

BÀI TẬP THỰC HÀNH TUẦN 1

Yêu cầu

Tạo một csdl có tên mydb với các bảng dữ liệu account (account_id, full_name, password, email, phone, status), bảng role (role_id, role_name, description, status). Một account thuộc về nhiều role, mỗi role có thể có nhiều account. Trong đó status là trường để chỉ trạng thái mẫu tin (1-active, 0-deactive, -1-xóa)

(Tip: tạo 1 bảng thứ ba có tên grant_access (account_id, role_id, is_grant, note) với trường is_grant nhận giá trị 0-disable, 1-enable.)

Một bảng có tên log để ghi vết lần đăng nhập. Thông tin bao gồm: account đăng nhập, ngày giờ đăng nhập, ngày giờ đăng xuất, ghi chú.

Tạo một jakartaEE project có tên **week01_lab_HotenSv_mssv** (Có thể kết nối GitHub/GitLab để push code). Thực hiện các công việc sau:

- Tạo một servlet có tên **ControlServlet** (partern cùng tên). Servlet này nhận một tham số (parameter) có tên là **action**. Tham số này nhận các giá trị chuỗi để chỉ các hành động tương ứng.
- Kết nối với csdl, thực hiện các chức năng:
 - o Thêm, cập nhật, xóa ở các bảng đã cho.
 - o Đăng nhập
 - o Hiện thị thông tin tài khoản (nếu đăng nhập thành công)
 - o Hiện thị các quyền của một account.
 - o Hiện thị các account của một role
 - o Cấp quyền cho một account
 - o Ghi log mỗi lần account đăng nhập, đăng xuất.
- Một trang html hiển thị cửa sổ đăng nhập. Nếu đăng nhập thành công và là quyền admin thì hiển thị trang dashboard cho phép quản lý các account khác (bao gồm các quyền thêm, xóa, sửa và cấp quyền). Còn không (không phải admin) thì hiển thị thông tin của người đăng nhập cùng các quyền mà người đó có.

Upload project lên Github/GitLab/BitBucket và chia sẻ code với guider để chấm điểm.

Hướng dẫn thực hành:

Cấu trúc của project có dạng Figure 1.

Trong Controller Servlet, ở các phương thức (GET, POST,...) ta lấy tham số action bằng cách

```
String action = request.getParameter("action");
```

Sau đó tùy thuộc action mà xử lý công việc khác nhau.

Client driver cho MariaDB:

```
Gradle:
implementation 'org.mariadb.jdbc:mariadb-java-client:3.2.0'

Maven:
<dependency>
```

```
<groupId>org.mariadb.jdbc</groupId>
<artifactId>mariadb-java-client</artifactId>
<version>3.2.0</version>
</dependency>
```

