

# 실시간 채팅 애플리케이션

프로젝트명: 웹 기반 실시간 채팅 시스템

개발자: 김서진

학번: 202268006

개발 기간: 2025년 12월

사용 기술: PHP, MySQL, JavaScript

웹페이지 URL: <https://php-production-8f26.up.railway.app/>

## 프로젝트 개요

본 프로젝트는 PHP와 MySQL을 기반으로 한 웹 기반 실시간 채팅 애플리케이션입니다. 사용자 인증, 메시지 송수신, 온라인 상태 관리 등의 핵심 기능을 구현하였으며, AJAX polling 방식을 통해 실시간 통신을 구현하였습니다.

## 개발 목적

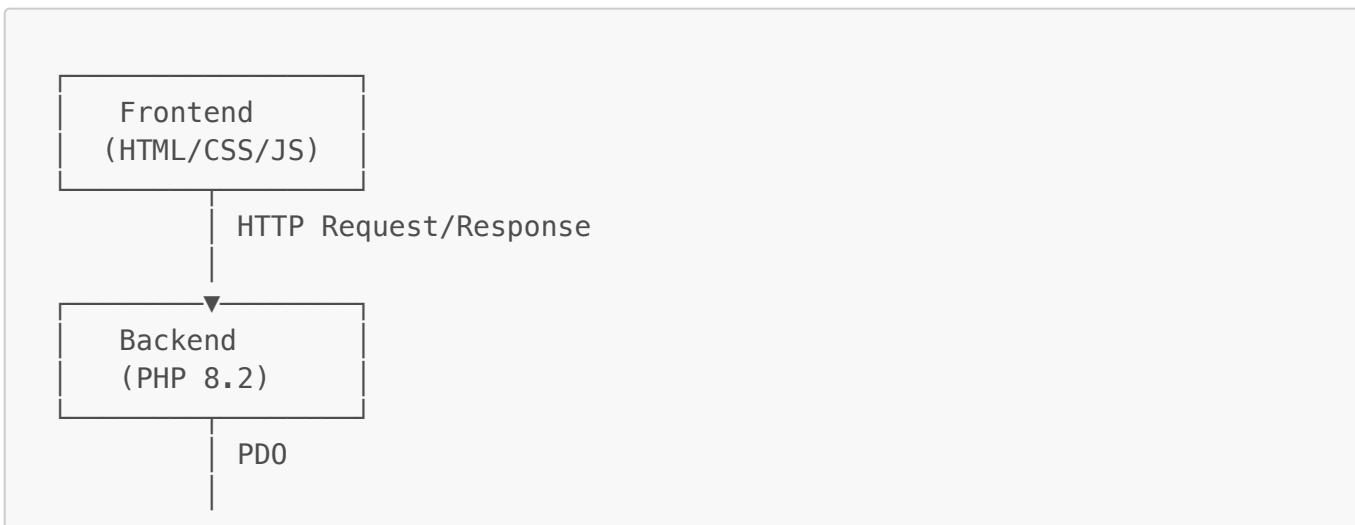
- 웹 프로그래밍 기술 (PHP, MySQL, JavaScript)의 실전 활용
- 실시간 데이터 통신 메커니즘의 이해 및 구현
- 보안을 고려한 사용자 인증 시스템 구축
- RESTful API 설계 및 구현 경험

## 주요 특징

- 사용자 인증 - 안전한 회원가입 및 로그인 시스템
- 실시간 메시징 - AJAX polling을 활용한 메시지 실시간 전송/수신
- 온라인 상태 관리 - 접속 중인 사용자 실시간 표시
- 반응형 UI - 모던하고 직관적인 사용자 인터페이스
- 보안 - SQL Injection, XSS 공격 방지

## 시스템 구성

### 시스템 아키텍처



Database  
(MySQL 5.7+)

## 기술 스택

구분	기술	버전	용도
Backend	PHP	8.2+	서버 사이드 로직 처리
Database	MySQL	5.7+	데이터 저장 및 관리
Frontend	HTML5/CSS3	-	사용자 인터페이스
Script	JavaScript (Vanilla)	ES6+	클라이언트 사이드 로직
통신	AJAX (Fetch API)	-	비동기 데이터 통신
보안	PDO, password_hash	-	SQL Injection, 비밀번호 보호

## 기능 명세

### 사용자 인증 기능

#### 회원가입 (register.php)

- 입력 검증 - 사용자명(3-20자), 비밀번호(4자 이상)
- 중복 체크 - 사용자명 중복 확인
- 비밀번호 암호화 - `password_hash()` 함수 사용
- 데이터 저장 - users 테이블에 안전하게 저장

