



# “어디가농?” 포팅 매뉴얼

작성일: 2025-04-09

작성자: 정나금

목적: GitLab에서 소스 클론 후 로컬/운영 환경에서 빌드 및 배포 가능한 상태로 문서화

## 1 GitLab 소스 클론 이후 빌드 및 배포 문서

### 1) 개발 환경 버전

<Backend>

- **JVM** : Temurin JDK 17
- **WAS** : Spring Boot (embedded Tomcat)
- **SpringBoot** : 3.4.3
- **Web Server** : Nginx 1.27.4
- **IDE** : IntelliJ IDEA 2023.3.8
- **Build Tool** : Gradle
- **plugin** : SonarQube: 6.0.1.5171

<Frontend>

- **Framework** : React 18.3.1
- **Language** : JavaScript
- **Build Tool** : Vite 6.0.5
- **Package Manager** : npm
- **Proxy** : '/api' → http://farmbti-backend:8081

### 2) 빌드 시 사용되는 환경 변수

[ Backend ]

```
DB_URL=jdbc:postgresql://postgres:5432/farmbti_db
DB_USERNAME=ssafy
DB_PASSWORD=ssafy209!@
```

```
REDIS_HOST=redis
REDIS_PORT=6379
```

```
JWT_SECRET=iAmSSAFYedUAYoo_iamddubi_AYOO_WAHTSUP

S3_BUCKET=farmbtibuckets
S3_ACCESS=AKIA2KC2A7AKTJZ37LPO
S3_SECRET=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

KAKAO_API_KEY=a24122d363b6f9d33ce3566b2f018f12

client_id=1TgtpJ9xAyA_fHiy9pRI
client_secret=ijnEAaSVcU
```

[ Frontend ]

```
VITE_BASE_URL=https://j12d209.p.ssafy.io/api/
```

### 3) 배포 시 특이사항

1. **GitHub Flow**를 사용하여 **프론트엔드는 front 브랜치**, **백엔드는 backend 브랜치**에 모든 코드를 올리고, 각 브랜치를 서버에 연결해 배포
2. **Redis, PostgreSQL** 선 기동 필수
3. **포트 충돌 유의**
  - 백엔드: 내부 8081 → 외부 8081 (Nginx는 443 → 프록시)
  - 프론트엔드: 내부 80 → 외부 3000 (Docker에서 Nginx를 통해 노출됨)
4. **.env 파일은 Jenkins에서 Config File Plugin 으로 주입됨**
  - 백엔드: `env-config`
  - 프론트엔드: `front_env`
5. **빌드 순서 주의 (CI/CD 파이프라인 자동화)**
  - 1) GitLab에서 분기 체크아웃 ( `backend` , `front` )
  - 2) `.env` 파일 로드
  - 3) `gradlew build / npm run build`
  - 4) Docker 이미지 빌드 및 배포
6. **React/Vite 정적 리소스는 Nginx에서 `/dist` 로 서빙됨**
  - Jenkins에서 프론트 `dist/` → `/home/ubuntu/nginx/dist` 복사 후 Nginx 재시작
  - `vite.config.js` 의 프록시 설정은 서버단에서 미사용
7. **Nginx는 HTTP → HTTPS 자동 리디렉션**
  - `default.conf` 에서 설정: 80 → 443 리다이렉트
  - SSL 인증서 ( `fullchain.pem` , `privkey.pem` )는 `/etc/letsencrypt` 에서 마운트됨
8. **Nginx는 Spring Boot 백엔드에 다음 경로를 프록시**

- `/api/`, `/swagger-ui.html`, `/v3/api-docs`, `/gs-guide-websocket` 등

#### 9. Docker 네트워크 이름은 'farmbti'로 고정

- 모든 컨테이너는 이 네트워크를 공유해야 통신 가능

#### 10. Prometheus + Grafana + nginx-exporter 사용 시 유의

- `/stub_status` 엔드포인트는 `default.conf` 에서 오픈됨 (nginx-exporter가 수집)

