Amirreza Shams

B.S. Mathematics–Computer Science | University of California, San Diego
Email: amirrezash007@gmail.com | GitHub: github.com/AmirrezaSH007 | LinkedIn: linkedin.com/in/amirreza-shams-123361299

1. URL Shortener API

Developed a RESTful API using Node.js, Express, and PostgreSQL with Docker. Implemented URL shortening, redirects, and click analytics with validation, rate limiting, and Jest testing.

```
app.post('/api/shorten', async (req, res) => {
  const { url } = req.body;
  const slug = nanoid(7);
  await pool.query('INSERT INTO urls (slug, target) VALUES ($1, $2)', [slug, url]);
  res.json({ short_url: `${req.protocol}://${req.get('host')}/${slug}` });
});
```

2. Job Application Tracker

Created an offline-ready web dashboard to manage internship applications. Implemented analytics with Chart.js and full test coverage using Jest and Puppeteer for UI automation.

```
// add_application.js
form.addEventListener('submit', (e) => {
  e.preventDefault();
  const app = getFormData();
  saveToLocalStorage(app);
  alert('Application added successfully!');
  window.location.href = 'applications.html';
});
// applications-page.js
function renderCards(apps) {
  cards.innerHTML = '';
  apps.forEach(app => {
    const card = document.createElement('job-app-card');
    card.setAttribute('data-id', app.id);
    cards.appendChild(card);
  });
}
```

Testing Snippet (Puppeteer):

```
await page.goto('http://localhost:5500/add_application.html');
await page.type('#company', 'OpenAI');
await page.click('#submit-btn');
await page.waitForSelector('.success-message');
```

3. Cryptographic Authentication Attack (CSE 107)

Implemented a simplified Blast-RADIUS attack using PlayCrypt in Python to simulate adversarial message-forging scenarios and authentication bypass, deepening understanding of protocol vulnerabilities.

```
def forge_tag(valid_tag):
    fake_tag = xor(valid_tag, random_pad())
    return fake_tag
```

4. Playing Card Project

Built card deck simulation with shuffle, draw, and hand-evaluation logic. Wrote modular Jest tests for deck state validation and randomization correctness.

```
class Deck {
  constructor() { this.cards = shuffle(generateDeck()); }
  draw(n=1) { return this.cards.splice(0, n); }
}
```