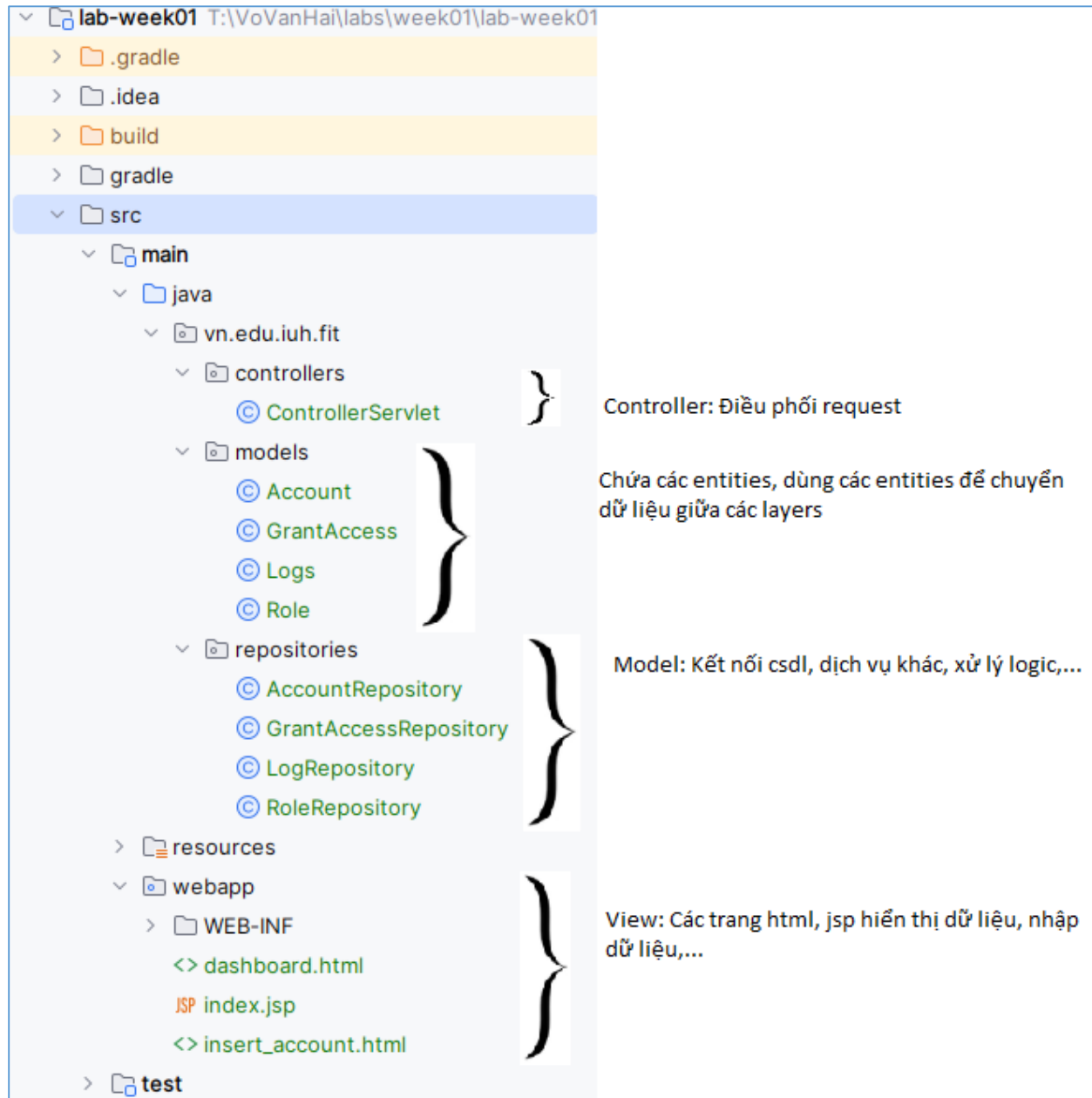
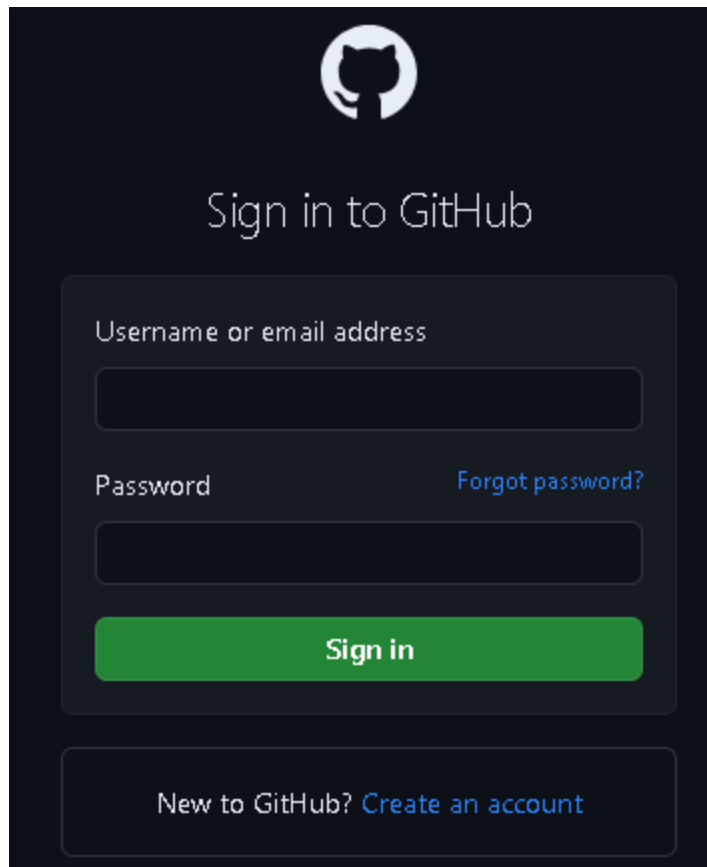


Figure 1. Project structure

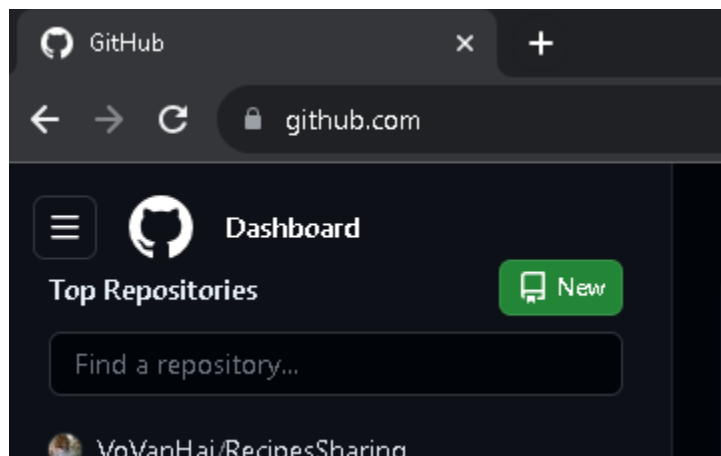
Hướng dẫn Git-Hub:

Vào trang : <https://github.com/login>



Đăng nhập vào Git-Hub. Nếu chưa có tài khoản thì tạo tài khoản mới.

