Card style={{ height: '20rem', width: '26rem'}} className="mt-4 QuestionBackground">
    <Card.Body className="align-items-center">
      <Row style={{ height: '4rem'}} className="text-center mb-5">
        <Card.Title className="text-light fw-bold"><h2>{props.name}</h2></Card.Title>
      </Row>
      <Row style={{ height: '7rem'}} className="text-center ">
        <Card.Text style={{ color: 'white'}} className="text-center">
          {props.desc}
        </Card.Text>
      </Row>
      <Row style={{ height: '1rem'}}>
        <Card.Text className="text-center">
          <Button className="QuestionButton" onClick={handleClick}>Start</Button>
        </Card.Text>
      </Row>
    </Card.Body>
  </Card>
</Col>

```



Behavioural Question Page

The Behavioral Question allows users to access behavioral questions that are often asked in interviews.

```
<div>
  <NavBar/>

  <Container className="pt-5 pb-5">

    <h1 className="display-1">Welcome to Behavioral Questions! </h1>
    <h3>Here you will be given common behavioral questions asked during interviews. </h3>
    <p>These are slightly different from questions on your dashboard.
      To answer these questions, think of your past experiences and what you have learned.</p>

    <p>Remember to use the STAR method</p>
    <ul>
      <li><strong>Situation:</strong> Set the scene and give the necessary details of your example</li>
      <li><strong>Task:</strong> Describe what your responsibility was in that situation</li>
      <li><strong>Action:</strong> Explain exactly what steps you took to address it</li>
      <li><strong>Result:</strong> Share what outcomes your actions achieved</li>
    </ul>
    { /* Taken from https://www.themuse.com/advice/star-interview-method */ }

    { /* <div className="row mt-1"> */ }
    <QueryClientProvider client={ queryClient }>
      <GetBehaProbs />
    </QueryClientProvider>
    { /* </div> */ }
  </Container>
</div>
```

Welcome to Behavioral Questions!

Here you will be given common behavioral questions asked during interviews.

These are slightly different from questions on your dashboard. To answer these questions, think of your past experiences and what you have learned.

Remember to use the STAR method

- **Situation:** Set the scene and give the necessary details of your example
- **Task:** Describe what your responsibility was in that situation
- **Action:** Explain exactly what steps you took to address it
- **Result:** Share what outcomes your actions achieved

Recent Projects

Tell me about the most recent project you worked on.
What were your responsibilities?

Start

Programming Languages

What languages have you programmed in?

Start

Improving Design

Describe a time you were able to improve upon the
design that was originally suggested.

Start

Proud Projects

Going Above & Beyond

Managing Code

Behavioral Question Form

The form allows users to answer behavioral questions that are often asked in interviews.

```
<div>

  <div className="container mt-4">

    <section className='Question'>
      <h1 className="display-1">{location.state.behavior_name}</h1>
      <h3> {location.state.behavior_ques} </h3>
      <p>Note that these textfields are here for you to get your thoughts down, but our site will not record your responses!</p>
      <p>This is for you to reflect and organize your thoughts for the real interview!</p>
    </section>

    <div className='form'>

      <div className="mb-3">
        <Form.Label><h6>Situation</h6></Form.Label>
        <Form.Control as="textarea" rows={3}/>
      </div>

      <div className="mb-3">
        <Form.Label><h6>Task</h6></Form.Label>
        <Form.Control as="textarea" rows={3}/>
      </div>

      <div className="mb-3">
        <Form.Label><h6>Action</h6></Form.Label>
        <Form.Control as="textarea" rows={3}/>
      </div>

      <div className="mb-3">
        <Form.Label><h6>Result</h6></Form.Label>
        <Form.Control as="textarea" rows={3}/>
      </div>
    </div>
  </div>
```

```

<div className="mb-3">
  <a className="nav-link active" aria-current="page" href="/behavioral"><button className="btn btn-primary mb-4">Done</button></a>
</div>

</div>

</div>

```

Tight Deadline

How do you accomplish tasks when under a tight deadline? Give me an example.

Note that these textfields are here for you to get your thoughts down, but our site will not record your responses!

This is for you to reflect and organize your thoughts for the real interview!

Situation

Task

Action

Result

Done

User Profile Page

The page allows the user to view their profile, along with their account stats in terms of their experience with a topic

