



ព្រះរាជាណាចក្រកម្ពុជា
ជាតិ សាសនា ព្រះមហាក្សត្រ

Software Engineering

Project : Student Internship Management System

Group: B2

Name of Student	ID of Student	Score
SAM NANGALEX	e20220253
PENG SEYHA	e20220620
OUN RITHI	e20220420
SOEUN SEREYVATH	e20220907
YI MONIROM	e20220175

Lecturer: Roeun Pacharoth

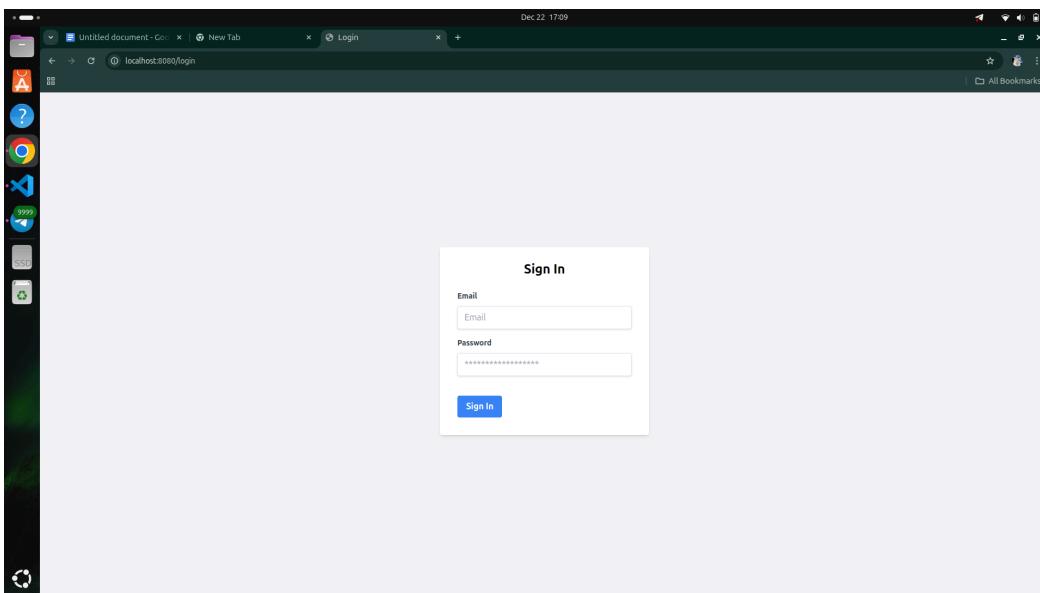
Academic year 2025-2026

Progress Report 1

Name: SAM NANGALEX

- Create login page UI

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org" >
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://cdn.tailwindcss.com"></script>
    <title>Login</title>
</head>
<body class="bg-gray-100 flex items-center justify-center h-screen">
    <div class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4 w-full max-w-sm">
        <h2 class="text-center text-2xl font-bold mb-6">Sign In</h2>
        <div class="text-red-500 text-sm mb-4" th:action = "@{error}">
            <span th:text="${error}"></span>
        </div>
        <form th:action = "@{/login}" method="post">
            <div class="mb-4">
                <label class="block text-gray-700 text-sm font-bold mb-2" for="email">
                    Email
                </label>
                <input
                    class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 focus:outline-none focus:ring-2 focus:ring-blue-500"
                    name="email" type="email" placeholder="Email" required>
            </div>
            <div class="mb-6">
                <label class="block text-gray-700 text-sm font-bold mb-2" for="password">
                    Password
                </label>
                <input
                    class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 mb-3 focus:outline-none focus:ring-2 focus:ring-blue-500"
                    name="password" type="password" placeholder="*****" required>
                <p class="text-red-500 text-xs italic hidden" id="password-error">Please enter a password.</p>
            </div>
            <div class="flex items-center justify-between">
                <button
                    class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded"
                    type="submit">
                    Sign In
                </button>
                <!-- <a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-700" href="#">
                    Forgot Password?
                </a> -->
            </div>
        </form>
    </div>
</body>
```



- Spring Security Configuration:

- Create SecurityConfig class

- Configure authentication manager
- Configure password encoder (BCrypt)
- Set up login URL and login processing URL

```

demo > src > main > java > project > demo > security > SecurityConfig.java > Language Support for Java(TM) by Red Hat > SecurityConfig.java
1 package project.demo.security;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
6 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
7 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
8 import org.springframework.security.crypto.password.PasswordEncoder;
9 import org.springframework.security.web.SecurityFilterChain;
10
11
12 @EnableMethodSecurity
13     .BCryptPasswordEncoder DefaultSecurityFilterChain
14 @Configuration
15 public class SecurityConfig{
16
17     @Bean
18     public PasswordEncoder passwordEncoder(){
19         return new BCryptPasswordEncoder();
20     }
21
22     @Bean
23     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception{
24         http
25             .authorizeHttpRequests(auth ->
26                 auth.requestMatchers(...patterns: "/login").permitAll()
27                 // .requestMatchers("/supervisor/").hasRole("SUPERVISOR")
28                 .anyRequest().authenticated()
29             )
30             .formLogin(form -> form
31                 .loginPage(loginPage: "/login")
32                 .loginProcessingUrl(loginProcessingUrl: "/login")
33                 .usernameParameter(usernameParameter: "email")
34                 .passwordParameter(passwordParameter: "password")
35                 .defaultSuccessUrl([defaultSuccessUrl: "/home", alwaysUse: true])
36                 .permitAll()
37             );
38
39     }
40 }
```

- Database:

- Create user table
- Create user_role linking table
- Create student table
- Add foreign key constraints
- Insert default roles