#### 로그인 (login.php)

- 인증 검증 - 사용자명 및 비밀번호 확인
- 세션 관리 - PHP 세션을 통한 로그인 상태 유지
- 온라인 상태 업데이트 - 로그인 시 online\_status 테이블 갱신

#### 로그아웃 (logout.php)

- 세션 종료 - 세션 데이터 삭제
- 온라인 상태 제거 - online\_status 테이블에서 제거

### 채팅 기능

#### 메시지 전송

- 실시간 전송 - AJAX를 통한 비동기 전송
- 입력 검증 - 빈 메시지 방지, 최대 500자 제한
- XSS 방지 - 출력 시 HTML 이스케이핑

## 메시지 수신

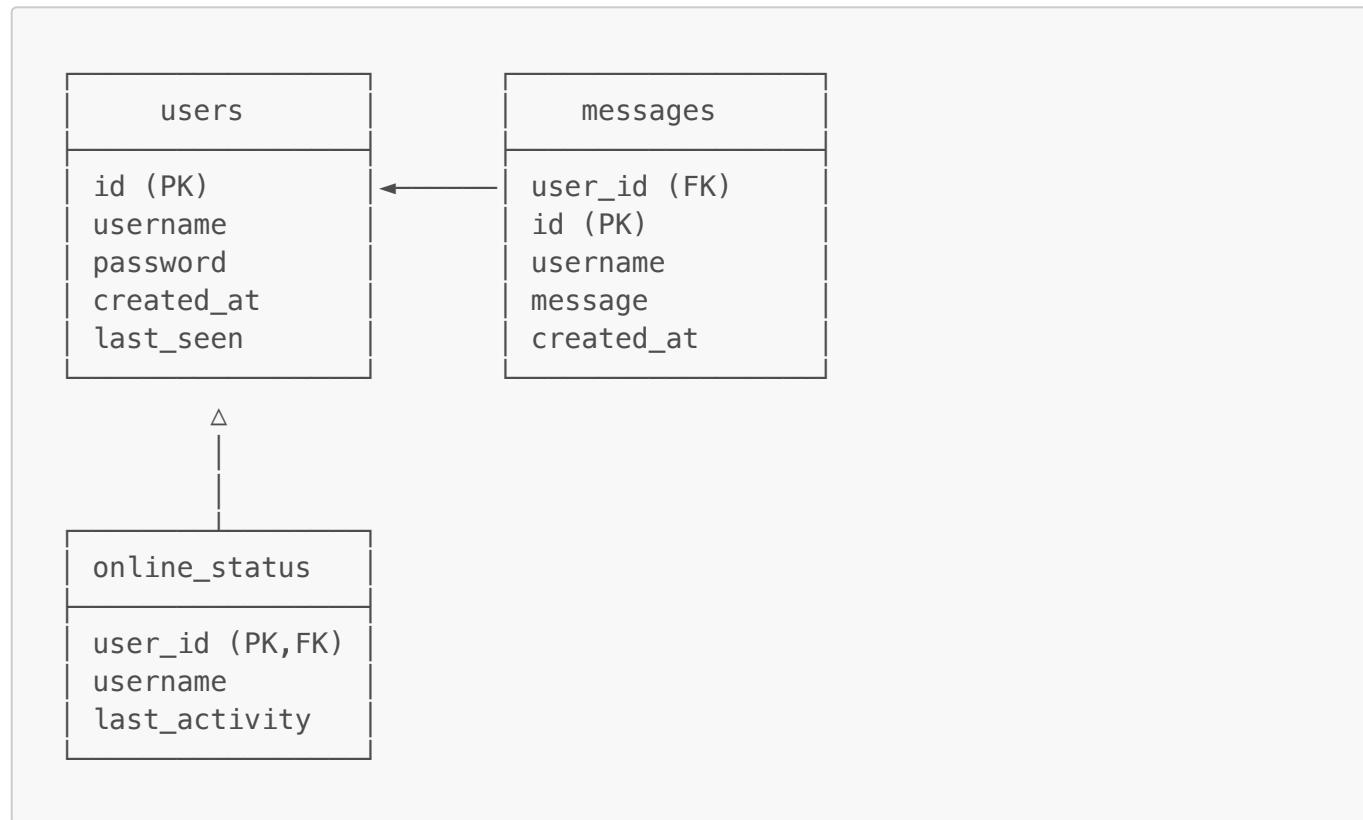
- **Polling 방식** - 1초마다 새 메시지 확인
- **증분 로딩** - 마지막 메시지 ID 이후의 메시지만 조회
- **자동 스크롤** - 새 메시지 수신 시 자동으로 하단 이동