### 4) DB 접속 정보

#### 1. PostgreSQL 14

- DB명: farmbti\_db
- 사용자명: ssafy
- 비밀번호: ssafy209!@
- 호스트: postgres (Docker 내 서비스명 기준)
- 포트: 5432

#### 2. Redis

- 호스트: redis
- 포트: 6379

### 5) 배포 관련 파일

[Backend - docker-compose.yml]

```
services:
  backend:
    container_name: farmbti-backend
    build: .
    image: nageum/farmbti_backend
    ports:
      - "8081:8081"
    env_file:
      - .env
    volumes:
      - ./src/main/resources/application.yml:/app/application.yml
    depends_on:
      - postgres
      - redis
    restart: always
    networks:
      - farmbti

  postgres:
    container_name: postgres
    image: postgres:14
```

```

restart: always
environment:
  POSTGRES_DB: farmbti_db
  POSTGRES_USER: ssafy
  POSTGRES_PASSWORD: ssafy209!@
volumes:
  - postgres_data:/var/lib/postgresql/data
ports:
  - "5432:5432"
networks:
  - farmbti

```

```

redis:
  image: redis:7.2-alpine
  container_name: redis
  restart: always
  ports:
    - "6379:6379"
  volumes:
    - redis_data:/data
  networks:
    - farmbti

```

```

networks:
  farmbti:
    external: true

```

```

volumes:
  postgres_data:
  redis_data:

```

#### [Backend - Jenkinsfile]

```

pipeline {
  agent any

  environment {
    DOCKER_HUB_USERNAME = "nageum"
    IMAGE_NAME = "farmbti_backend"
    CONTAINER_NAME = "farmbti-backend"
    PORT = "8081"
    GIT_CREDENTIALS_ID = 'gitlab-credentials'
    DOCKER_CREDENTIALS_ID = 'docker-hub'
    DOCKER_COMPOSE_PATH = "/usr/local/bin/docker-compose"
  }

  stages {

```

```

stage('Checkout Code') {
  steps {
    checkout([
      $class: 'GitSCM',
      branches: [[name: 'backend']],
      userRemoteConfigs: [[
        url: 'https://lab.ssafy.com/s12-bigdata-dist-sub1/S12P21D209.git',
        credentialsId: 'gitlab-credentials'
      ]]
    ])
    sh 'echo "✅ 코드 체크아웃 완료"'
  }
}

stage('Load .env File') {
  steps {
    configFileProvider([configFile(fileId: 'env-config', targetLocation: 'backend/farmbti/.env')
    script {
      sh '''
        #!/bin/bash
        set -e
        echo "✅ .env 파일 로드 중..."
        export $(grep -v '^#' backend/farmbti/.env | xargs)
        '''
    })
  }
}

stage('Build Backend') {
  steps {
    dir("${env.WORKSPACE}/backend/farmbti") {
      sh '''
        #!/bin/bash
        set -e
        echo "✅ 백엔드 빌드 시작"
        export JAVA_TOOL_OPTIONS="-Dfile.encoding=UTF-8"
        export $(grep -v '^#' .env | xargs)
        chmod +x gradlew
        ./gradlew clean build -x test
        echo "✅ 빌드 완료"
        '''
    }
  }
}

stage('Build Docker Image') {
  steps {