Sau khi đăng nhập thành công, cửa sổ sẽ ở dạng này



Nhấn New để tạo mới 1 project

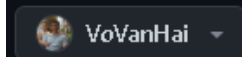
Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *

www_lab_week1

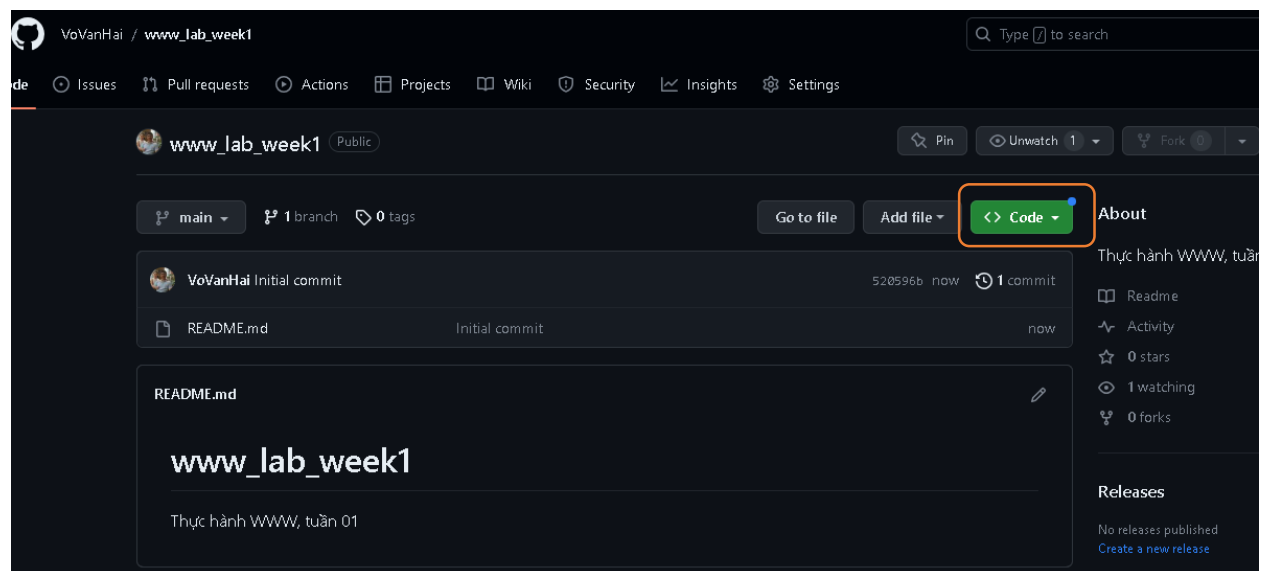
✓ www_lab_week1 is available.

Great repository names are short and memorable. Need inspiration? How about [laughing-octo-potato](#) ?

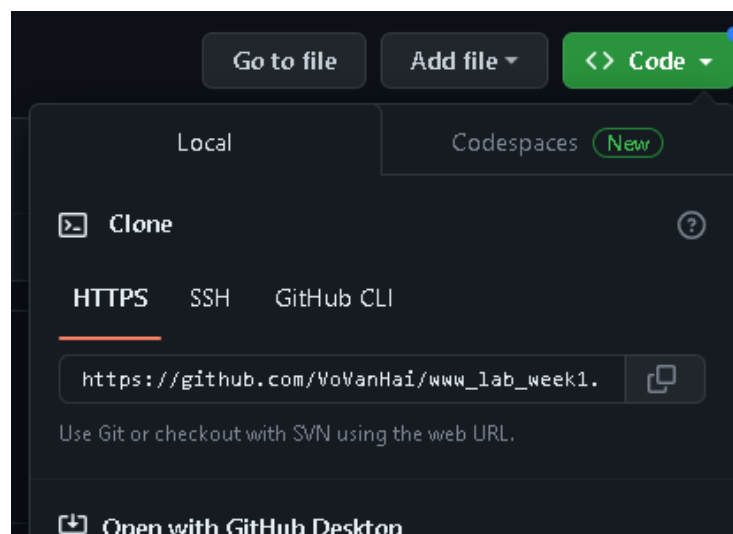
Description (optional)

