사용도구

협업도구

- 이슈관리 : JIRA
- 형상관리 : GitLab, Git
- 커뮤니케이션 : MatterMost, Notion

배포도구

- CI/CD : Jenkins, Docker, EC2

설계도구

- 와이어프레임 : Figma
- 요구사항 정의서 : Notion, Google Sheet
- 기능명세서 : Google Sheet

개발도구

- VSCode, Visual Studio 2022
- Unity Engine 2022.3.22f
- IntelliJ IDEA

개발환경

Web FrontEnd

```
    "@tanstack/react-query": "^5.28.9",
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "@types/jest": "^27.5.2",
    "@types/node": "^16.18.91",
    "@types/react": "^18.2.67",
    "@types/react-dom": "^18.2.22",
```

```
    "axios": "^1.6.8",
    "i18next": "^23.10.1",
    "js-cookie": "^3.0.5",
    "react": "^18.2.0",
    "react-cookie": "^7.1.0",
    "react-dom": "^18.2.0",
    "react-i18next": "^14.1.0",
    "react-router-dom": "^6.22.3",
    "react-scripts": "5.0.1",
    "typescript": "^4.9.5",
    "web-vitals": "^2.1.4",
    "zustand": "^4.5.2"
```

BackEnd (Yaml File)

```
server:
  port: 8080
spring:
  data:
    mongodb:
      uri: mongodb:8888/user?authSource=admin

  mail:
    host: smtp.naver.com
    port: 465
    username:
    password:
    properties:
      mail.smtp.auth: true
      mail.smtp.ssl.enable: true
      mail.smtp.ssl.trust: smtp.naver.com
      mail.smtp.starttls.enable: false

logging:
  level:
    root: DEBUG
```

NginX Setting

- **SSL 보안설정용**

```
user nginx;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;

        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        server {
            listen 80;
            server_name j10a609.p.ssafy.io www.j10a609.p.ssafy.io;

            return 301 https://$server_name$request_uri;
        }

        ##
        # SSL Settings
        ##
        server {
            listen 443 ssl;
            server_name j10a609.p.ssafy.io www.j10a609.p.ssafy.io;
```

```
            access_log /var/log/nginx/access.log;
            error_log /var/log/nginx/error.log;


            ssl_certificate
/etc/letsencrypt/live/j10a609.p.ssafy.io/fullchain.pem;
            ssl_certificate_key
/etc/letsencrypt/live/j10a609.p.ssafy.io/privkey.pem;

            ssl_protocols TLSv1.2 TLSv1.3;
            ssl_prefer_server_ciphers on;
            ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384';
            ssl_session_cache shared:SSL:10m;
            ssl_session_timeout 10m;

            location /download {
                alias /usr/share/nginx/html/game;
                autoindex on;
                try_files $uri $uri/ =404;
            }

            location /api {
                proxy_pass http://michelin_de_hanyang:8080;
                proxy_set_header Authorization $http_authorization;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                add_header 'Access-Control-Allow-Origin' '*';
                add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS,
PUT, DELETE';
                add_header 'Access-Control-Allow-Headers' 'X-Requested-
With,Content-Type,X-Token-Auth,Authorization';
                add_header 'Access-Control-Allow-Credentials' 'true';
            }

            location / {
                proxy_pass http://front:80;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
            }
        }
```

```
        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;


        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}
```

- **프론트엔드 호스팅용**

```
-  events {}
-
-  http {
-      server {
-      listen  80;
-      include /etc/nginx/mime.types;
-      server_name www.j10a609.p.ssafy.io j10a609.p.ssafy.io;
-
-      location / {
-          root    /usr/share/nginx/html;
-          index   index.html index.htm;
-          try_files $uri $uri/ /index.html;
-          }
-      }
-  }
```

Docker-Compse File

```
services:
  frontend:
    image: mjbae327/front:latest
    container_name: front
    ports:
      - "3000:80"
    volumes:
```

```
      - /etc/letsencrypt:/etc/letsencrypt
    environment:
      TZ: Asia/Seoul
    networks:
      - mynetwork

  backend:
    image: mjbae327/michelin_de_hanyang:latest
    container_name: michelin_de_hanyang
    ports:
      - "8080:8080"
    extra_hosts:
      - "host.docker.internal:host-gateway"
    volumes:
      - /etc/letsencrypt:/etc/letsencrypt
      - /home/ubuntu/config/application.yml:/config/application.yml
    environment:
      - SPRING_CONFIG_LOCATION=file:/config/application.yml
      - TZ=Asia/Seoul
    networks:
      - mynetwork

  nginx:
    image: nginx:alpine
    volumes:
      -
/home/ubuntu/jenkins/workspace/release/nginx.conf/nginx.conf:/etc/nginx/nginx.
conf
      - /etc/letsencrypt:/etc/letsencrypt
      - /home/ubuntu/game:/usr/share/nginx/html/game
    ports:
      - "443:443"
    depends_on:
      - backend
    networks:
      - mynetwork
networks:
  mynetwork:
    driver: bridge
```

