

# 뇌출혈 조기 진단 프로젝트 - Version 3.0

## 버전 3.0 업데이트 내용

로그인/회원가입 시스템 추가 - JWT 기반 사용자 인증 기능 구현

## 시스템 개요

AI 기반 CT 영상 분석을 통한 뇌출혈 진단 보조 시스템입니다.

### 핵심 기능

기능	설명
뇌출혈 진단	ResNet50 기반 딥러닝 모델로 CT 이미지 분석
Grad-CAM 시각화	진단 근거를 히트맵으로 시각화
사용자 인증	JWT 기반 로그인/회원가입 (v3.0 신규)
웹 인터페이스	Streamlit 기반 사용자 친화적 UI

## Version 3.0 신규 기능

### 1. 로그인 시스템

- JWT(JSON Web Token) 기반 인증
- 세션 상태 유지 (페이지 새로고침 시에도 로그인 유지)
- 로그아웃 기능

### 2. 회원가입 시스템

- 아이디, 비밀번호, 이름, 이메일 입력
- 실시간 유효성 검사 (비밀번호 일치, 이메일 형식)
- 아이디/이메일 중복 확인
- BCrypt 비밀번호 암호화

### 3. Spring Boot 백엔드

- REST API 서버
- Spring Security 보안 설정
- MySQL 데이터베이스 연동
- CORS 설정 (Streamlit 연동)

# 기술 스택

## Frontend (Streamlit)

기술	버전	용도
Python	3.8+	메인 언어
Streamlit	1.28+	웹 UI 프레임워크
TensorFlow	2.x	딥러닝 모델
OpenCV	4.x	이미지 처리
Requests	2.x	API 통신

## Backend (Spring Boot)

기술	버전	용도
Java	17	메인 언어
Spring Boot	3.2.0	웹 프레임워크
Spring Security	6.x	보안/인증
Spring Data JPA	3.x	ORM
MySQL	8.x	데이터베이스
JWT	0.12.3	JWT 토큰 처리
Lombok	1.18+	코드 간소화

## 프로젝트 구조

```
뇌출혈/
├── backend/                # Spring Boot 백엔드
│   ├── pom.xml            # Maven 설정
│   └── src/main/
│       ├── java/com/brain/hemorrhage/
│       │   ├── BrainHemorrhageApplication.java
│       │   ├── config/      # Security, CORS 설정
│       │   ├── controller/  # 인증 API
│       │   ├── service/     # 인증 서비스
│       │   ├── repository/  # 사용자 Repository
│       │   ├── entity/      # 사용자 엔티티
│       │   ├── dto/         # 요청/응답 DTO
│       │   └── security/    # JWT 처리
│       ├── resources/
│       └── application.yml
├── Streamlit/              # Streamlit 프론트엔드
│   ├── brain_ct_improved.py # 메인 (로그인 + 진단)
│   ├── auth_utils.py       # 인증 유틸리티
│   └── pages/
│       └── signup.py        # 회원가입 페이지
├── model_files/            # 학습된 모델
└── Dataset/               # 데이터셋
```

# 설치 및 실행 방법

---

## 1. 사전 요구사항

- Java 17 이상
- Python 3.8 이상
- MySQL 8.x
- Spring Tool Suite (STS) 또는 IntelliJ

## 2. 데이터베이스 설정

MySQL에서 다음 명령 실행:

```
CREATE DATABASE brain_hemorrhage;
```

## 3. 백엔드 설정

1. application.yml 수정 (backend/src/main/resources/)
  - username: root
  - password: [본인 비밀번호]
2. STS에서 프로젝트 Import
  - File → Import → Maven → Existing Maven Projects
  - Root Directory: D:\뇌출혈\backend 선택
3. 서버 실행
  - BrainHemorrhageApplication.java 우클릭
  - Run As → Spring Boot App

## 4. 프론트엔드 실행

```
cd D:\뇌출혈\Streamlit
pip install requests
streamlit run brain_ct_improved.py
```

## 5. 접속

- 프론트엔드: <http://localhost:8501>
- 백엔드 API: <http://localhost:8080>

## API 명세

---

### 회원가입 - POST /api/auth/signup

Request Body:

```
{
  "username": "testuser",
  "password": "password123",
  "name": "홍길동",
  "email": "hong@example.com"
}
```

Response (성공):

```
{
  "success": true,
  "message": "회원가입이 완료되었습니다."
}
```

### 로그인 - POST /api/auth/login

Request Body:

```
{
  "username": "testuser",
  "password": "password123"
}
```

Response (성공):

```
{
  "success": true,
  "message": "로그인 성공",
  "token": "eyJhbGciOiJIUzI1NiIs... ",
  "username": "testuser",
  "name": "홍길동"
}
```

### 토큰 검증 - GET /api/auth/validate

Header: Authorization: Bearer {token}

Response:

```
{
  "success": true,
  "message": "토큰 유효",
  "username": "testuser",
  "name": "홍길동"
}
```

# 데이터베이스 설계

## users 테이블

컬럼명	타입	설명
id	BIGINT	기본키 (AUTO_INCREMENT)
username	VARCHAR(50)	로그인 아이디 (UNIQUE)
password	VARCHAR(255)	암호화된 비밀번호
name	VARCHAR(100)	사용자 이름
email	VARCHAR(100)	이메일 (UNIQUE)
created_at	TIMESTAMP	생성 시간
updated_at	TIMESTAMP	수정 시간

## 보안 설정

### JWT 토큰

- 알고리즘: HS256
- 유효기간: 1시간 (3600000ms)
- 비밀키: application.yml에서 설정

### 비밀번호 암호화

- 알고리즘: BCrypt
- 특징: Salt 자동 생성, 단방향 해시

### CORS 설정

- 허용 Origin: localhost:8501, localhost:8502, localhost:3000
- 허용 Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS

## 주의사항

### 의료 면책 조항

- 본 시스템은 연구 및 교육 목적으로 개발되었습니다.
- 실제 의료 진단을 대체할 수 없습니다.
- 최종 진단은 반드시 전문의가 수행해야 합니다.

### 개발 환경 주의

- application.yml의 비밀번호는 본인 환경에 맞게 수정하세요.
- JWT 비밀키는 운영 환경에서 반드시 변경하세요.
- 디버그 로그는 운영 환경에서 비활성화하세요.

## 버전 히스토리

버전	날짜	주요 변경사항
1.0	2025-11	초기 버전 (CNN 모델)
2.0	2025-11	ResNet50 Fine-tuning, Grad-CAM 추가
3.0	2025-12	로그인/회원가입 시스템 추가

마지막 업데이트: 2025-12-04