Thực hành WWW, tuần 01

Nhấn nút Create bên dưới để tạo project



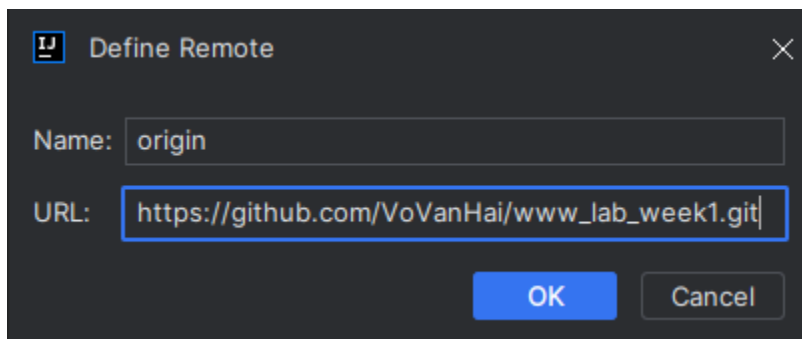
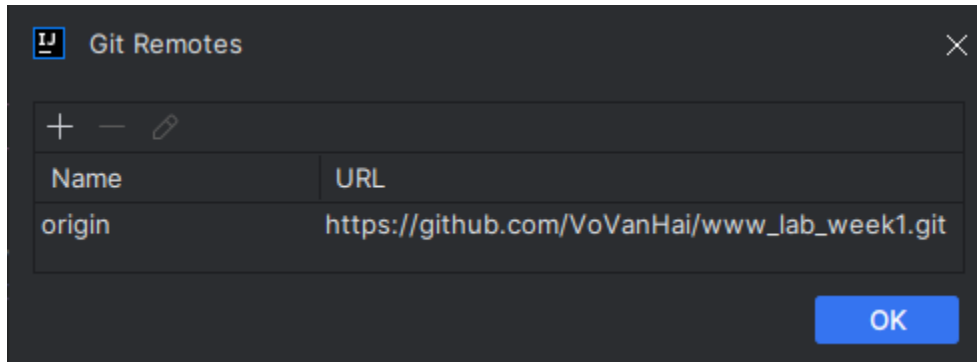
Nhấn nút Code, copy URL.



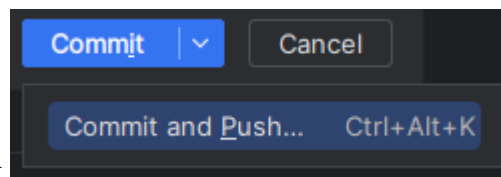
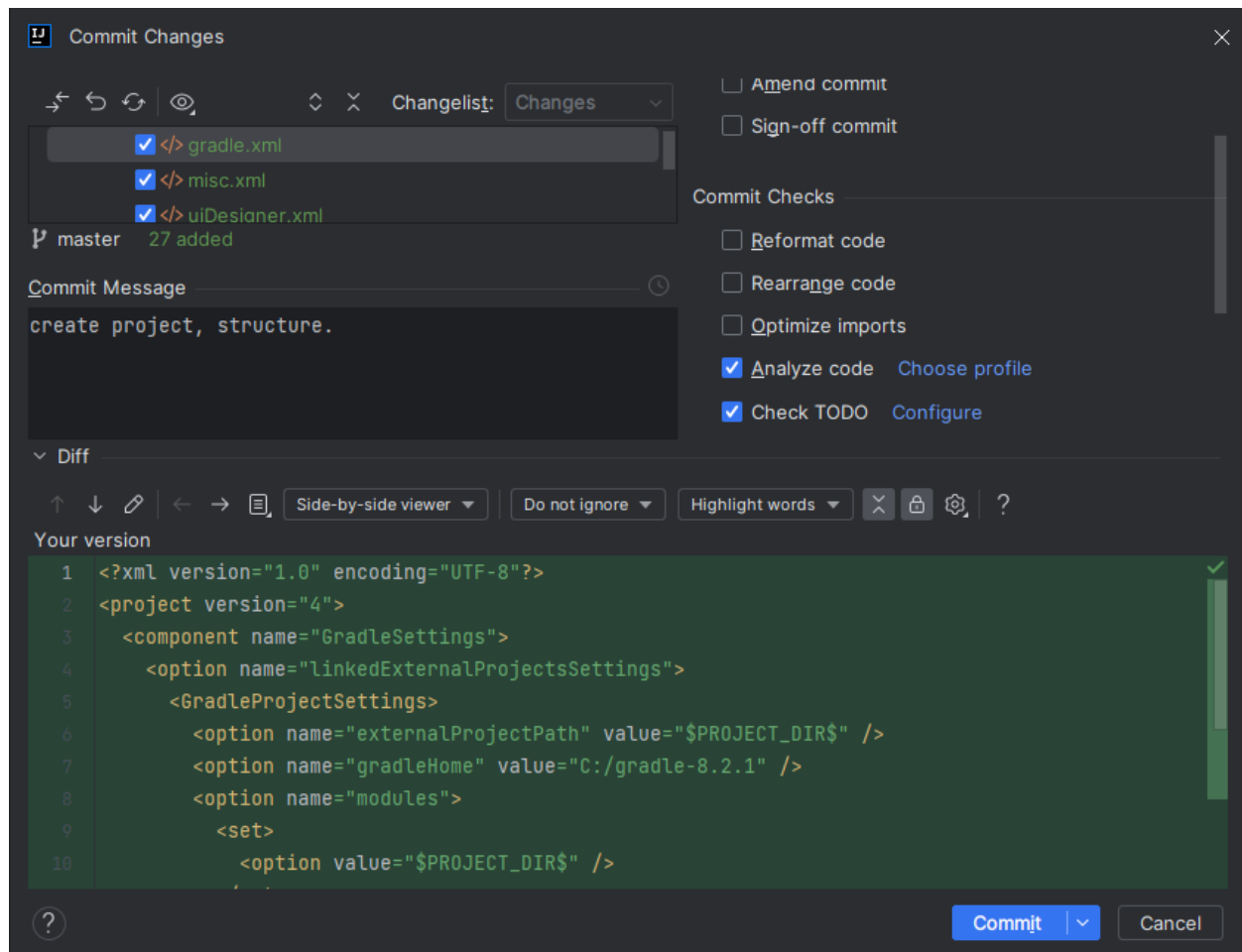
Trên IntelliJ IDEA, vào menu Git (sẽ xuất hiện nếu ban đầu bạn check vào