```

demo > src > main > resources > db > migrations > V1__create_all_tables.sql
1  create table users (
2    id int auto_increment primary key,
3    email varchar(128) unique not null,
4    username varchar(64),
5    password varchar(128) not null,
6    enable boolean default true,
7    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
8  );
9
10 create table roles(
11   id int auto_increment primary key,
12   role_name varchar(64) unique not null
13 );
14
15 create table user_role(
16   user_id int,
17   role_id int,
18   primary key (user_id, role_id)
19 );
20
21 create table students(
22   id int auto_increment primary key,
23   full_name varchar(64) not null,
24   phone_number int unique,
25   email varchar(128) unique,
26   year int,
27   major varchar(64),
28   user_id int
29 );
30

```

```

src > main > resources > db > migrations > V2__add_foreign_keys.sql
1 alter table user_role
2 add constraint user_role(userID foreign key (user_id) references users(id)
3 on update cascade on delete restrict;
4
5 alter table user_role
6 add constraint user_role(roleID foreign key (role_id) references roles(id)
7 on update cascade on delete restrict;
8
9 alter table students
10 add constraint students_userID foreign key (user_id) references users(id)
11 on update cascade on delete restrict;
12

```

Name: OUN RHITI

- Create company controller
- Validate input (name, location, type)

```

o > src > main > java > project > demo > controller > CompanyController.java > Language Support for Java(TM) by Red Hat > CompanyController > companyRepository
package project.demo.controller;

import java.util.List;

import org.springframework.web.bind.annotation.*;

import project.demo.entity.Company;
import project.demo.repository.CompanyRepository;

... CompanyRepository
@RestController
@RequestMapping("/api/companies")
public class CompanyController {

    private final CompanyRepository companyRepository;

    ... CompanyRepository
    public CompanyController(CompanyRepository companyRepository) {
        this.companyRepository = companyRepository;
    }
}

```

● Database

- Create company table
- Create document table
- Add foreign keys (user_id / company_id)
- Add indexing (for fast search)
- Add seed data for testing

```

CREATE TABLE company (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(100),
    phone_number int
);

CREATE TABLE document (
    id int PRIMARY KEY AUTO_INCREMENT,
    student_id int,
    internship_app_id int,
    file_name VARCHAR(255) NOT NULL,
    file_path VARCHAR(255) NOT NULL,
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```

alter table internship_application
add constraint internship_app_company_id foreign key (company_id) references company(id)
on update cascade on delete restrict;

alter table document
add constraint document_student_id foreign key (student_id) references students(id)
on update cascade on delete restrict;

```

Name: PENG SEYHA

● Backend

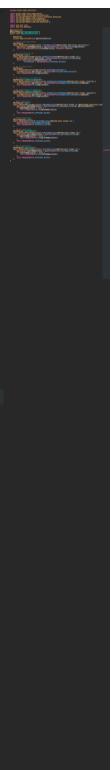
- Create Application model
- Create Application controller/service

Create endpoints:

- Create application
- Read application(s)
- Update application
- Delete application

```

1 import project.demo.service.ApplicationService;
2 import org.springframework.beans.factory.annotation.Autowired;
3 import org.springframework.http.HttpStatus;
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.web.bind.annotation.*;
6
7 import java.util.List;
8 import java.util.Optional;
9
10 ...
11 ApplicationService
12 @RestController
13 @RequestMapping("/api/applications")
14 public class ApplicationController {
15
16     ...
17     @Autowired
18     private ApplicationService applicationService;
19
20     // Create a new application
21     ...
22     @PostMapping
23     public ResponseEntity<Application> createApplication(@RequestBody Application application) {
24         Application createdApplication = applicationService.createApplication(application);
25         return new ResponseEntity<(createdApplication, HttpStatus.CREATED);
26     }
27
28     // Get application by ID
29     ...
30     @GetMapping("/{id}")
31     public ResponseEntity<Application> getApplicationById(@PathVariable Integer id) {
32         Optional<Application> application = applicationService.getApplicationById(id);
33         return application.map(ResponseEntity::ok)
34             .orElseGet(() -> ResponseEntity.notFound().build());
35     }
36
37     // Get all applications
38     ...
39     @GetMapping
40     public ResponseEntity<List<Application>> getAllApplications() {
41         List<Application> applications = applicationService.getAllApplications();
42         return ResponseEntity.ok(applications);
43     }
44
45     // Get applications by student ID
46     ...
47     @GetMapping("student/{studentId}")
48     public ResponseEntity<List<Application>> getApplicationsByStudent(@PathVariable Integer studentId) {
49         List<Application> applications = applicationService.getApplicationsByStudent(studentId);
50         return ResponseEntity.ok(applications);
51     }
52 }
```



Name: YI MONIROM

- Create middleware to check if user role = supervisor

```

1 package project.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7
8
9 @Controller
10 public class SupervisorController {
11     http://127.0.0.1:8080/viwe_Applications_Status
12     @GetMapping("/viwe_Applications_Status")
13     public String viwe_Applications_Status(Model model) {
14         return "here you are!";
15     }
16 }

```

```

@Configuration
public class SecurityConfig{

    @Bean
    public PasswordEncoder passwordEncoder(){
        return new BCryptPasswordEncoder();
    }
    ↳ 1 bean | ↳ 1 bean
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception{
        http
            .authorizeHttpRequests(auth ->
                auth.requestMatchers(...patterns: "/login").permitAll()
                .requestMatchers(...patterns: "/supervisor/").hasRole(role: "/SUPERVISOR")
                .anyRequest().authenticated()
            )
            .formLogin(form -> form
                .LoginPage(loginPage: "/login")
                .loginProcessingUrl(loginProcessingUrl: "/login")

```

- Create supervisor model
- Create supervisor controller

Name: SOEUN SEREVATH

- SL ke mnak eg