

Week 2 Report

1. Commits on Dec 15, 2025

- Kimheng:
 - config build.gradle

```

21 21 dependencies {
22 - implementation 'org.springframework.boot:spring-boot-starter'
23 - implementation 'org.flywaydb:flyway-mysql'
24 - implementation 'org.springframework.boot:spring-boot-starter-jdbc'
25 - testImplementation 'org.springframework.boot:spring-boot-starter-test'
26 - testRuntimeOnly 'org.junit.platform:junit-platform-launcher'
27 + implementation("org.springframework.boot:spring-boot-starter:4.0.0")
28 + testImplementation("org.springframework.boot:spring-boot-starter-test:4.0.0")
29 + testRuntimeOnly("org.junit.platform:junit-platform-launcher")
30 +
31 + implementation("org.springframework.boot:spring-boot-starter-web:4.0.0")
32 + testImplementation("org.springframework.boot:spring-boot-starter-test:4.0.0")
33 + testRuntimeOnly("org.junit.platform:junit-platform-launcher")
34 +
35 + implementation("org.springframework.boot:spring-boot-starter-thymeleaf:4.0.0")
36 + implementation("org.springframework.boot:spring-boot-starter-actuator:4.0.0")
37 + implementation("org.springframework.boot:spring-boot-starter-data-jpa:4.0.0")
38 + implementation("org.springframework.boot:spring-boot-starter-validation:4.0.0")
39 +
40 + implementation("com.mysql:mysql-connector-j:8.3.0")
41 +
42 + implementation("org.springframework.boot:spring-boot-starter-jdbc:4.0.0")
43 + implementation("org.flywaydb:flyway-core:11.0.1")
44 + implementation("org.flywaydb:flyway-mysql:11.0.1")
45 }
  
```

2. Commits on Dec 16, 2025

- Kimheng:
 - Adding "FlywayConifg.java"
 - stop using `spring.jpa.hibernate.ddl-auto=none` and organize the `src/main/resources/application.properties` file into sections

```

2 2
3 - # db variables
4 + # Database
5 + #
6 + spring.datasource.url=jdbc:mysql://localhost:3306/aletheia?createDatabaseIfNotExist=true
7 spring.datasource.username=Aletheia
8 spring.datasource.password=killion123
9 - spring.datasource.url=jdbc:mysql://localhost:3306/aletheia
10 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
11 + # JPA/Hibernate configuration - disable auto-schema creation since we're using Flyway
12 - spring.jpa.hibernate.ddl-auto=none
13 + #
14 + # JPA / Hibernate
15 + #
16 + spring.jpa.hibernate.ddl-auto=none
17 + spring.jpa.show-sql=true
18 + spring.jpa.properties.hibernate.format_sql=true
19 + spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
20 +
21 + # Flyway
22 + #
23 + spring.flyway.enabled=true
24 + spring.flyway.locations=classpath:db/migration
25 + spring.flyway.baseline-on-migrate=true
26 + spring.flyway.baseline-version=0
27 + spring.flyway.baseline-description=Initial
28 + #
29 + # Logging
30 + #
31 + logging.level.org.flywaydb=DEBUG
  
```

- Changed some syntax inside `src/main/resources/db/migration/V1__first_db_design.sql` Example From `user`

to users

- Setha:
 - Accepting pull request to merge with main branch from Kimheng

3. Commits on Dec 17, 2025

- Pheaktra:
 - Add security, Register and Login logic by using the concepts we learned from our TP lessons
 - Adding springboot security and jwt dependencies in `build.gradle`
 - The files created are:
 - AuthRepsonse.java
 - AuthenticationController.java
 - CorsConfig.java
 - CustomUserDetailsService.java
 - JwtFilter.java
 - JwtProperties.java
 - JwtUtil.java
 - LoginRequest.java
 - LoginResponse.java
 - SecurityConfig.java
 - UserEntity.java
 - UserRepository.java
 - Adding security properties

```

src/main/resources/application.properties
@@ -29,3 +29,9 @@ spring.flyway.baseline-description=Initial
29 29 # Logging
30 30 # =====
31 31 logging.level.org.flywaydb=DEBUG
32 +
33 + # =====
34 + # Security
35 + # =====
36 + app.jwt.secret=U2FtcGx1U2VjdXJ1S2V5QmFzZTY0MTIzNDU2Nzg5MDE=
37 + app.jwt.exp=3600000
  
```

4. Commits on Dec 18, 2025

- Kity:
 - organize the src folders structure from everything inside the `src/main/java/aletheia/project/Aletheia/` folder into multiple folders for thier specific purposes, like config folder, controller folder, dto folder, etc.
 - Creating `WebContoller.java` and UI for "dashboard", "login", and "signup" pages.
- Setha:
 - Accepting pull request to merge from Pheaktra.

- Editing README.md file to show the project objectives.
- Editing `Report/Aletheia_Week1_Report.md` to show week 1 progress in simple sentences.
- Accepting pull request to merge from Kity (He accidentally delete the report folder because he thought it might be the causes for build error).
- Adding Report folders back alongside week2 report

5. Commits on Dec 21, 2025

- Setha:
 - Edited Week 2 report to add Pheaktra and Kity activity on the report
- Kimheng:
 - Added his own commit on week 2 report progress
 - Created Diagram folder inside the report folder which included `activty_diagram.puml`, `login-register_activity_diagram.puml`, and `use_case.puml`.
 - Created `PaperEntity.java` file

6. Commits on Dec 22, 2025

- Kimheng:
 - feat: Add paper submission system with entity, service, controller and UI
 - Create complete V1 migration with papers table (BIGINT id, abstract_text, file_name, created_at)
 - Implement PaperEntity, PaperRepository, and PaperService
 - Build PaperController with paper creation endpoints
 - fix files ReviewRepository and AssignmentRepository reviewerID -> reviewer.id
- Phayuk:
 - created repositories, entities, and drafted controller (no logic yet) for
 - Assignment
 - CoAuthor
 - CoAuthorId
 - Paper (This one conflicted with Kimheng commit)
 - Review
- Setha:
 - Accepted merge pull request from Kimheng
 - Fixing conflict between Kimheng and Phayuk commit (Both of them done on Paper repositories, entities, and controller), and i choose kimheng commit as Kimheng had finished All of paper controller, entities and repositories
 - Accepted merge pull request from Phayuk
 - Accepted merge pull request from Kity
- Kity:
 - Fix securityConfig bug

```
.authenticationProvider(authProvider())
.authorizeHttpRequests(auth -> auth
    // 1. Only allow these without login
    .requestMatchers("/login",
"/register").permitAll()
    .requestMatchers("/css/**", "/js/**",
"/images/**").permitAll()
    // 2. Everything else requires authentication
    .anyRequest().authenticated()
```

to

```
.authenticationProvider(authProvider())
.authorizeHttpRequests(auth -> auth
    // 1. Only allow these without login
    .requestMatchers("/login", "/register", "/login-
process", "/register-process").permitAll()
    .requestMatchers("/css/**", "/js/**",
"/images/**").permitAll()
    // 2. Everything else requires authentication
    .anyRequest().authenticated()
```