☒ Create Git repository

khi tạo mới project), chọn item “Manage Remotes...”. Một cửa sổ xuất hiện. Nhấn nút + để thêm vào một remote git repository, như hình

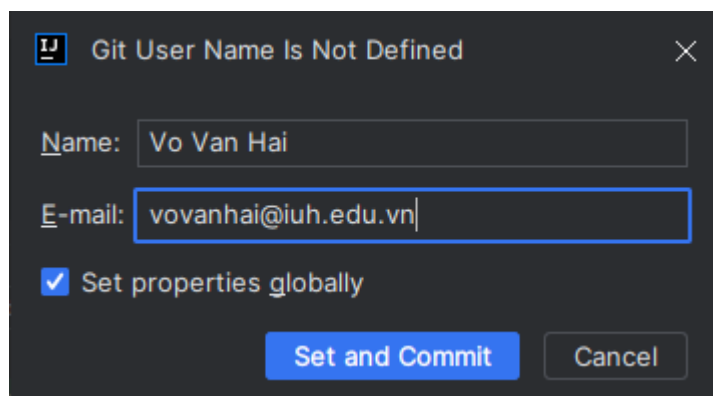


Vào lại menu Git, chọn Commit, cửa sổ Commit Changes xuất hiện. Nhập vào Commit Message

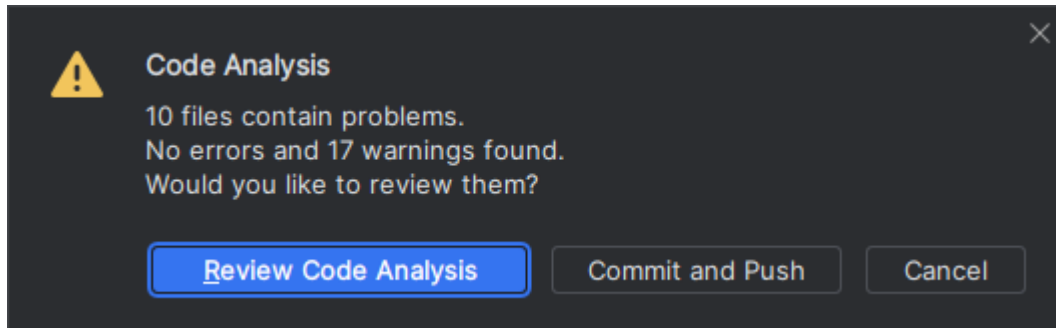


rồi nhấn xổ xuống nhỏ bên cạnh nút Commit. Chọn Commit and Push. (hoặc nhấn Commit rồi sau đó vào menu Git nhấn Push).

