Hi!

I'm Bibek Saini

I am a passionate software engineer who love to learn, experiment and build something that creates value.

I specialise in designing architecture, developing seamless user experiences, and handling end-to-end deployment. Passionate about solving complex problems, I work across the front-end and back-end technologies — Java, .NET, Kotlin, Angular, React, Spring Boot, Docker, Kubernetes, and Azure, to name just a few. Whether it's a single line of clean code or performance optimisation, my ultimate aim is always to build trusted systems with real value.

Bachelors of Technology in Computer Science.

Contact Me

Toronto, Canada

+1 (416) 559-9209

itsbibeksaini@gmail.com

Social Media

linkedin.com/in/itsbibeksaini/

github.com/itsbibeksaini

itsbibeksaini.com

Highlights

- 7 years of experience specializing in designing architecture, developing, deploying SaaS applications and everything in between.
- Well proven analytical, communication and decision making skills with ability to learn fast, innovate and problem solve under pressure.
- Successful delivered multiple sessions on various topics like factory pattern, at organization level.
- Bachelor’s degree in Computer Science.

Relevant experience

- Developed Gradle command line tasks to automate generation new SpringBoot projects along with custom code files using Gradle API and Kotlin.
- Developed Gradle custom plugins to encapsulate and centralize gradle common build logic used by multiple micro-services using Gradle API and Kotlin.
- Worked on custom StringBoot starters and associated libraries developing various auto-configs to provide consistent dependency alignment and configurations.
- Collaborated with architects to contribute in migrating as well as extracting out capabilities from monolithic application to micro-services pattern.
- Created as well and re-structured/re-invented multiple micro-service application using SpringBoot, ASP.NET as backend framework along with Java, Kotlin, C#.NET as backend languages to write business logic respectively.
- Designed RESTful APIs using REST patterns and used Swagger UI(Open API) to generate API documentation.
- Create gRPC clients and well as servers for internal communication between micro-services.
- Researched and proposed various security by default methods like Spring Security method, claims based authorization and many more.
- Used Microsoft SQL Server and made Entity Relationship Diagrams(ERD) to visualize business model.
- Utilized various ORM tools like String Data JPA for Spring applications, Entity Framework for .NET applications for storing and accessing the data from database.
- Used Gradle build tool and wrote Gradle build scripts for projects.
- Followed TDD approach while creating micro-services and wrote test cases using jUnit, Mockito and xUnit for .NET applications.
- Developed multiple frontend applications, re-usable components using Angular(8+), React, RxJS, Observables, TypeScript, JavaScript(ES6+).

Skills

Frontend

Angular | React | HTML5 | CSS3 | SCSS | JavaScript(ES6+) | TypeScript

Backend

Java 7+ | C#.NET | Kotlin | ASP.NET | SprinBoot | Gradle API

Database tools

SQL Server | Azure SQL | MongoDB | PostgresSQL | JPA | SpringData | Entity Framework

Deployment tools

Docker | Kubernetes | Helm | Azure Kubernetes | Azure Pipelines

Frameworks & design patterns

- Developed UI/UX components using HTML5, CSS3(Bootstrap, Material UI and Animate.css) and CSS preprocessors such as SCSS.
- Deployed Azure Pipelines templates for CI/CD to automate deployment on Azure Kubernetes Services(AKS).
- Created various Kubernetes resources .yml(s) templates for services to be deployed on Kubernetes cluster.
- Used Helm deployment tool for automating creation, packaging, configuration and deployment of applications and services to Kubernetes cluster.
- Used Docker tools to create container images for services and utilized Azure Container Registry to store container images.
- Utilized Git version control to manage projects, created pull requests(PR), did code reviews and managed branches on GitHub.
- Mentored team members as well as other teams for building strategies of various modules.
- Used Jira, Confluence for software documentation & sprint management.

Work history

Freelance

Senior software engineer

08/2023 - Present

CatalystOne Info Solution Pvt Ltd.

Senior software engineer

04/2016 - 03/2023

Projects

Pose Points

React, SpringBoot and Kotlin.

- PosePoints is a mobile responsive as well as web based application, to manage the Indian's National level Yoga competitions.
- **Technology used:**
  - React for designing web client along with TypeScript.
  - SpringBoot for creating production-grade application backed with Kotlin.
  - MS SQL Server for storing application data.

Custom gradle plugins

Gradle API & Kotlin

- An assortment of gradle plugins customized to configure essentials plugins and settings required by projects.

Micro-Services

RESTful API | gRPC

Consul | Ocelot

Version control

Git | Mercurial | GitHub

Bitbucket

Education

Information System Business Analysis

Northern college

05/2024 - 12/2024

Mobile application development

Northern college

05/2023 - 12/2023

Bachelors in Computer Science

I. K. Gujral Punjab Technical University

08/2012 - 06/2016

Technology used:

- Gradle API to create Custom gradle plugin.
- Kotlin as backend language to write plugin logic.

Project generator(code as automation)

Gradle API & Kotlin

Custom gradle plugin with dynamic code generation capabilities to auto generate boiler-plate repetitive code for project.

Technology used:

- Gradle API to create Custom gradle plugin.
- Kotlin as backend language to write plugin logic.
- JavaPoet to dynamically generate java code files(An API to generate java source code files).

Custom SpringBoot starters

SpringBoot & Kotlin

Collection of BOMs, libraries and SpringBoot Starters to provide consistent dependency alignment and configuration conventions across microservices projects to build bootiful services.

Technology used:

- Spring Framework for building custom libraries and SpringBoot Starters.
- Kotlin as backend language to write business logic.
- Swagger UI(Open API) to provide an interface with consistent behavior across projects.
- JetBrains Exposed database ORM tool for performing database operations.
- H2 Database to perform in memory operations for testing.
- KFlyway used for migrating database schema automatically.
- JUnit to write test cases for libraries and SpringBoot Starters.

Notification service

SpringBoot & Java

Micro-service providing email and sms notification capabilities. Implements JMS and Azure Service Bus.

Technology used:

- SpringBoot for building RESTful Service.
- Java as backend language to write business logic.
- JMS API to build loosely coupled, reliable and asynchronous communicating application.
- Azure Service Bus message broker with message queues to decouple applications and services from each other.

Pending task service

SpringBoot & Java

Pending task service collects various task information from different services in micro-services architecture and helps in presenting it in singular form.

Technology used:

- SpringBoot for creating production-grade application backed with Java.
- Swagger UI(Open API) for documenting RESTful API endpoints.
- SpringData & JPA ORM tools for data access layer in an application.
- MS SQL Server for storing application data.

Task manager

Micro-services

Built on micro-services architecture this application provides management services like Invoice Manager and Task Board which allow users to manage their task and help organization's digitize their invoice generation system. The whole project is divided into various micro-apps based on their individual responsibility that each micro app need to perform, which are deployed on Azure Kubernetes Services.

Technology used:

- Angular(8+) for frontend client application along with TypeScript.
- SpringBoot with Java and Kotlin for creating micro-services.
- ASP.NET Core Web API with C#.NET for creating micro-services.
- gRPC for service to service internal communication in cluster and RESTful APIs for communicating with web clients.
- Consul Service Mesh utilizing Service Discovery for service-to-service connection.
- Docker for creating container images, Kubernetes for container-orchestration for automating computer application deployment, scaling and management.

Workflows

Java EE

Workflows modules help organizations to manage their tasks which occurs in particular order. With workflows organizations can design the flow with various pre-defined task.

Technology used:

- Java EE & Core Java as backend language.
- HTML, JavaScript, CSS as frontend presentation languages.
- Apache velocity template engine to merge dynamic response with HTML templates.