

Financial management system using Spring Web with Lombok and verification dependencies.
This system manages users information and financial accounts for them.

Create: FinanceModel ,User Model and FinanceController.

User Model has: id,userName,email,password,address,phoneNumber,dateofbirth.

Finance Model has: id,accountBalance,currency,accountType,creationDate,
bankaccountnumber, creditcardNumber.

- Id:
 - Cannot be null.
 - length must be more than 2 characters.

@NotNull(message = "id cant be null")

@Size(min = 3,message = "length must be more than 2 characters")

private String id;

- Username:
 - Cannot be null.
 - Length must be between 5 and 20 characters.

@NotNull(message = "Username cannot be null")

@Size(min = 5, max = 20, message = "Username length must be between 5 and 20 characters")

private String username;

- Password:
 - Cannot be null.
 - Should be at least 8 characters long.
 - Must include a combination of uppercase and lowercase letters, numbers, and special characters.

@NotNull(message = "Password cannot be null")

@Size(min = 8, message = "Password length must be at least 8 characters")

@Pattern(regex = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@\$!%*?&])[A-Za-z\\d@\$!%*?&]+\$",
message = "Password must contain at least one uppercase letter, one lowercase letter, one
number, and one special character")

private String password;

- Email :
 - Cannot be null.
 - Must be in a valid email format.

@NotNull(message = "Email address cannot be null")

@Email

private String email;

- Date of Birth:
 - Must be provided.
 - Should be in the past.

@NotNull(message = "Date of birth cannot be null")

@Past(message = "Date of birth must be in the past")

private Date dateOfBirth;

- Address:
 - Must be provided.
 - Length should not exceed 100 characters.

@NotNull(message = "Address cannot be null")

@Size(max = 100, message = "Address length cannot exceed 100 characters")

private String address;

- Phone Number:
 - Must start with "05"
 - Must consists of exactly 10 digits.

@NotNull(message = "cant be null")

@Pattern(regexp = "^05\\d{8}\$", message = "Must start with '05' and consists of exactly 10 digits")

private String phonenumber;

- Bank Account Number:
 - must be between 9 and 18 digits

@Pattern(regexp = "\\d{9,18}", message = "Bank account number must be between 9 and 18 digits")

private String bankAccountNumber;

- Credit Card Number:
 - Should be valid and unique if provided.

@CreditCardNumber(message = "Invalid credit card number")

private String creditCardNumber;

- AccountBalance:
 - Cannot be null.

@NotNull(message = "Account balance cannot be null")

private BigDecimal accountBalance;

- Currency:
 - cannot be null.

@NotNull(message = "Currency cannot be null")

private String currency;

- AccountType:
 - cannot be null.

@NotNull(message = "Account type cannot be null")

private String accountType;

- CreationDate:
 - Cannot be null.
 - specify the date format.
 - specify the column name in the database if it differs from the property name in the code.

@NotNull(message = "Creation date cannot be null")

@Temporal(TemporalType.DATE)

@Column(name = "creation_date")

private Date creationDate;