```

```

    dir('backend/farmbti') {
        sh """
        #!/bin/bash
        set -e
        echo "✅ Docker 이미지 빌드 시작"
        docker build -t ${DOCKER_HUB_USERNAME}/${IMAGE_NAME}:latest .
        echo "✅ Docker 이미지 빌드 완료"
        """
    }
}

stage('Push Docker Image') {
    steps {
        withDockerRegistry([credentialsId: DOCKER_CREDENTIALS_ID, url: 'https://index.docker
        sh """
        #!/bin/bash
        set -e
        echo "✅ Docker Hub에 이미지 푸시 중..."
        docker push ${DOCKER_HUB_USERNAME}/${IMAGE_NAME}:latest
        echo "✅ Docker 이미지 푸시 완료"
        """
    }
}

stage('Deploy') {
    steps {
        script {
            echo "✅ 빌드된 파일 확인"
            sh 'ls -la backend/farmbti/build/libs/'

            echo "✅ 배포 경로 확인 및 생성"
            sh 'mkdir -p /var/jenkins_home/app/backend/farmbti/build/libs/'

            echo "✅ 빌드 파일 복사"
            sh '''
            cp -f backend/farmbti/build/libs/*.jar /var/jenkins_home/app/backend/farmbti/build/
            cp -f backend/farmbti/docker-compose.yml /var/jenkins_home/app/backend/farmb
            cp -f backend/farmbti/Dockerfile /var/jenkins_home/app/backend/farmbti/
            cp -f backend/farmbti/.env /var/jenkins_home/app/backend/farmbti/
            '''

            echo "✅ 배포 시작"
            sh 'cd /var/jenkins_home/app/backend/farmbti/ && $DOCKER_COMPOSE_PATH down'
            sh 'cd /var/jenkins_home/app/backend/farmbti/ && $DOCKER_COMPOSE_PATH build'
            sh 'cd /var/jenkins_home/app/backend/farmbti/ && $DOCKER_COMPOSE_PATH up -d'
        }
    }
}

```

```

        echo "✅ 배포 완료!!"
        sh 'docker ps -a | grep farmbti-backend'
    }
}
}
}
}

```

#### [Monitoring - docker-compose.yml]

```

version: '3'

services:
  prometheus:
    image: prom/prometheus
    container_name: prometheus
    volumes:
      - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
    ports:
      - "9090:9090"

  grafana:
    image: grafana/grafana
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana-storage:/var/lib/grafana
    environment:
      - GF_SECURITY_ADMIN_PASSWORD=admin123

  node-exporter:
    image: prom/node-exporter
    container_name: node-exporter
    ports:
      - "9100:9100"
    restart: always

volumes:
  grafana-storage:

```

#### [Monitoring - prometheus.yml]

```

scrape_configs:
  - job_name: 'spring-boot-app'

```

```
metrics_path: '/actuator/prometheus'
static_configs:
  - targets: ['farmbti-backend:8081']
```

[Frontend - Jenkinsfile]

```
pipeline {
  agent any

  tools {
    nodejs 'node18'
  }

  environment {
    GIT_CREDENTIALS_ID = 'gitlab-credentials'
    FRONT_PROJECT_DIR = "frontend/farmbti"
    DIST_DIR = "frontend/farmbti/dist"
    NGINX_DIR = "backend/infra/nginx"
  }

  stages {
    stage('Checkout Code') {
      steps {
        checkout([
          $class: 'GitSCM',
          branches: [[name: 'front']],
          userRemoteConfigs: [[
            url: 'https://lab.ssafy.com/s12-bigdata-dist-sub1/S12P21D209.git',
            credentialsId: "${GIT_CREDENTIALS_ID}"
          ]]
        ])
        sh 'echo "✅ 코드 체크아웃 완료"'
      }
    }

    stage('Apply Environment Config') {
      steps {
        configFileProvider([configFile(fileId: 'front_env', targetLocation: "${FRONT_PROJECT_DIR}")]
        sh 'echo "✅ 환경 설정 파일 적용 완료"'
      }
    }

    stage('Build Frontend') {
      steps {
        dir("frontend/farmbti") {
          sh '''
```



```

        echo "✅ 프론트엔드 빌드 시작"
        rm -rf node_modules dist
        npm install --legacy-peer-deps
        npm run build
        echo "✅ 프론트엔드 빌드 완료"
        ""
    }
}
}

stage('Copy dist to Nginx') {
    steps {
        script {
            sh '''
            echo "[INFO] 기존 dist 삭제 중..."
            rm -rf /home/ubuntu/nginx/dist

            echo "[INFO] 새로운 dist 복사 중..."
            cp -r frontend/farmbti/dist /home/ubuntu/nginx/dist

            echo "[INFO] dist 복사 완료 ✅"

            echo "[INFO] Nginx 컨테이너 재시작 중..."
            docker restart farmbti-nginx
            echo "[INFO] Nginx 재시작 완료 ✅"
            ""
        }
    }
}
}
}
}
}