```

<div>
  <Container className="pt-3 pb-5">
    <h1>Profile Information</h1>

    <Card className="mb-4 mb-lg-0 mt-3">
      <Card.Body>
        <Row>
          <Col sm="5">
            <Card.Text>First Name</Card.Text>
          </Col>
          <Col sm="7">
            <Card.Text className="text-muted">{user.first_name ? user.first_name : 'No first name found'}</Card.Text>
          </Col>
        </Row>
        <hr />
        <Row>
          <Col sm="5">
            <Card.Text>Last Name</Card.Text>
          </Col>
          <Col sm="7">
            <Card.Text className="text-muted">{user.last_name ? user.last_name : 'No last name found'}</Card.Text>
          </Col>
        </Row>
        <hr />
        <Row>
          <Col sm="5">

```

```

        <Card.Text>Email</Card.Text>
      </Col>
      <Col sm="7">
        <Card.Text className="text-muted">{user.email ? user.email : 'No email found'}</Card.Text>
      </Col>
    </Row>
    <hr />
    <Row>
      <Col sm="5">
        <Card.Text>Username</Card.Text>
      </Col>
      <Col sm="7">
        <Card.Text className="text-muted">{user.username}</Card.Text>
      </Col>
    </Row>
  </Card.Body>
</Card>

<br />
</Container>
</div>

```

CodeNext

[Dashboard](#) [Browse](#) [Behaviorals](#) [Mocks](#) [Stats](#) [Profile](#) [Logout](#)

Profile Information

First Name	John
Last Name	Smith
Email	a@gmail.com
Username	a

Edit Profile Form

The form allows the user to edit their current profile

```

<Accordion>
  <Accordion.Item eventKey="1">
    <Accordion.Header>Change Name</Accordion.Header>
    <Accordion.Body>
      <Form>
        <Form.Group className="mb-3" id="formBasicFirstName">
          <Form.Label>First Name</Form.Label>
          <Form.Control
            type="text"
            placeholder="Enter first name"
          />
          {errors.firstName && (
            <Form.Text className="text-danger">
              {errors.firstName}
            </Form.Text>
          )}
        </Form.Group>

        <Form.Group className="mb-3" id="formBasicLastName">
          <Form.Label>Last Name</Form.Label>
          <Form.Control
            type="text"
            placeholder="Enter last name"
          />
          {errors.lastName && (
            <Form.Text className="text-danger">
              {errors.lastName}
            </Form.Text>
          )}
        </Form.Group>

        <Button

```

```

        variant="success"
        type="submit"
        onClick={(e) => handleNameChange(e)}
      >
        Submit
      </Button>
    </Form>
  </Accordion.Body>
</Accordion.Item>
</Accordion>

<br/>

<Accordion>
  <Accordion.Item eventKey="2">
    <Accordion.Header>Change Email</Accordion.Header>
    <Accordion.Body>
      <Form>
        <Form.Group className="mb-3" controlId="formBasicEmail">
          <Form.Label>Email address</Form.Label>
          <Form.Control type="email" placeholder="Enter new email" />
          {errors.email && <p className="text-danger">{errors.email}</p>}
        </Form.Group>

        <Button variant="success" type="submit" onClick={(e) => handleEmailChange(e)}>
          Submit
        </Button>
      </Form>
    </Accordion.Body>
  </Accordion.Item>
</Accordion>

<br/>

<Accordion>
  <Accordion.Item eventKey="3">
    <Accordion.Header>Change Password</Accordion.Header>
    <Accordion.Body>
      <Form>
        <Form.Group className="mb-3" controlId="formBasicPassword1">
          <Form.Label>Password</Form.Label>
          <Form.Control type="password" placeholder="Enter new password" />
          {errors.password && <p className="text-danger">{errors.password}</p>}
        </Form.Group>

        <Form.Group className="mb-3" controlId="formBasicPassword2">
          <Form.Label>Retype Password</Form.Label>
          <Form.Control type="password" placeholder="Retype new password" />
          {errors.confirmPassword && (
            <p className="text-danger">{errors.confirmPassword}</p>
          )}
        </Form.Group>

        {successMessage && (
          <div className="alert alert-success mt-3" role="alert">
            {successMessage}
          </div>
        )}

