Ariel Semel

arielsemel12@gmail.com | 050-9435516 | https://www.linkedin.com/in/ariel-semel
https://github.com/ArielSemel

SUMMARY

Computer Science graduate with experience building scalable, secure full-stack applications. Developed real-time football tracking and auction platforms using **ReactJS**, **Spring Boot**, and **MySQL**. Proficient in testing and automation with **JUnit** and **Selenium**, ensuring high-quality deliverables. Optimized **AWS** deployments, reducing costs by **20%** with auto-scaling and load balancing. Skilled in building **RESTful APIs** and leveraging **cloud technologies**. Eager to apply my skills to drive impactful, high-performance solutions.

EDUCATION

2020 – 2023 **B.Sc in Computer Science**
Academic College of Ashkelon

2023 – 2024 **DevOps Certification**
Udemy (125 hours)
AWS, Ansible, CI/CD, Jenkins, Git, Docker, K8, Linux, Terraform, Microservices

PROJECTS

Full-Stack REST API - Football Tracker & Auction Platform
Technologies: Java Spring Boot | AWS | ReactJS & MUI | MySQL | Docker | JMeter | Postman | JUnit5 | Selenium

Real-Time Football Match Tracker
Built a platform for users to create games, manage scores, and view live update scores and league tables.
Real-Time Auction Management System
Built a platform for users to upload products, place bids, and manage balances with live bid updates.

Frontend
- Built responsive **SPA**s with modular, reusable components for efficiency and scalability.
- Deployed frontend applications on **AWS S3** with **CloudFront** for secure and fast content delivery.

Backend
- Developed backend services following the **MVC** architecture, ensuring clean code separation and scalability.
- Used Spring Data **JPA** and Hibernate **ORM** for efficient database interactions.
- Ensured data consistency with **Transactional Management** for bids, live scores, and balance management.
- Designed **WebSocket** for real-time updates like live scores, bid placements, auction closing).

Database
- Integrated **MySQL** and deployed it on **AWS RDS** with multi-AZ replication for reliability.

Security
- Implemented **JWT**-based authentication and role-based authorization for secure access.
- Used **AWS Secret Manager**, and **IAM** roles for secure deployments.

Deployment & Scalability
- Containerized applications using **Docker** to ensure consistent and scalable deployments across environments.
- Deployed on **EC2** and **S3**, and optimized auto-scaling, Reduced costs by **20%** while maintaining high availability.
- Configured **Route 53** for **DNS** management and ensured secure, high-performance deployment under a **VPC**.

Performance & Testing & Automation
- Conducted load testing using **JMeter**, ensuring the system could handle real-world traffic spikes.
- Improved throughput by **15%** during peak usage, ensuring low latency even under high load.
- Utilized **Selenium** and **JUnit** to automate browser interactions and perform functional testing.