```

[Nginx - docker-compose.yml]

```

services:
  nginx:
    build:
      context: .
    image: nginx-nginx
    container_name: farmbti-nginx
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./dist:/usr/share/nginx/html
      - ./default.conf:/etc/nginx/conf.d/default.conf
      - /etc/letsencrypt:/etc/letsencrypt

```

```
networks:
  - farmbti
restart: always
```

```
networks:
  farmbti:
    name: farmbti
    external: true
```

#### [Nginx - default.conf]

```
# HTTP → HTTPS 리디렉션
server {
    listen 80 default_server;
    server_name j12d209.p.ssafy.io;
    client_max_body_size 10M;

    location /stub_status {
        stub_status;
        allow all;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

# HTTPS 처리
server {
    listen 443 ssl;
    server_name j12d209.p.ssafy.io;
    client_max_body_size 10M;

    ssl_certificate /etc/letsencrypt/live/j12d209.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j12d209.p.ssafy.io/privkey.pem;

    location /swagger-ui.html {
        proxy_pass http://farmbti-backend:8081/swagger-ui.html;
    }
    location /swagger-ui/ {
        proxy_pass http://farmbti-backend:8081/swagger-ui/;
    }
    location /v3/api-docs {
        proxy_pass http://farmbti-backend:8081/v3/api-docs;
    }
    location /api-docs {
        proxy_pass http://farmbti-backend:8081/api-docs;
    }
}
```

```

}

location /gs-guide-websocket/ {
    proxy_pass http://farmbti-backend:8081/gs-guide-websocket/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_set_header Host $host;
}

location /api/ {
    proxy_pass http://farmbti-backend:8081;
    rewrite ^/api(/.*)$ $1 break;

    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;
}

location / {
    root /usr/share/nginx/html;
    index index.html;
    try_files $uri $uri/ /index.html;
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}
}

```

[Hadoop - docker-compose.yml]

```

version: '3'
services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    hostname: namenode
    ports:
      - "9870:9870" # NameNode UI
      - "9000:9000" # HDFS 포트
    environment:
      - CLUSTER_NAME=my-hadoop-cluster
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
      - HDFS_CONF_dfs_replication=3

```

```
volumes:
  - ./namenode:/hadoop/dfs/name
  - ./input:/opt/hadoop/input # ✅ 여기에 CSV 파일 접근을 위한 마운트 추가!
networks:
  - hadoop
```

```
secondarynamenode:
  build:
    context: .
    dockerfile: Dockerfile.secondary
  container_name: secondarynamenode
  hostname: secondarynamenode
  depends_on:
    - namenode
  environment:
    - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
  volumes:
    - ./secondarynamenode:/hadoop/dfs/secondary
  networks:
    - hadoop
```

```
datanode1:
  image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
  container_name: datanode1
  hostname: datanode1
  depends_on:
    - namenode
  environment:
    - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    - HDFS_CONF_dfs_datanode_hostname=datanode1
  volumes:
    - ./datanode1:/hadoop/dfs/data
  networks:
    - hadoop
```

```
datanode2:
  image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
  container_name: datanode2
  hostname: datanode2
  depends_on:
    - namenode
  environment:
    - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    - HDFS_CONF_dfs_datanode_hostname=datanode2
  volumes:
    - ./datanode2:/hadoop/dfs/data
  networks:
    - hadoop
```

```

datanode3:
  image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
  container_name: datanode3
  hostname: datanode3
  depends_on:
    - namenode
  environment:
    - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    - HDFS_CONF_dfs_datanode_hostname=datanode3
  volumes:
    - ./datanode3:/hadoop/dfs/data
  networks:
    - hadoop

networks:
  hadoop:
    driver: bridge

