

Bibek Saini

Senior Software Engineer.

CONTACT ME

itsbibeksaini.com

itsbibeksaini@gmail.com

+91 9478889924

github.com/itsbibeksaini

linkedin.com/in/itsbibeksaini

SKILLS

FRONTEND

Angular 7+ React.js HTML5
CSS3 SCSS Javascript(ES6+)
Typescript

BACKEND

Java 7+ Asp.net core
Entity framework Spring boot
JPA Spring data

DATABASE TOOLS

SQL Server Azure SQL
MongoDB

DEPLOYMENT TOOLS

Docker Kubernetes
Azure kubernetes Helm

FRAMEWORKS & DESIGN PATTERNS

Micro-services RESTful API
gRPC Consul Ocelot Helm

VERSION CONTROL

Git Mercurial Github
Bitbucket

EDUCATION

I. K. Gujral Punjab

Technical University

2012 - 2016 Amritsar, Punjab IN
Bachelor's of Technology Computer
Science and Engineering

HIGHLIGHTS

- 7 years of experience as full stack software engineer in developing SaaS applications, building HCM cloud application.
- Bachelor's degree in Computer Science and Engineer.
- 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.

WORK EXPERIENCE

CatalystOne Info Solutions Pvt Ltd.

Senior Software Engineer

(July, 2018 - March, 2023)

- Developed multiple frontend applications, re-usable components using Angular(8+), RxJS, Observables, TypeScript, JavaScript(ES6+).
- Developed UI/UX components using HTML5, CSS3(Bootstrap, Material UI and Animate.css) and CSS preprocessors such as SAAS and LESS.
- Built RESTful API's to perform CRUD operations and used Swagger(OpenAPI) to design, build and document.
- Mentored multiple teams for building strategies of various modules.
- Used Jira, Confluence for software documentation & sprint management.
- Used Microservices architecture to isolate software functionality into multiple independent modules which were responsible for performing predefined, standalone tasks. These modules communicated with each other through universally accessible APIs.
 - Used Consul service mesh to implement service discovery for various services in Kubernetes cluster.
 - Incrementally extract modules out of the monolith and convert them into multiple independent services.
 - Fine tunes services for the resources that they can request and has limits in Kubernetes environment. Also, added various horizontal pod autoscale(HPA) strategies.
 - Added Kubernetes probes which ensures container inside a pod is healthy and ready to serve.
 - Used cert-manager CRD in Kubernetes to automate the management and issuance of TLS certificates.
 - Used docker tool to create containers to package up an application with all of the parts it needs.
 - Added support for SpringBoot Admin(SBA) to collect metrics provided by actuator endpoints.
- Deployed Azure Pipelines for CI/CD, TDD to automate deployment on Azure Kubernetes Services(AKS).
- Used Microsoft SQL Server and made Entity Relationship Diagrams(ERD) to visualize business model.
- Version control system using GIT repository, created pull requests, did code reviews and managed branches.

Software Developer

(April, 2016 - July, 2018)

- Designed rewards and recognition(R&R) module that will help to increase motivation of employee to get the best achievement.
 - Designed interactive UI using HTML5, CSS3, jQuery and used Apache velocity template engine to merge dynamic data in template.
 - Created REST API endpoints using SpringBoot, JPA, Spring Data.
 - Developed various layers of business logic, data access layer using Java(8) and used Microsoft SQL Server.
 - Designed and implemented Invoice Manager to help client's organisation digitize their invoice generation system and manage their sales data.
 - Designed backend application using ASP.NET MVC and implemented core framework using C#.
 - Used Dependency Injection to reduce the tight coupling among software components.
 - Used Entity Framework Object Relational Mapping (ORM) framework for storing and accessing the data in the database.
 - Wrote unit test using xUnit.

PROJECTS

Task manager

Micro-services app

- 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 organisation'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.
- Technologies used:**
 - Angular(8+) for frontend client application
 - Java Spring Framework, ASP.NET Core web api for backend micro web apps, gRPC for inner app communication and RESTful API for communicating with client.
 - Consul Service Mesh combined with Ocelot Api to built a gateway api for communication with client application.
 - Docker for container image creation, Kubernetes for container-orchestration for automating computer application deployment, scaling, and management.

Workflows

Java EE

- Workflows modules help organizations to manage tasks which occurs in particular order. With workflows organizations can design the flow with various pre-defined task.
- Technologies used:**
 - Core java and java ee as backend languages.
 - HTML, JavaScript, CSS as forntend presentation languages.

Pending task service

SpringBoot

- Pending task service collects various task information from different services in micro-services architecture and helps in presenting it in singular form.
- Technologies used:**
 - SpringBoot framework to build core framework and Java to write business logic.
 - JPA and SpringData for storing and accessing the data in database.
 - Swagger(OpenAPI) for api documentation.

Custom gradle plugins

Gradle API & Kotlin

- An assortment of gradle plugins customised to configure essentials plugins and settings required by project.
- Technologies used:**
 - Gradle API to create Custom gradle plugin.
 - Kotlin as backend language to write plugin logic.

Project generator(scaffolding new projects)

Gradle API & Kotlin

- Custom gradle plugin with which we can easily scaffold(auto generate boiler-plate code) projects as per requirements.
- Technologies used:**
 - Gradle API to create Custom gradle plugin.
 - Kotlin as backend language to write plugin logic.
 - JavaPoet to scaffold java code files.(An API to generate java source code files).