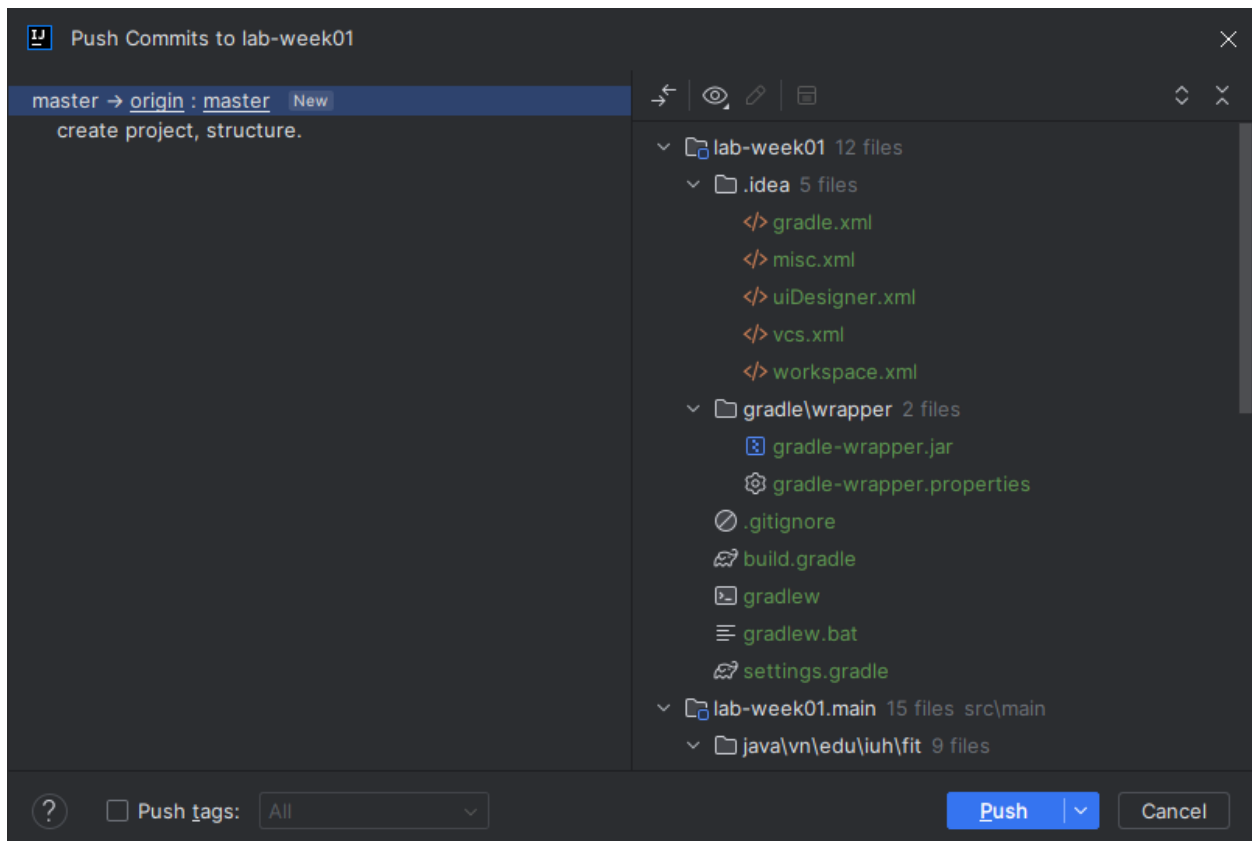
Nếu bạn dev trên máy của trường, sẽ có cửa sổ xuất hiện cho việc nhập tên, email của người commit. Nhập thông tin và nhấn Set and Commit



Git sẽ đánh giá code và đưa cảnh báo nếu code của bạn còn warning hay error. Nếu cần check lại thì nhấn nút “Review Code Analysis”, nếu cứ chọn commit thì nhấn nút “Commit and Push”. Trong trường hợp thử nghiệm này, “Commit and Push” sẽ được chọn. Khuyến cáo nên chọn Review Code Analysis.