```

[Spark - docker-compose.yml]

```

version: "3"
services:
  spark-master:
    build: .
    container_name: spark-master
    environment:
      - SPARK_MODE=master
    ports:
      - "7077:7077"
      - "8082:8080"
    networks:
      - hadoop

  spark-worker-1:
    build: .
    container_name: spark-worker-1
    environment:
      - SPARK_MODE=worker
      - SPARK_MASTER_URL=spark://spark-master:7077
    networks:
      - hadoop

  spark-worker-2:
    build: .
    container_name: spark-worker-2
    environment:

```

```
- SPARK_MODE=worker
- SPARK_MASTER_URL=spark://spark-master:7077
networks:
  - hadoop
```

```
networks:
  hadoop:
    external: true
    name: hadoop_hadoop
```

#### [Zeppelin - docker-compose.yml]

```
version: '3.8'

services:
  zeppelin:
    image: apache/zeppelin:0.11.0
    container_name: zeppelin
    ports:
      - "18080:8080"
    volumes:
      - ./conf:/zeppelin/conf
      - ./notebook:/zeppelin/notebook
      - ./logs:/zeppelin/logs
      - /home/ubuntu/spark:/home/ubuntu/spark
    environment:
      - ZEPPELIN_LOG_DIR=/zeppelin/logs
    restart: unless-stopped
    networks:
      - hadoop

networks:
  hadoop:
    external: true
    name: hadoop_hadoop
```

#### [Zookeeper - docker-compose.yml]

```
version: '3.8'

services:
  zookeeper:
    image: zookeeper:3.7
    container_name: zookeeper
    restart: always
```

```

ports:
  - "2181:2181"
environment:
  ZOO_MY_ID: 1
  ZOO_PORT: 2181
  ZOO_SERVERS: server.1=zookeeper:2888:3888
volumes:
  - ./data:/data
  - ./datalog:/datalog

```

## 2 외부 서비스 정보

서비스	용도	설정 파일 위치	주요 키 / 설명
AWS S3	프로필 이미지 저장	<code>.env</code> , <code>application.yml</code>	<code>S3_BUCKET</code> , <code>S3_ACCESS</code> , <code>S3_SECRET</code>
JWT	인증 토큰 관리	<code>.env</code> , <code>application.yml</code>	<code>JWT_SECRET</code>
Kakao API	지역 기반 검색 등	<code>.env</code>	<code>KAKAO_API_KEY</code>
Naver News API	뉴스 검색 기능	<code>.env</code>	<code>client_id</code> , <code>client_secret</code>
Prometheus	메트릭 수집	<code>prometheus.yml</code> , backend	<code>/actuator/prometheus</code> 엔드포인트 필요
Grafana	메트릭 시각화 대시보드	Grafana Web UI	Prometheus에서 수집한 데이터 시각화
nginx-exporter	Nginx 상태 수집	<code>default.conf</code> , <code>prometheus.yml</code>	<code>/stub_status</code> 엔드포인트에서 상태 수집

## 3 DB 덤프 파일 최신본

- 파일 경로: `exec/db`
- 포함된 주요 SQL 파일

파일명	설명
<code>dump_crops.sql</code>	작물(crops) 테이블의 초기 데이터
<code>dump_crops_price.sql</code>	작물별 연간 가격 데이터
<code>dump_region.sql</code>	행정 구역 데이터 (시/도/군)
<code>dump_policy.sql</code>	귀농 정책 데이터
<code>dump_character_type.sql</code>	성격 유형 데이터
<code>dump_property.sql</code>	지역 속성(환경, 접근성 등)
<code>dump_region_crops.sql</code>	지역-작물 매핑 관계 데이터

- 복원 예시

```
psql -U ssafy -d farmbti_db < exec/db/dump_crops.sql
```

## 4 시연 시나리오

- 메인 페이지 접속 (<https://j12d209.p.ssafy.io>)

## 귀농의 첫걸음, 똑똑하게

귀농, 어디로 가야 할지 고민되시나요?  
빅데이터 분석을 통해 당신에게 딱 맞는 최적의 귀농 지역을 추천해 드립니다.  
지금 바로 클릭하고 새로운 시작을 준비해보세요!