        <Button variant="success" type="submit" onClick={(e) => handlePasswordChange(e)}>
          Submit
        </Button>

      </Form>
    </Accordion.Body>
  </Accordion.Item>
</Accordion>

```

Change Name

First Name

John

Last Name

Smith

Submit

Change Email

Email address

Enter new email

Submit

Change Password

Password

Enter new password

Retype Password

Retype new password

Submit

Mock Interview Page

This page provides users with a mix of behavioral and leet code questions to simulate a real interview

```
<Container className="pt-5 pb-4">
  <Row>
    <Col md={{ span: 6, offset: 3 }}>
      <h1 className="display-1 text-center">Mock Interview</h1>
      <h5 className="text-center mb-4">Practice for your job interview with our expertly crafted questions!</h5>
    </Col>
  </Row>
  <Row>
    <Col md={{ span: 8, offset: 2 }}>
      <div className="card mb-4">
        <div className="card-body">
          <h5 className="card-title text-center mb-0" style={{ color: '#5A5A5A' }}>Please select the level of difficulty for your mock
        </div>
      </div>
    </Col>
  </Row>
  <Row>
    {cardItems}
  </Row>
</Container>
```


Mock Interview

Practice for your job interview with our expertly crafted questions!

Please select the level of difficulty for your mock interview.

Easy

Step into coding with our easy mock interview! Tackle 2 LeetCode-style coding Qs & 1 behavioral Q to test communication skills. Gain confidence before the real deal!

Start the interview

Medium

Take your coding skills to the next level! Our medium mock interview features 2 challenging LeetCode-style coding Qs, 1 behavioral and 1 technical question to refine your skills. Don't miss out!

Start the interview

Hard

Ready for a challenge? Our hard mock interview puts your skills to the test! 2 medium and 1 hard LeetCode-style coding Qs, plus 2 technical questions that test communication & problem-solving. Showcase your expertise!

Start the interview

User Stats Page

This page lets the user know some personal statics on their personal performance

```
<div>
  <Container className="pt-5 pb-5"> {/* cards for highest skill, total time coding, average time coding, etc */}
    <h1 className="display-1">Statistics</h1>
    <h3>This page contains your statistics, progress in each skill, and achievements.</h3>
    <div className="row mt-1 mb-1">
      <Row style={{height: '25rem'}} className="text-center mb-5">
        <Col sm={4}>
          <Spotlight
            name={"Highest Skill"}
            desc={"Your highest skill is: "} // get highest skill from a user query
            stat={getMax(s)[0]} // fix this to say the name
          />
        </Col>
        <Col sm={4}>
          <Spotlight
            name={"Highest Proficiency"}
            desc={"The proficiency of your highest skill is: "} // get time spent from an attempt query
            stat={getMax(s)[1]}
          />
        </Col>
        <Col sm={4}>
          <Spotlight
            name={"Number of Skills Maxed Out"}
            desc={"The number of skills you've maxxed out is: "}
            stat={getCount(s)}
          />
        </Col>
      </Row>
    </div>
  </Container>
</div>
```

Statistics

This page contains your statistics, progress in each skill, and achievements.



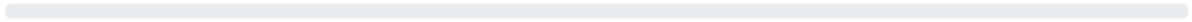
User Progress Bars

These bars let the user know how far they have come in a particular topic

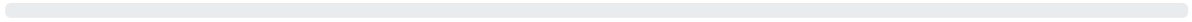
```
<div className="row mt-1 mb-1">
  <h3>Skill Progress</h3>
  <div className="row mt-1">
    <ul>
      {s.map((skill) => (
        <Progress
          key={skill[0]}
          name={skill[0]}
          value={skill[1]}
        />
      ))}
    </ul>
  </div>
</div>
<div className="row mt-1 mb-1">
  <h3>Achievements</h3>
  <div className="row mt-1">
    <ul>
      {achievements.map(achievement => (
        <Achievements
          key={achievement["name"]}
          name={achievement["name"]}
          // desc={achievement["user"]} // for now
          // img={achievement.image}
        />
      ))}
    </ul>
  </div>
</div>
```

Skill Progress

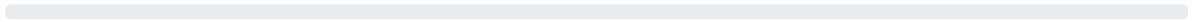
Arrays



Trees



Linked Lists



Achievements

First!