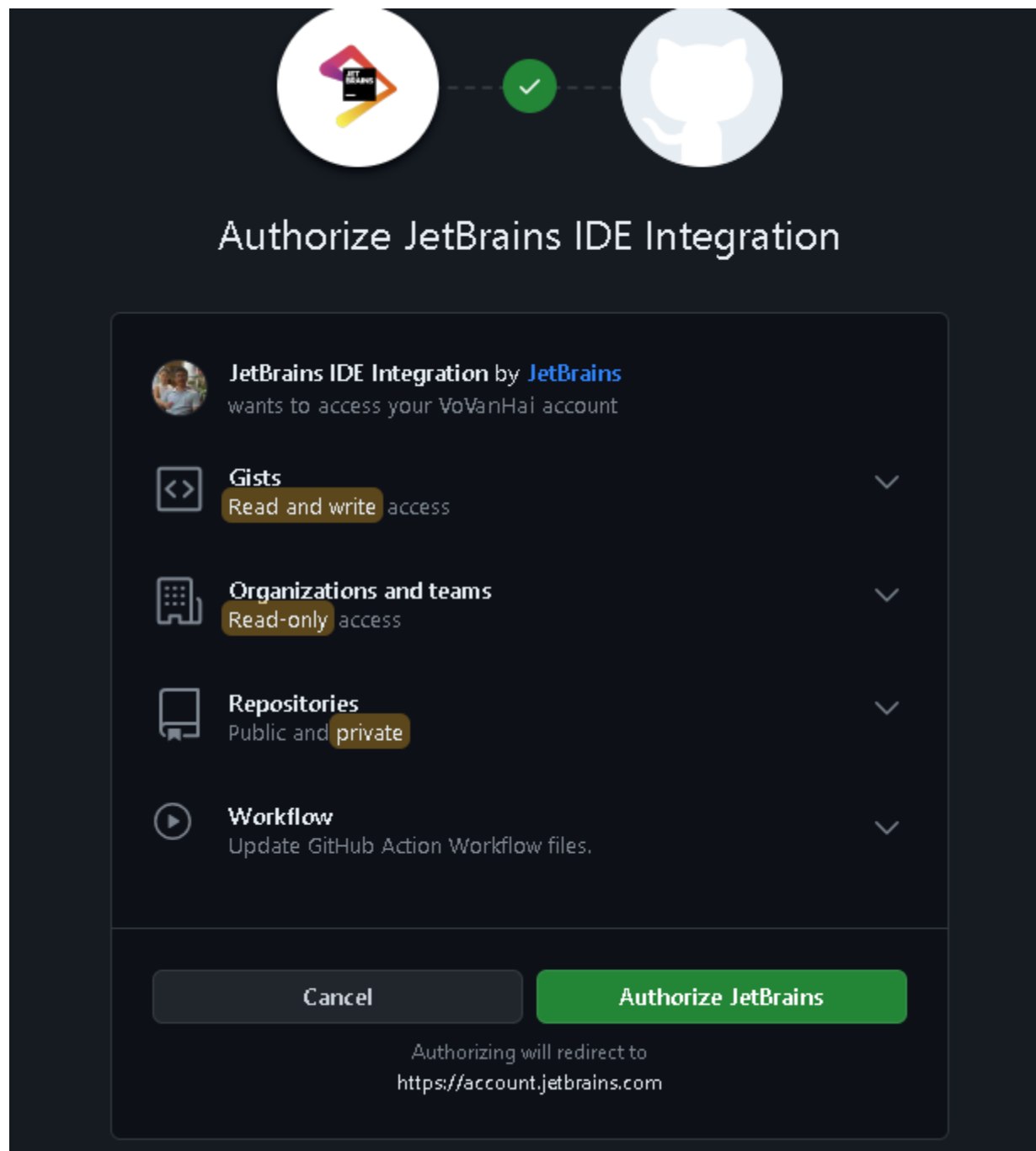


Sau khi Commit, cửa sổ Push sẽ xuất hiện




Mọi thứ OK thì bạn nhấn Push để đẩy lên GitHub.


Bạn có lẽ sẽ được nhận 1 khuyến cáo Authorize để IntelliJ có thể tích hợp với Git. Bạn nên đồng ý như sau




Sau khi commit, bạn sẽ thấy thay đổi trên GitHub



master



2 branches




0 tags

Go to file










Add file

<> Code



VoVanHai Create README.md

401bf2c 3 minutes ago 2 commits

	.idea	create project, structure.	22 minutes ago
	gradle/wrapper	create project, structure.	22 minutes ago
	src/main	create project, structure.	22 minutes ago
	.gitignore	create project, structure.	22 minutes ago
	README.md	Create README.md	3 minutes ago
	build.gradle	create project, structure.	22 minutes ago
	gradlew	create project, structure.	22 minutes ago
	gradlew.bat	create project, structure.	22 minutes ago
	settings.gradle	create project, structure.	22 minutes ago

Database script

```
-- Dumping database structure for mydb
CREATE DATABASE IF NOT EXISTS `mydb` /*!40100 DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci */;
USE `mydb`;

-- Dumping structure for table mydb.account
CREATE TABLE IF NOT EXISTS `account` (
  `account_id` varchar(50) NOT NULL,
  `full_name` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `email` varchar(50) DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `status` tinyint(4) NOT NULL DEFAULT 1,
  PRIMARY KEY (`account_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- Dumping data for table mydb.account: ~2 rows (approximately)
INSERT INTO `account` (`account_id`, `full_name`, `password`, `email`,
`phone`, `status`) VALUES
  ('met', 'Tran Thi Met', '123', 'met@gmail.com', '0904567890', 1),
  ('teo', 'NGuyen Van Teo', '123', 'teo@gmail.com', '0903123456', 1);

-- Dumping structure for table mydb.grant_access
CREATE TABLE IF NOT EXISTS `grant_access` (
  `role_id` varchar(50) NOT NULL,
  `account_id` varchar(50) NOT NULL,
  `is_grant` bit(1) NOT NULL DEFAULT b'1',
  `note` varchar(250) DEFAULT '',
  PRIMARY KEY (`role_id`, `account_id`),
  KEY `account_grant` (`account_id`),
  CONSTRAINT `account_grant` FOREIGN KEY (`account_id`) REFERENCES `account`
(`account_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `role_grant` FOREIGN KEY (`role_id`) REFERENCES `role`
(`role_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

```

-- Dumping data for table mydb.grant_access: ~2 rows (approximately)
INSERT INTO `grant_access` (`role_id`, `account_id`, `is_grant`, `note`)
VALUES
  ('admin', 'teo', b'1', ''),
  ('user', 'met', b'1', '');