## 온라인 사용자 관리

- **활동 추적** - 5초마다 사용자 활동 업데이트
- **온라인 판단** - 최근 30초 이내 활동 사용자를 온라인으로 표시
- **실시간 갱신** - 온라인 사용자 목록 5초마다 갱신

## 데이터베이스 설계

ERD (Entity Relationship Diagram)



## 테이블 상세 설계

### users 테이블

컬럼명	데이터 타입	제약조건	설명
id	INT	PRIMARY KEY, AUTO_INCREMENT	사용자 고유 ID
username	VARCHAR(50)	UNIQUE, NOT NULL	사용자명
password	VARCHAR(255)	NOT NULL	해시된 비밀번호
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	가입일시

컬럼명	데이터 타입	제약조건	설명
last_seen	TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	마지막 접속 시간

인덱스: username에 인덱스 설정으로 로그인 조회 성능 향상

### messages 테이블

컬럼명	데이터 타입	제약조건	설명
id	INT	PRIMARY KEY, AUTO_INCREMENT	메시지 고유 ID
user_id	INT	FOREIGN KEY, NOT NULL	발신자 ID
username	VARCHAR(50)	NOT NULL	발신자명
message	TEXT	NOT NULL	메시지 내용
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	작성 시간

인덱스: created\_at에 인덱스 설정으로 시간순 조회 성능 향상 외래키: user\_id → users.id (ON DELETE CASCADE)

### online\_status 테이블

컬럼명	데이터 타입	제약조건	설명
user_id	INT	PRIMARY KEY, FOREIGN KEY	사용자 ID
username	VARCHAR(50)	NOT NULL	사용자명
last_activity	TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	마지막 활동 시간

외래키: user\_id → users.id (ON DELETE CASCADE)

## 파일 구조

```

php/
├── config.php          # 데이터베이스 연결 및 공통 설정
├── database.sql         # 데이터베이스 스키마 정의
├── index.php            # 메인 페이지 (자동 리다이렉션)
├── login.php             # 로그인 페이지
├── register.php          # 회원가입 페이지
├── chat.php              # 채팅 메인 페이지
├── api.php                # REST API 엔드포인트
├── logout.php             # 로그아웃 처리
├── composer.json          # PHP 의존성 관리 (Railway 배포용)
├── nixpacks.toml          # Railway 빌드 설정
├── RAILWAY_DEPLOY.md      # Railway 배포 가이드
└── README.md               # 프로젝트 문서

```

## API 엔드포인트

메서드	엔드포인트	파라미터	기능
GET	api.php?action=load	-	초기 메시지 로드 (최근 50개)
GET	api.php?action=get	last_id	특정 ID 이후 새 메시지 조회
POST	api.php?action=send	message	메시지 전송
POST	api.php?action=update_online	-	온라인 상태 갱신
GET	api.php?action=online_users	-	온라인 사용자 목록 조회

## 보안 고려사항

### SQL Injection 방지

**PDO Prepared Statements 사용** - 모든 데이터베이스 쿼리에 적용

```
// 안전한 쿼리 실행 예시
$stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
$stmt->execute([$username]);
```

### XSS (Cross-Site Scripting) 방지

출력 이스케이핑 - `htmlspecialchars()` 함수 사용

```
function sanitize($data) {
    return htmlspecialchars(trim($data), ENT_QUOTES, 'UTF-8');
}
```

## 비밀번호 보안

해싱 알고리즘 - `password_hash()` 함수 사용 (bcrypt)

```
// 회원가입 시
$hashed = password_hash($password, PASSWORD_DEFAULT);

// 로그인 시
if (password_verify($password, $user['password'])) {
    // 인증 성공
}
```

## 세션 관리

- 세션 기반 인증 - 로그인 상태를 서버 세션으로 관리

- 접근 제어 - 미인증 사용자의 채팅 페이지 접근 차단
- 

## 설치 및 실행 방법

### 시스템 요구사항

- PHP 8.1 이상
- MySQL 5.7 이상
- 웹 브라우저 (Chrome, Firefox, Safari 등)

### 로컬 설치

#### 1단계: 데이터베이스 생성

```
# 데이터베이스 생성
mysql -u root -e "CREATE DATABASE IF NOT EXISTS chat_app CHARACTER SET
utf8mb4 COLLATE utf8mb4_unicode_ci;"

# 테이블 생성
mysql -u root chat_app < database.sql

# 확인
mysql -u root -e "USE chat_app; SHOW TABLES;"
```

#### 2단계: 설정 파일 확인

config.php 파일에서 데이터베이스 접속 정보 확인:

```
define('DB_HOST', '127.0.0.1');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'chat_app');
```

#### 3단계: 웹 서버 실행

```
# PHP 내장 서버 사용
php -S localhost:8000

# 또는 XAMPP PHP 사용 (Mac)
/Applications/XAMPP/xamppfiles/bin/php -S localhost:8000
```

#### 4단계: 애플리케이션 접속

브라우저에서 <http://localhost:8000> 접속

# Railway 배포

## 배포 방법

자세한 Railway 배포 가이드는 [RAILWAY\\_DEPLOY.md](#) 파일을 참고하세요.

