Vishal Pawar

Transforms business requirements into efficient solutions; versatile full-stack developer and automation engineer.

Email: vshalofficial@gmail.com | **Contact No.** +91 7588056314 | **LinkedIn**: https://www.linkedin.com/in/vishalpawar1010
Portfolio: https://sites.google.com/view/vishalpawar | **GitHub**: https://github.com/VishalPawar1010

Experience

Company: **Argusoft India** — Gandhinagar, Gujrat, India

Position: ***Solution Analyst*** — July 2022-Present

Project Domain: University Management Suite (USA): (AngularJS, Spring Boot, GraphQL, ReactJS)

- Modernization of Web app from REST to GraphQL: Developed a GraphQL resolver generator in Java/Spring Boot, automating resolver class creation for 900+ services, handling non-standard cases, optimizing data extraction, and reducing generation time to 2 minutes.
    - **Reduced the number of APIs by 89%**, from 3193 REST APIs to 350 GraphQL APIs.
    - Decreased manual coding, development, integration, and testing time using the generator
        - **Achieved 95+% reduction**: From 20 days to 1 day for 250+ controllers.
        - **Achieved 96+% reduction**: From 1 day to 1 hour per resolver (250+ resolvers).
    - Achieved a 99+% acceptance rate of the code generator by developers.
- Developed ID-resolver to generate required methods by in-depth analysis to build robust solution logic for complex problems, **which scans and identifies specific information in 900+ services, 3000+ methods**, their respective metadata, and method arguments.
- Developed and maintained React components for university course management. Managed component state using React hooks and Context API
- Implemented GraphQL queries and mutations using Apollo Client. Integrated REST API calls using Axios.
- Designed and styled user interface components using Tailwind CSS
- Developed UI with pixel-perfect design, ensuring high fidelity to design specifications
- Implemented error handling and debugging techniques. Optimized code for better readability and performance
- Designed and optimized performance testing scripts using K6, **reducing CPU usage by 70%** and **code duplication by 10%** through modularization.
- Automated testing with scheduled Bamboo jobs and developed JS scripts to cut function listing time from days to seconds, enhancing efficiency and reliability in performance analysis.
- Designed and implemented a robust CourseQuery API to fetch courses by IDs with strict JSON validation, input sanitization, and detailed error reporting via response headers.
- Explored ANTLR and JEXL technologies to build a proof-of-concept parser. Refactored legacy code for maintainability and implemented multithreading, **reducing execution time by 73%**. Contributed to performance and scalability improvements.

Project Domain: Api Test Automation Tool (Java, Quarkus, Automation, Microservices)

- Engineered osid-api-tester, a robust backend tool to automate validation of Single and Chained OSID REST APIs, reducing manual API QA **effort by 90%**.
- Enabled testing of dynamic API endpoints (GET/POST) with configurable path/query parameters and support for JSON input files or directories.
- Built logic to extract dynamic values (FETCH) from API responses and use them in chained API calls—ensuring full test coverage across interdependent endpoints.
- Handled validation of responses containing both single JSON objects and nested arrays.
- **Covered over 95% of functional edge cases**, including chained request failures, data mismatches, and inconsistent attribute sets.
- Developed detailed logging mechanisms (.log, .json) with support for FAILURE_ONLY and ALL modes, aiding faster debugging and audit trails
- Significantly improved backend stability and deployment confidence by catching integration issues before QA or production environments.

Project Domain: DevOps-CICD (Jenkins, Kubernetes, Automation)

- **Built and maintained a complete Jenkins CI/CD pipeline** to automate the entire backend service lifecycle—from code integration to deployment and validation.
- Configured Email-SCM integration, enabling automated notifications for pipeline status, failures, and successful builds to ensure prompt team visibility.
- **Designed a multi-stage pipeline** that:
    - Compiles and builds a Gradle-based Quarkus project.

    - Builds and tags a Docker University Image.
    - Authenticates with Docker registry.
    - Pushes/pulls Docker image for deployment.
    - Executes a custom Bash script to export the database and configure necessary privileges.
    - Deploys and runs the Quarkus application in a containerized environment.
- Implemented a **sequential job trigger**, ensuring:
    - After successful deployment, a secondary Jenkins job validates the app using the `osid-api-tester` framework.
    - Automatically verifies if the app is live before executing REST API validations.
    - Accepts single or multiple JSON input files for comprehensive test coverage.
- Automated post-validation email reports summarizing **success or failure**, improving deployment feedback loops and reducing manual checks.
- Migrated **CI/CD from Jenkins to Kubernetes pods and pipelines**, improving deployment speed and scalability across environments.
- **Developed API Endpoint Health Check Script**: Checks multiple API endpoints for expected HTTP status codes and logs detailed errors for failures.
- **Developed Jenkins CI/CD Automation Script**: Automates sequential triggering, monitoring, and artifact retrieval of two Jenkins jobs with optional input JSON.
- **Developed Even-week DB Backup Cron Script**: Automates conditional database backup via cron, executing only on even weeks with detailed logging to avoid unnecessary operations.
- **Developed PostgreSQL Backup and Upload Script**: Backs up PostgreSQL DB, logs progress, uploads to Google Drive via rclone, and sends email alerts based on success/failure.

Project Domain: Employee Management Suite (MERN, ReactJS, NodeJS, SpringBoot)

- Developed backend with NodeJS, ExpressJS, and MongoDB.
- Employed JWT-based spring security to ensure secure authentication and authorization.
- Created full-stack functionalities to validate forms, guaranteeing data integrity and accuracy.
- Implemented a pagination feature on the user interface, thereby improving usability and performance.
- Utilized flyway to facilitate seamless data migration during application setup on new machines.
- Conducted comprehensive testing on the server side using Junit, Jacoco, and SonarQube, and on the client side using Karma and Cypress. **The code coverage threshold was set at 80%** to ensure thorough testing.
- Integrated functionality to enable the use of avatar images for user profile pictures, thereby enhancing the user experience.

Open Source Contribution (React and Typescript**)**

Project Domain: AI Development Tool SDK (pieces-app/example-typescript)

- Fixed response formatting using Rehype and ReactMarkdown to enhance UI consistency and readability.
- Added config to set baseURL automatically, improving deployment efficiency and reducing manual configuration.
- Implemented login-logout functionality, ensuring secure user authentication and session management.
- Upgraded methods for Copilot using pieces Copilot SDK
- Added snippet preview feature, allowing users to quickly view code snippets, improving usability and developer experience

Skills

- **Programming Languages:** Java, Javascript, TypeScript, SQL, HTML, CSS, REST
- **Frameworks**: Spring Boot, NodeJS, ReactJS, Angular 2+, AngularJS, Quarkus
- **DevTools:** IntelliJ, Visual Studio Code, SQL-Developer, GitHub, Jira
- **DevOps & Monitoring:** Jenkins, CICD, k6, Grafana, Docker, Bamboo, Kibana, AWS, Linux

Education

2020-2022	**Visvesvaraya National Institute of Technology, Nagpur, Maharashtra, India** *Masters of Technology (MTech): Materials Engineering*	8.88
2016-2019	**JSPM's Rajarshi Shahu COE, Pune, Maharashtra, India. (BE:** *Mechanical Engineering)*	8.31