`

**Gilded:**

1)You can buy and sell 4 different types of ethically sourced gold including: LMBA standard, Traceable, ESG compliant, and Socially Responsible choices\*. These are all available in a variety of bar sizes and can be customized.

2)You can store your gold in numerous vaults around the world, and delivery of gold is available.

3) With the Gilded platform, you can instantly move gold just like invested cash, making gold usable for the first time.

4) Gilded provides you with reliable ownership of physical gold bars using our independently audited and verified private permission blockchain technology.

5) Gilded provides you with a certificate of authenticity and the serial numbers of your gold bars, including the location of the vault where your physical bars are stored.

6) Your gold bars are insured by Lloyd’s of London with 24/7 reporting. Your gold is protected, both digitally and physically.

### 7) Borrow against their gold holdings

### 8)It also provides tail risk without premimum

### 9)it helps in determining any potential losses which may happen in future

Signup Module:

For validation:

@Pattern(regexp = "^[A-Za-z0-9+.!#$%&'\*+-/?^\_`{ |-]+@[A-Za-z0-9.-]+$", message = "Invalid email id format.")  
@NotEmpty(message = "Email id must not be null or empty.")

SignupDto will have:

firstName, lastName, emailId,

, password,country, dateOfBirth, fullName

validation for mobileno:

@MobileNumberConstraint

Check Whether mobile no or email id is taken by other user or not

If then not available.

Foreget Password on basis of mobile no.

forgotPasswordDto: mobileNo, String newPassword, String confirmPassword

**ChangePassword on basis of email:**

oldPassword, String newPassword, String confirmPassword

**Security with Jwt :**

**When we run our project it will go into configuration file where we called filter Filter will send request to AuthenticationFilter in which authentication method will authentic ate using UserdetailsService which use loadByUsername to load user from db or inmemory database ,if authenticate it will send Jwt token.**

**It will also get all the authorities of user from token who is already register and check during each api call coz. We have added following method in configuration file**

http.addFilterBefore(jwtFilter, UsernamePasswordAuthenticationFilter.class)

During the first request the client sends a POST request with username and password. Upon successful authentication the server generates the JWT sends this JWT to the client. This JWT can contain a payload of data. On all subsequent requests the client sends this JWT token in the header. Using this token the server authenticates the user. So we don't need the client to send the user name and password to the server during each request for authentication, but only once after which the server issues a JWT to the client.

**In project we have use DataBase Authentication + but we can use ldap also.**

**Ldap:**

LDAP (Lightweight Directory Access Protocol) is a protocol used to access and manage directory information services.

1. The LdapAuthoritiesPopulator can be used to retrieve authorization information for authenticated users.
2. LDAP is used to store user credentials and authorization data.
3. Spring Security provides the LdapAuthenticationProvider class to authenticate users using an LDAP server.
4. The LdapAuthoritiesPopulator class is used to retrieve user authorization information from an LDAP server.

Send Otp module:

mimeMessage: is used to set from, to ,subject, mesg etc

dependency:  
spring boot starter mail

method to generate opt:

**@Override public String generateOneTimePassword(String email) { int low = 100000; int high = 1000000; String otp = String.valueOf(random.nextInt(high - low) + low); return otp; }**