### 주요 서비스

귀농인이 되기 위한 첫 걸음,  
어디가농과 함께 시작하세요.



가이드북



지역 추천



작물 계산기



매물



혜택

## 2. 회원가입 (상단 메뉴 → 회원가입 버튼 클릭, 기본 정보 입력 후 가입 완료)



### Signup

어디가농의 회원이 되어보세요

이메일

subi123@ssafy.com

비밀번호

.....

비밀번호 확인

.....

이름

황수비두밥

성별

☐ 남성

☒ 여성

생년월일

2020

1

1

주소

경북 구미시 도량동 259-4 (도량동)


주소 검색

회원가입

이미 계정이 있으신가요? [로그인](#)




### 3. 로그인



지역 추천 받기작물 수익 계산기귀농 멘토 찾기귀농 뉴스

로그인회원가입



## Login

로그인 후 다양한 서비스를 이용하세요

이메일

비밀번호

로그인

계정이 없으신가요? 회원가입

### 4. Farmbti 설문조사



## FARMbti 테스트

나는 어떤 유형의 농부일까?

내 성향에 맞는 귀농 지역이 궁금하다면?

16가지 유형으로 보는 나만의 귀농 성향 테스트!

시작하기



Q3. 학교나 교육시설이 가까운 지역을 얼마나 선호하시나요?

