Build a Modern Web Application using ReactJS and GraphQL

Anand Sudhanaboina
Mohan Kumar
Rachel Susan
Shashank Keshava
Vaishakh K

14 Feb 2020
NSUT Delhi
WHO ARE WE?

We are Intuit
A company conceived 35 years ago at our co-founder’s kitchen table to help small businesses and individual customers grow, eliminate work and give them complete confidence.
Powering Prosperity Around the World
OUR JOURNEY
SO FAR

1980s
1990s
2000s
2010s
2020s

ERA OF DOS
Intuit Founded
Quicken DOS

ERA OF WINDOWS
Quicken Windows
TurboTax Windows
QuickBooks Windows

ERA OF WEB
Began services in India
Mint launched out of India

ERA OF MOBILE & CLOUD
QuickBooks Mobile
Tax Knowledge Engine

ERA OF AI, CHAT & VR
Launched Money Manager Exclusively for Indian market
QuickBooks & TurboTax Self-Employed

CUSTOMER-OBSESSED • DESIGN-INSPIRED • TECHNOLOGY-POWERED
PRODUCTS THAT POWER PROSPERITY

Our technology has helped us innovate four of our major products that are simplifying work of millions, worth millions.

50M CUSTOMERS

$6B COMPANY
Agenda

● What are we building?
● Prerequisite Learnings
● Platform Setup
● Hands-on Session
● Recap
● Quiz + Wrap Up :)
What are we building?

Build a microblogging web application using ReactJS, GraphQL and NodeJS
Outcomes

- Hands-on experience in building a modern web application
- Familiarity with ReactJS, GraphQL and NodeJS
- Understand industry best practices
Full Stack Engineering

Developer Type

<table>
<thead>
<tr>
<th>Developer Type</th>
<th>All Respondents</th>
<th>United States Unweighted</th>
<th>United States Weighted by Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer, full-stack</td>
<td>51.9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer, back-end</td>
<td>50.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer, front-end</td>
<td>32.8%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer, desktop or enterprise applications</td>
<td>21.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data analysis or machine learning specialist</td>
<td>7.9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer, QA or test</td>
<td>7.8%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data or business analyst</td>
<td>7.7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Academic researcher</td>
<td>7.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering, data</td>
<td>7.2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Educator</td>
<td>6.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer, games or graphics</td>
<td>5.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering manager</td>
<td>5.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product manager</td>
<td>5.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scientist</td>
<td>4.4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineer, site reliability</td>
<td>3.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Senior executive/VP</td>
<td>2.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing or sales professional</td>
<td>1.2%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Source: Insights stackengineer.com/survey/0164/developer-profile-developer-type-all-respondents*
Tech We’ll Use Today

- Node.js
- React
- GraphQL
- SQLite
System Architecture
Tasks for Today

• Build a Read List UI Component
• GraphQL backend integration
• Build Post Component
• Backend integration to Post Component
• Update the GraphQL schema
Prerequisite Learnings
Web Concepts

- HTML
- CSS
- Javascript
Web Concepts

- HTML
- CSS
- Javascript
Web Concepts

- HTML
- CSS
- Javascript
Web Concepts
Traditional Architecture
Why React?

- Declarative
- Component-Based
- Learn once write anywhere
React Lifecycle

**Mounting**
- `constructor`
- `getDerivedStateFromProps`
- `shouldComponentUpdate`
- `render`
- `componentDidMount`

**Updating**
- `New props`
- `setState()`
- `forceUpdate()`
- `getSnapshotBeforeUpdate`
- `React updates DOM and refs`
- `componentDidUpdate`

**Unmounting**
- `componentWillUnmount`
Virtual DOM (VDOM)

Hello, world!

It is 12:26:46 PM.

React Only Updates What’s Necessary
Components

- Independent
- Reusable
- Isolation
Components

Function Components

```javascript
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Class Components

```javascript
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```
Composing Components aka Container

```javascript
function App() {
  return (
    <div>
      <Welcome name="Sara" />
      <Welcome name="Cahal" />
      <Welcome name="Edite" />
    </div>
  );
}
```
Quiz Time 🥳
What is the traditional frontend architecture?
What is responsible for dynamic rendering in React?
A Component is Independent
Isolate
AND?
GraphQL
https://graphql.org

Why GraphQL?

- Describe your data
- Ask for what you want
- Get predictable result
Apollo Client

https://www.apollographql.com/docs/react/

What is Apollo Client?

- State management library for JavaScript apps.
- Write a GraphQL query, and Apollo Client will take care of requesting and caching your data, as well as updating your UI.
Setup

Pre-requisites

- Node.JS. Recommend version: 12.14.1 LTS.
- Git
- Type node --version in the command line (cmd), this should give you the version of node that's installed.
### Batch Transactions

<table>
<thead>
<tr>
<th>Invoice No.</th>
<th>Customer</th>
<th>Email</th>
<th>Terms</th>
<th>Invoice Date</th>
<th>Due Date</th>
<th>Product/Service</th>
<th>Description</th>
<th>Qty</th>
<th>Rate</th>
<th>Amount</th>
<th>Send Late</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Net 30</td>
<td>02/11/2020</td>
<td>03/12/2020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Yes! Built using the same technology stack**
Take Home Challenges

- Add title
- Increase char limit
Your opportunity to ask and learn