# Daniel J. Berg

https://www.danielbergport.org
U.S. Citizen • djberg6@asu.edu/danielberg313@gmail.com • (602)-715-8964 • linkedin.com/in/dj-berg • github.com/dj-berg

## Technical Skills

- **Languages:** Python, JavaScript, HTML, CSS, Java, C/C++
- **Technologies:** React.js, Node.js, Flask, SQL (SQLite), MongoDB, REST APIs, Pandas, NumPy, scikit-learn, Chart.js, AWS (EC2 concepts), Render, Vercel
- **Tools:** Git/GitHub, VS Code
- **Knowledge:** OOP, DSA, Database Design & SQL, Cloud Computing Concepts, Deployment Fundamentals, AI/ML Integration, Debugging & Testing

## Experience

**Full-Stack Task Manager Web Application (In Progress) | React, Node, JavaScript**

- Developing a full-stack task management web application using **React.js** and a **Node.js** backend, enabling users to **create, update, and organize tasks**, supported by a **MongoDB** data model with **5+ task fields** and full **CRUD** functionality.

**Full-Stack Expense Tracker Web Application | Python, Flask, SQL**

- Implemented a full-stack expense tracking application using **Python** and **Flask**, supporting **100% authenticated user access** with session-based login.
- Designed a 2-table **SQL** schema and aggregation queries to compute **weekly spending totals** and **category breakdowns**, reducing manual tracking effort by **~80%**.
- Built a responsive frontend with **HTML, CSS,** and **JavaScript**, visualizing **100%** of **user expenses** in interactive **pie charts** for faster financial insight.

**Front-End Portfolio Web Application | JavaScript, HTML, CSS**

- Designed and developed a fully responsive personal portfolio website to showcase projects and technical skills across desktop and mobile devices
- Achieved a **90+ Lighthouse performance score** through asset optimization and layout stability

## Education

**Arizona State University** **Tempe, Arizona**
*Bachelor's of Science in Computer Science (Software Engineering)* *Expected May 2027*

## Activities

**ACM at ASU** *Aug 2024 - Present*
*Member*

- Solved **50+ LeetCode-style data structures and algorithms problems** through weekly practice sessions and competitive programming exercises
- **Collaborated** in small teams (3–5 members) to **analyze and solve algorithmic challenges**, improving problem-solving speed and team communication