선호하지  
않음

1

2

3

4

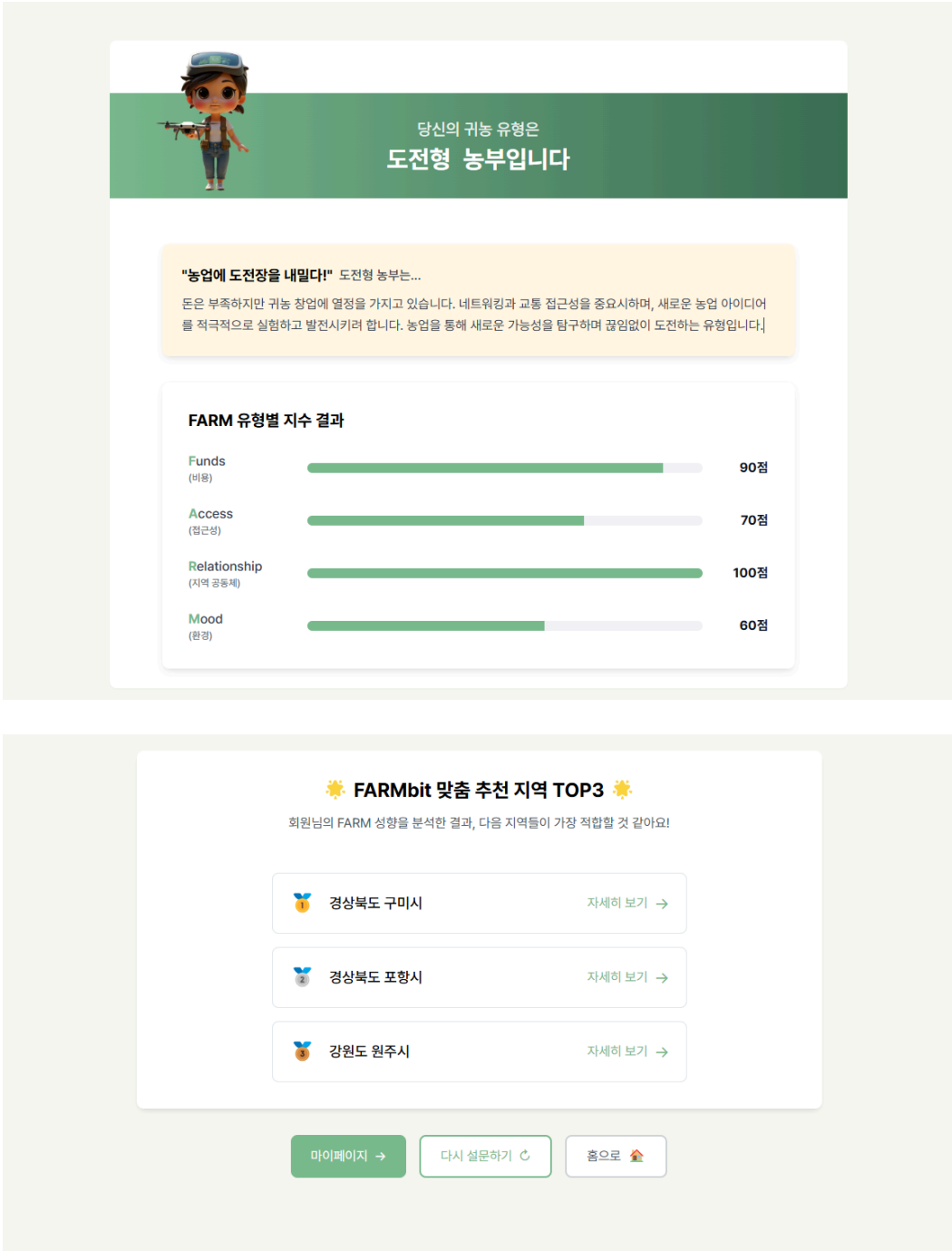
5

매우  
선호함

이전

다음

### 5. 맞춤형 귀농지 추천 결과 레포트



6. 작물 계산기 이용 (면적 입력 → 작물 선택 → 예상 수익 계산 결과 확인)

## 작물 수익 계산기

원하는 작물 재배에 따른 예상 수익을 계산해보세요.

1 평수 입력

2 작물 선택

3 결과 확인

### 재배 평수 입력

\*본 예상 수익 계산기는 농촌진흥청에서 제공된 <2023 지역별 농산물 소득 자료>와 농식품 빅데이터 거래소 <광역시도별 농산물 생산량 품목 산지별 물량 및 금액> 데이터를 기반으로 한 것으로, 실제 수익과 차이가 있을 수 있습니다. 따라서 참고용으로만 활용하시기 바랍니다.

총 경지 면적 =  평 × 3.3058m<sup>2</sup> = 3966.96 m<sup>2</sup> [확인](#)



사과



배



감귤



포도



수박



오이



대파



양파



상추



고구마



### 황공격님의 예상 수익 보고서

선택하신 작물은 **포도**입니다.

입력하신 재배 면적은 **1200 평 (3966.96 m<sup>2</sup>)**입니다.

1년 1기작 기준 **6128.95 Kg** 수확을 할 수 있을 것으로 예상됩니다.

### 연간 예상 작물 수익

\*본 수치는 노지 재배를 기준으로 합니다.

