| **Feature** | **Release 1** | **Release 2** | **Release 3** |
| --- | --- | --- | --- |
| **FE-1**, Scalability | Basic horizontal scaling using load balancers | Advanced horizontal scaling with auto-scaling groups | Full implementation of horizontal and vertical scaling with auto-scaling, load balancing, and caching |
| **FE-2**, Enhanced Customer Experience | Basic user interface improvements | Advanced user interface enhancements with responsive design and accessibility features | Full implementation of user interface improvements with optimized performance and personalized experiences |
| **FE-3**, Developer Agility | Essential integration with version control systems | Advanced integration with CI/CD pipelines and automated testing | Full implementation of developer agility with continuous deployment, feature flagging, and automatic rollback mechanisms |
| **FE-4**, CI/CD | Basic setup of CI/CD pipelines for build and deployment | Advanced CI/CD pipelines with automated testing and release management | Full implementation of CI/CD with continuous integration, automated testing, deployment pipelines, and release orchestration |
| **FE-5**, Disposable Infrastructure | Not Implemented | Basic implementation of infrastructure as code using Terraform | Full implementation of disposable infrastructure with automated provisioning, scaling, and teardown using Terraform and Kubernetes |
| **FE-6**, Decoupling of Services | Essential service decoupling using API gateways | Advanced service decoupling with event-driven architecture and microservices | Full implementation of service decoupling with microservices, event-driven communication, and service mesh |
| **FE-7**, Cloud Native Technologies (Kubernetes, Microservices, Containers, Terraform) | Basic containerization using Docker | Advanced container orchestration with Kubernetes | Full implementation of cloud-native technologies with containerization, orchestration, and infrastructure as code using Kubernetes, Docker, and Terraform |

In this table:

* **FE-1, Scalability**: The project will initially focus on implementing basic horizontal scaling using load balancers. **Release 2** will advance to using auto-scaling groups, and **Release** **3** will fully implement horizontal and vertical scaling with auto-scaling, load balancing, and caching.
* **FE-2, Enhanced Customer Experience**: The project will start with basic user interface improvements in the first release. **Release 2** will introduce advanced enhancements with responsive design and accessibility features. Finally, **Release 3** will fully implement user interface improvements with optimized performance and personalized experiences.
* **FE-3, Developer Agility**: The project will begin by integrating with version control systems in the first release. In **Release 2**, it will advance to integration with CI/CD pipelines and automated testing. In **Release 3**, it will fully implement developer agility with continuous deployment, feature flagging, and automatic rollback mechanisms
* **FE-4, CI/CD**: The project will start by setting up primary CI/CD pipelines for build and deployment in the first release. **Release 2** will introduce advanced CI/CD pipelines with automated testing and release management. Finally, **Release 3** will fully implement CI/CD with continuous integration, automated testing, deployment pipelines, and release orchestration.
* **FE-5, Disposable Infrastructure**: This feature will not be implemented in the first release. However, **Release 2** will have a basic implementation of infrastructure as code using Terraform. **Release 3** will fully implement it with automated provisioning, scaling, and teardown using Terraform and Kubernetes.
* **FE-6, Decoupling of Services**: The project will start with basic service decoupling using API gateways in the first release. In **Release 2**, it will advance to decoupling services with event-driven architecture and microservices. Finally, **Release 3** will fully implement service decoupling with microservices, event-driven communication, and service mesh.
* **FE-7, Cloud Native Technologies** (Kubernetes, Microservices, Containers, Terraform): The project will begin by implementing basic containerization using Docker in the first release. In **Release 2,** it will introduce advanced container orchestration with Kubernetes. Finally, **Release 3** will fully implement cloud-native technologies with containerization, orchestration, and infrastructure as code using Kubernetes, Docker, and Terraform.