## 간단 요약

### 1. Railway 프로젝트 생성

- GitHub 저장소 연결

### 2. MySQL 데이터베이스 추가

- Railway 대시보드에서 MySQL 추가
- 환경변수 자동 설정

### 3. 자동 배포

- Git push 시 자동으로 빌드 및 배포

### 4. 데이터베이스 테이블 생성

- 배포 후 브라우저에서 [/setup\\_database.php](#) 접속
- 테이블 자동 생성

## 환경변수

Railway는 다음 환경변수를 자동으로 설정합니다:

- MYSQL\_PUBLIC\_HOST** - MySQL 호스트
- MYSQL\_PUBLIC\_PORT** - MySQL 포트
- MYSQLUSER** - 데이터베이스 사용자
- MYSQLPASSWORD** - 데이터베이스 비밀번호
- MYSQLDATABASE** - 데이터베이스 이름

## 테스트 및 검증

### 기능 테스트

기능	테스트 항목	결과
회원가입	정상 가입	✓
회원가입	중복 사용자명 방지	✓
회원가입	입력 검증 (최소 길이)	✓
로그인	정상 로그인	✓
로그인	잘못된 비밀번호 거부	✓

기능	테스트 항목	결과
메시지 전송	실시간 전송	✓
메시지 수신	1초 간격 polling	✓
온라인 상태	실시간 갱신	✓
로그아웃	세션 종료 및 상태 제거	✓

## 보안 테스트

공격 유형	방어 기법	검증
SQL Injection	PDO Prepared Statements	✓
XSS	htmlspecialchars()	✓
CSRF	세션 기반 인증	✓
비밀번호 노출	password_hash()	✓

## 문제 해결

### XAMPP MySQL 실행 오류

**증상:** XAMPP에서 MySQL이 시작되지 않음

**원인:** 다른 MySQL 서비스가 3306 포트 사용 중

**해결:**

```
# 포트 사용 확인
lsof -i :3306

# 기존 MySQL 중지
sudo /usr/local/mysql/support-files/mysql.server stop
```

### 데이터베이스 연결 오류

**증상:** SQLSTATE[HY000] [2002] No such file or directory

**해결:** config.php에서 localhost 대신 127.0.0.1 사용

```
define('DB_HOST', '127.0.0.1'); // localhost 대신 사용
```

## 성능 최적화

## 데이터베이스 최적화

- **인덱싱** - username, created\_at 컬럼에 인덱스 설정
- **외래키 제약** - CASCADE 옵션으로 데이터 무결성 유지
- **문자 인코딩** - utf8mb4로 이모지 및 다국어 지원

## 네트워크 최적화

- **증분 로딩** - 마지막 메시지 ID 이후만 조회
  - **LIMIT 절** - 초기 로드 시 최근 50개 메시지만 조회
  - **Polling 간격** - 적절한 간격(1초) 설정으로 서버 부하 최소화
- 

## 향후 개선 계획

### 기술적 개선

- **WebSocket 도입** - Socket.IO를 활용한 진정한 실시간 통신
- **파일 업로드** - 이미지 및 파일 공유 기능
- **메시지 검색** - 전체 텍스트 검색 기능 구현
- **읽음 표시** - 메시지 읽음/안읽음 상태 표시

### 기능적 개선

- **개인 메시지(DM)** - 1:1 채팅 기능
- **채팅방 생성** - 다중 채팅방 지원
- **사용자 프로필** - 프로필 사진 및 상태 메시지
- **알림** - 새 메시지 알림 기능
- **이모지 지원** - 이모지 선택기 추가

### UI/UX 개선

- **다크 모드** - 테마 전환 기능
  - **모바일 최적화** - PWA(Progressive Web App) 전환
  - **접근성** - ARIA 레이블 및 키보드 내비게이션 개선
- 

## 참고자료

### 공식 문서

- [PHP Manual](#)
- [MySQL Documentation](#)
- [MDN Web Docs - JavaScript](#)

### 보안 가이드

- [OWASP Top 10](#)
- [PHP Security Best Practices](#)

### 배포 가이드

- [Railway Documentation](#)
  - [Nixpacks PHP Guide](#)
- 

## 라이선스

MIT License

---

개발 완료일: 2025년 12월 14일

프로젝트 저장소: [GitHub](#)

웹페이지 URL: [Railway App](#)