Jenkins Pipeline

```
pipeline {
  agent any
```

```
  environment {
      IMAGE_BACK = "mjbae327/michelin_de_hanyang"
      IMAGE_FRONT = "mjbae327/front"
      DOCKER_TAG = 'latest'
      SERVER_ACCOUNT = 'ubuntu'
      SERVER_IP_ADD = 'j10a609.p.ssafy.io'
      WEBHOOK_URL = credentials('webhook')
      APPLICATION_YML = credentials('yaml')
      REACT_APP_API = "https://j10a609.p.ssafy.io/api"
  }

  stages {
      stage('Git Checkout') {
          steps {
              git branch: 'BEDev', credentialsId: 'gitlab', url:
'https://lab.ssafy.com/s10-metaverse-game-sub2/S10P22A609'
          }
      }

      stage('Prepare Environment') {
          steps {
              script {
                  dir('BackEnd/michelin_de_hanyang') {
                      sh 'chmod +x ./gradlew'
                      sh './gradlew clean'
                      sh './gradlew build -x test'
                  }
              }
          }
      }

      stage('Copy application.yml') {
          steps {
              withCredentials([file(credentialsId: 'yaml', variable:
'APPLICATION_YML')]) {
                  sh "cp
$APPLICATION_YML ./BackEnd/michelin_de_hanyang/src/main/resources/"
              }
          }
      }

      stage('Build Image') {
          steps {
              script {
                  sh "docker build --no-cache -t
${IMAGE_BACK}:${DOCKER_TAG} ./BackEnd/michelin_de_hanyang"
                  sh "docker build --build-arg REACT_APP_API=${REACT_APP_API}
-t ${IMAGE_FRONT}:${DOCKER_TAG} ./Web/michelin-web"
```

```
            }
        }
    }

    stage('Docker Login and Push') {
        steps {
            script {
                withCredentials([usernamePassword(credentialsId:
'dockerhub', passwordVariable: 'PASSWORD', usernameVariable: 'USERNAME')]) {
                    sh "echo ${PASSWORD} | docker login -u ${USERNAME} --
password-stdin"
                    sh "docker push ${IMAGE_BACK}:${DOCKER_TAG}"
                    sh "docker push ${IMAGE_FRONT}:${DOCKER_TAG}"
                }
            }
        }
    }

    stage('Print Current Directory') {
        steps {
            sh 'pwd'
        }
    }

    stage('Deploy') {
        steps {
            script {
                sshagent(['ubuntu']) {
                    sh 'ssh -o StrictHostKeyChecking=no
${SERVER_ACCOUNT}@${SERVER_IP_ADD} "docker-compose -f
/home/ubuntu/jenkins/workspace/release/docker-compose.yml pull && docker-
compose -f /home/ubuntu/jenkins/workspace/release/docker-compose.yml up -d"'
                    sh 'ssh -o StrictHostKeyChecking=no
${SERVER_ACCOUNT}@${SERVER_IP_ADD} "cd /home/ubuntu/jenkins/workspace/release/
&& docker-compose exec -T nginx nginx -s reload"'
                }
            }
        }
    }
}

post {
    success {
        script {
            sh '''
            curl -i -X POST -H 'Content-Type: application/json' -d '{
                "attachments": [
                    {
```

```
                        "fallback": "BUILD SUCCESS !",
                        "color": "#6badff",
                        "title": "\n\nBuild Event Success",
                        "text": " ",
                        "fields": [
                            {
                                "short": false,
                                "title": "Branch",
                                "value": "${GIT_BRANCH}"
                            },
                            {
                                "short": true,
                                "title": "Commit",
                                "value": "${GIT_COMMIT}"
                            }
                        ]
                    }
                ]
            }' $WEBHOOK_URL
        '''
        }
    }
    failure {
        script {
            sh '''
            curl -i -X POST -H 'Content-Type: application/json' -d '{
                "attachments": [
                    {
                        "fallback": "BUILD FAIL",
                        "color": "#e78e77",
                        "title": "\n\nBuild Event Fail",
                        "text": " ",
                        "fields": [
                            {
                                "short": false,
                                "title": "Branch",
                                "value": "${GIT_BRANCH}"
                            },
                            {
                                "short": true,
                                "title": "Commit",
                                "value": "${GIT_COMMIT}"
                            }
                        ]
                    }
                ]
            }' $WEBHOOK_URL
            '''
```

```
            }
        }
    }
}
```