-- Dumping structure for table mydb.log
CREATE TABLE IF NOT EXISTS `log` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `account_id` varchar(50) NOT NULL,
  `login_time` datetime NOT NULL DEFAULT current_timestamp(),
  `logout_time` datetime NOT NULL DEFAULT current_timestamp(),
  `notes` varchar(250) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci COMMENT='ghi logs';

-- Dumping data for table mydb.log: ~0 rows (approximately)

-- Dumping structure for table mydb.role
CREATE TABLE IF NOT EXISTS `role` (
  `role_id` varchar(50) NOT NULL,
  `role_name` varchar(50) NOT NULL,
  `description` varchar(50) DEFAULT NULL,
  `status` tinyint(4) NOT NULL,
  PRIMARY KEY (`role_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- Dumping data for table mydb.role: ~2 rows (approximately)
INSERT INTO `role` (`role_id`, `role_name`, `description`, `status`) VALUES
  ('admin', 'administrator', 'admin role', 1),
  ('user', 'user', 'user role', 1);

```

Muốn xem relationship diagram: vào trang <https://dev.mysql.com/downloads/workbench/>, download MySQL Workbench (MW) sau đó cài đặt sau đó mở MySQL Workbench lên.

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters **SSL** **Advanced**

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Tạo 1 kết nối. Nhấn Test Connection để kiểm tra kết nối thành công không.

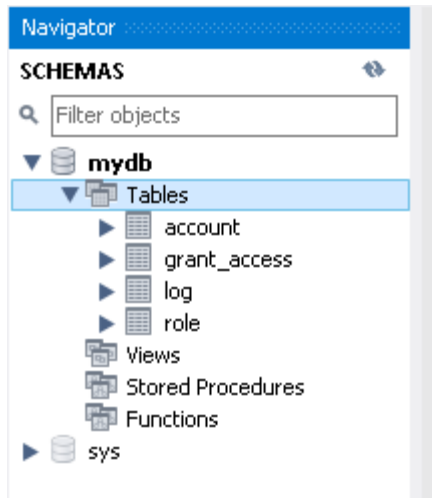
MySQL Workbench

i **Successfully made the MySQL connection**

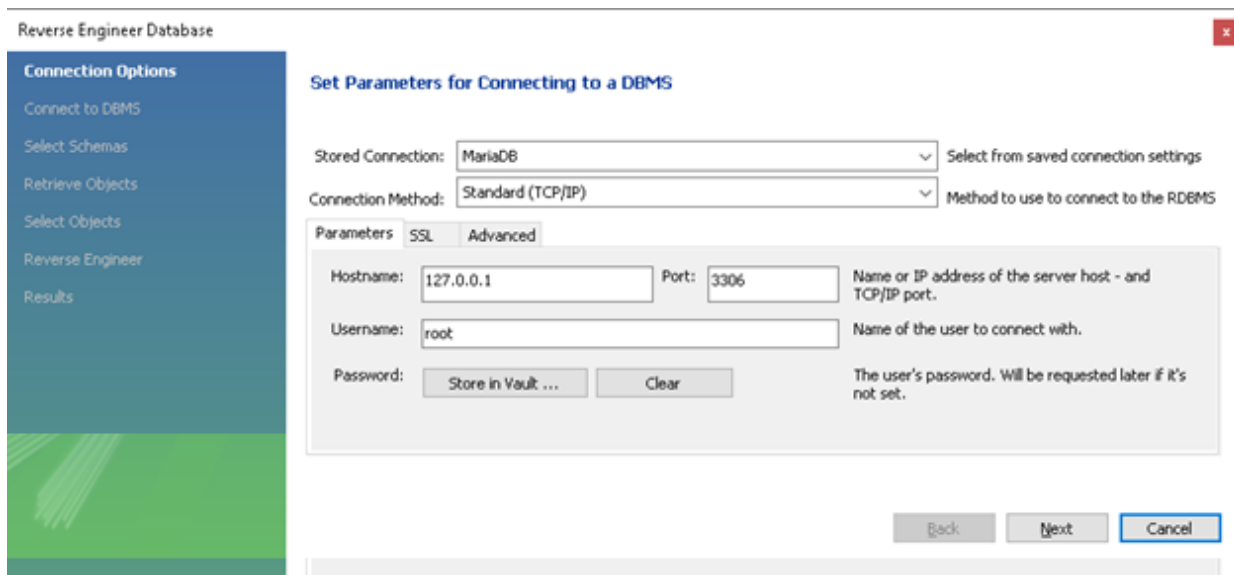
Information related to this connection:

Host: 127.0.0.1
Port: 3306
User: root
SSL: not enabled

A successful MySQL connection was made with the parameters defined for this connection.



Trong MW, Vào menu Database, chọn “**Reverse Engineer**”.



Nhấn Next cho đến khi hoàn tất.

ER diagram sẽ xuất hiện và bạn có thể hiệu chỉnh nó.

