

# Berkley Olmstead

 [LinkedIn](#) |  208-871-3882 |  [berkleyolmstead.com](http://berkleyolmstead.com) |  olms2074@gmail.com |  [GitHub](#)

## Summary

Devoted Full-Stack Developer with 4 years experience building web applications in Typescript, React, Node, Python, SQL, MongoDB, and others. Professional experience as a Salesforce Developer and strong history as a CPA candidate in Tax and Accounting.

## Skills

- JavaScript | TypeScript | Node | Express | React | Redux | Python | Flask | AWS | CSS | Material UI | MongoDB | MySQL | Postgres | Git
- Salesforce Development | LWCs | Aura Components | Apex | Flow Builder | Apex Testing | SOQL | SOSL | DB Schema Creation
- Full-Stack Development | REST API Development and Integrations | CI/CD Pipeline Management | Cryptocurrency/NFT Development

## Experience

### Full-Stack and Salesforce Developer

[TaxRise](#)

Remote 08/2023 - 07/2024

- Built several React components, Node/Python REST API routes, and Postgres/MySQL Database interfaces that support web applications currently used in production by TaxRise clients and employees.
- Taught myself the Salesforce Platform and contributed to our Salesforce codebase by creating LWC's, Aura Components, Objects along with their proper relationships, Apex Classes, Flows, and strong covered Apex tests.
- Created products such as: a Full-Stack SERN application that enables TaxRise Tax Associates to view/add/edit/delete financial records uploaded by our clients, Python backend code and React UI that enables TaxRise staff to create questions to dynamically appear on clients' tax questionnaires, and a Salesforce LWC UI and Apex code to enable clients to add/edit/delete their claimed tax assets.
- Practiced strong coding practices and version control such as: Continuous Integration/Deployment Pipeline Integration, pull requests, code reviews, load/stress testing, unit/integration/e2e testing.

### Full-Stack Software Developer

[TEKware](#)

Remote 08/2021 - 07/2023

- Created several React components, backend REST API logic, and MongoDB database architecture while managing the production databases and deployments for a Motor Carrier Safety Compliance SaaS website called SafetyAuditprep.net.
- Created UI and necessary backend functionality on the site for the following: Promotional Codes, Homepage Dashboard, Password Recovery, Task Assignments, Tractor/Trailer Assignments, Register User and Payment Validation, Automated Invoice Creation with Cron Jobs, Drug Test Requests, and others.
- Collaborated with our client and Product Manager to build their requested features and clarify priorities or design ambiguities.
- Integrated several external REST and SOAP APIs into our site which include the following: Quest Diagnostics, TazWorks, FMCSA, Google Maps, and Authorize.NET.

### Tax Auditor

[Idaho State Tax Commission](#)

Boise, ID, USA 02/2016 - 03/2020

- Calculated state individual tax deficiencies and prepared audit findings defending my position.
- Handled over 1,200 audits, increasing the state's income by \$100,000 per year on average.
- Met with taxpayers to communicate tax code ambiguities guiding them to an appropriate plan of action.

## Education

### Software Engineer Program

[Springboard](#)

Remote 08/2020 - 4/2021

- 800+ hours of course material, 1:1 industry expert mentor oversight, and 4 in-depth portfolio projects.
- Refined skills in front-end and back-end web development, databases, and data structures & algorithms.

### Bachelor of Science

[University of Idaho](#)

Moscow, ID, USA 08/2008 - 12/2012

- Accounting (received Mar. 2012) and Finance (received Dec. 2012) Degrees with a 3.5 cumulative GPA.

**Certified Public Accounting Exam** - passed all 4 sections of exam.

## Projects

- **Safety Audit Prep**: Link to an interactive portfolio displaying some of the Full-Stack components I built (MERN stack): [Portfolio](#)
- **FreeBay**: Mock eBay website that allows users to bid on fake products with fake money (SERN stack): [FreeBay Site](#)
- **The Crypto Garden**: Discord group that provides investing information with bot feeds I created with APIs and scraper code: [Discord](#)