연간 총 매출액

42,216,428 원

-

연간 총 경영비

16,716,714 원

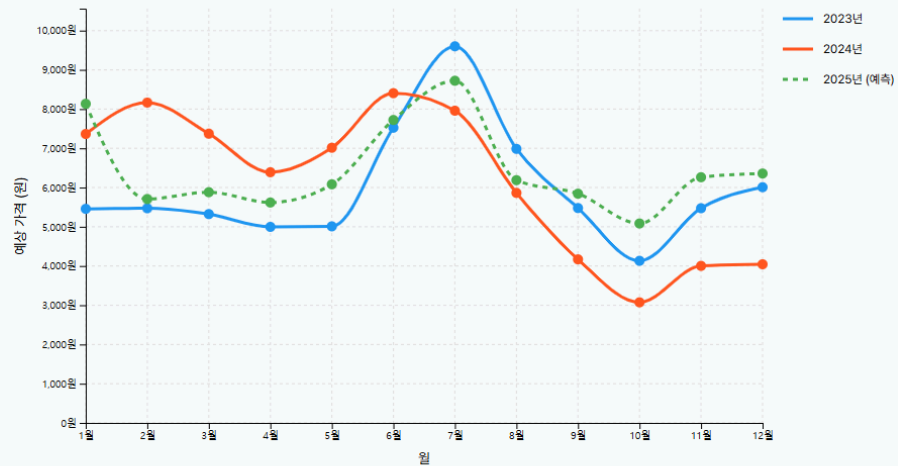
=

연간 예상 순 수익

25,499,714 원

총 매출액 대비 순 수익은 60.40%입니다.

포도의 월별 1Kg당 작물 매출액 추이 및 예측



\* 2023-2024년 데이터는 실제 가격, 2025년 데이터는 예측 가격입니다.

수익 계산 보고서 저장하기

다시 계산하기

## 7. 정책 조회 (귀농 정책 리스트 확인 - 지역 선택 가능)

## 지원 정책

관심있는 혜택을 찾아보세요.

지원 정책 목록

총 67개의 정책

도 선택

시/군/구 선택

필터 적용

## 운행경유차 배출가스 저감장치(DPF) 부착 지원사업

경상북도 예천군 2025.03.11

대기오염의 주된 원인인 미세먼지와 질소산화물 등을 저감하기 위하여 2025년 예천군 운행경유차 배출가스 저감장치(DPF) 부착 지원사업을 실시

## 2025년 하동군 신혼부부 주거자금 대출이자 지원사업

경상남도 하동군 2025.03.10

신혼부부 주거자금 대출이자 지원

## 2025년 하동군 첫만남 이용권

경상남도 하동군 2025.03.10

첫만남 이용권 지원

## 2025년 하동군 다동이 안전보험 지원

경상남도 하동군 2025.03.10

## 8. 뉴스 조회 (귀농 관련 네이버 뉴스 최신순으로 조회)

## 귀농 뉴스

귀농 관련 최신 소식과 정보를 확인하세요

### 최신 귀농 소식

총 100개의 뉴스

1	진도군, '진도에서 살아보기' 인기 만점	📅 2025-04-09
2	태백시, '귀농을 위한 태백 한 달 살기' 참여자 모집	📅 2025-04-09
3	신안군, 1004섬농업대학 입학식 개최	📅 2025-04-09
4	태백시, '태백 한 달 살기' 참여자 모집	📅 2025-04-09
5	신안군, 2025학년도 1004섬농업대학 입학식 개최	📅 2025-04-09
6	신안군, 2025학년도 1004섬농업대학 입학식 개최	📅 2025-04-09
7	충북 음성군 상상대로 음성에서 살아보기 프로그램 시작	📅 2025-04-08
8	신안군, '2025학년도 1004섬농업대학 입학식' 개최	📅 2025-04-08
9	음성군, 귀농·귀촌 희망 도시민 '상상대로 음성에서 살아보기' 진행	📅 2025-04-08
10	음성군, '상상대로 음성에서 살아보기' 4개월 여정 시작	📅 2025-04-08

### 9. 실시간 채팅 (귀농 멘토 찾기 → 지역 선택 → 해당 지역 멘토와 채팅 가능)



## 현장멘토 찾기

지역별 귀농 전문가를 찾아볼 수 있어요.  
귀농에 대한 장벽을 낮추고, 풍요로운 귀농 생활을 지원해드릴게요.



## 먼저 걸어온 길, 함께 걸어요

귀농에 대한 정보를 함께 나누고 싶다면, 멘토로 등록해 많은 멘티들과 귀농에 대한 이야기를 나누어 보세요!

[멘토 등록하기](#)



