

PERSONAL FINANCE MANAGEMENT SYSTEM

The **Personal Finance Management Application** is a web-based system designed to help users manage their income, expenses, and savings effectively. The application allows users to track their financial activities, set budget goals, and generate reports for better financial planning.

Key Features:

User Authentication: Secure login and registration using **Spring Security** and **JWT** authentication.

Income & Expense Tracking: Users can record their earnings and spending.

Savings Management: Allows users to set and monitor savings goals.

Financial Reports: Provides insights into spending patterns and budget planning.

Database Integration: Uses **MySQL/PostgreSQL** for data storage with **Spring Data JPA**.

Secure API: RESTful API built with **Spring Boot** for seamless frontend integration.

Deployment: Can be deployed using **Docker** for containerization and hosted on cloud platforms.

CODING:

```
package com.example.finance_management;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.web.bind.annotation.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.data.jpa.repository.JpaRepository;
import javax.persistence.*;
import java.util.*;
```

@SpringBootApplication

```
public class FinanceManagementApplication {  
    public static void main(String[] args) {  
        SpringApplication.run(FinanceManagementApplication.class, args);  
    }  
}
```

@Bean

```
public PasswordEncoder passwordEncoder() {  
    return new BCryptPasswordEncoder();  
}  
}
```

@RestController

@RequestMapping("/api/users")

```
class UserController {  
    @Autowired  
    private UserRepository userRepository;  
    @PostMapping("/register")  
    public User registerUser(@RequestBody User user) {  
        user.setPassword(new BCryptPasswordEncoder().encode(user.getPassword()));  
        return userRepository.save(user);  
    }  
    @GetMapping("/")  
    public List<User> getAllUsers() {  
        return userRepository.findAll();  
    }  
}
```

@Entity

```

class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String username;

    private String password;

    private String role;

}

```

```

interface UserRepository extends JpaRepository<User, Long> {}

```

```

@EnableWebSecurity

```

```

class SecurityConfig {

```

```

    @Bean

```

```

    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

```

```

        http

```

```

            .csrf().disable()

```

```

            .authorizeHttpRequests(auth -> auth.antMatchers("/api/users/register").permitAll()

```

```

                .anyRequest().authenticated())

```

```

            .sessionManagement(sess -> sess.sessionCreationPolicy(SessionCreationPolicy.STATELESS));

```

```

        return http.build();

```

```

    }

```

```

}

```

```

@RestController

```

```

@RequestMapping("/api/finance")

```

```

class FinanceController {

```

```

    @Autowired

```

```

    private FinanceRepository financeRepository;

```

```

    @PostMapping("/add")

```

```

    public FinanceRecord addFinanceRecord(@RequestBody FinanceRecord record) {

```

```

        return financeRepository.save(record);
    }

    @GetMapping("/")
    public List<FinanceRecord> getAllFinanceRecords() {
        return financeRepository.findAll();
    }
}

@Entity
class FinanceRecord {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String userId;

    private double income;

    private double expense;

    private double savings;
}

interface FinanceRepository extends JpaRepository<FinanceRecord, Long> {}

```