**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN**



**BẢO MẬT WEB**

**MÔ TẢ HỆ THỐNG**

**ỨNG DỤNG WEB HỖ TRỢ THI TRẮC NGHIỆM ONLINE**

**Nhóm:**

**SVTH: Trần Khương Duy 16110297**

**Huỳnh Công Chiến 16110282**

**Hồ Nguyễn Hoàng Quân 16110429**

**Hồ Chí Minh , Ngày 2 tháng 5 năm 2019**

**MỤC LỤC**

[**Chương 1. Giới thiệu chung 1**](#_1fob9te)

[**1. Nêu lý do chọn đề tài 1**](#_3znysh7)

[**2. Mục đích nghiên cứu 1**](#_tyjcwt)

[**3. Phương pháp nghiên cứu 1**](#_3dy6vkm)

**4. Hướng dẫn cài đặt ứng dụng trên Localhost**

[**​ Chương 2. Phân tích và thiết kế 3**](#_1t3h5sf)

[**​1. Mẫu mô tả phân công công việc 3**](#_4d34og8)

[**2. Các mẫu dùng cho thiết kế CSDL 4**](#_17dp8vu)

[**2.1. Hình vẽ biễu diễn quan hệ giữa các bảng : 4**](#_3rdcrjn)

[**2.2 Mẫu bảng mô tả các bảng trong CSDL 4**](#_26in1rg)

[**3. Các biểu mẫu dùng cho thiết kế giao diện 7**](#_lnxbz9)

[**4. Các biểu mẫu dùng cho đặc tả phần mềm 14**](#_35nkun2)

[**4.1 Use case 14**](#_1ksv4uv)

[**4.2 Bảng mô tả các actor 14**](#_44sinio)

[**4.3 Bảng mô tả các use case 15**](#_2jxsxqh)

[**4.4 Bảng mô tả các chức năng 17**](#_z337ya)

Chương 1. Giới thiệu chung

**1. Nêu lý do chọn đề tài**

Hiện nay nhu cầu học cũng như thi online phát triển càng ngày càng nhiều.

Nhóm em nhận thầy cần có 1 ứng dụng để hỗ trợ việc học cũng như thi trực tuyến. Từ đó nhóm em quyết định chọn đề tài của nhóm là xây dựng ứng dụng Web hỗ trợ thi trắc nghiệm.

## 2. Mục đích nghiên cứu

Nhẳm đáp ứng nhu cầu của người dùng , đồng thời thiết kế giao diện cho phù hợp là làm cho người sử dụng có thể tiếp nhận và sử dụng ứng dụng 1 cách dể dàng và đơn giản nhất.

## 3. Phương pháp nghiên cứu

Chúng tôi đã cố gắng tìm tòi tài liệu cả trong giáo trình và những tài liệu tham khảo bên ngoài với tinh thần trách nhiệm và sưu tầm, tổng hợp lại các nội dung một cách ngắn gọn, dễ hiểu nhất. Tuy nhiên nhiều mặt kiến thức còn hạn chế nên không tránh khỏi thiếu sót. Rất mong sự đóng góp ý kiến của Giảng viên cũng như của bạn đọc để chúng tôi có thể hoàn thiện kiến thức của mình.

Với đề tài đã lựa chọn ở trên chúng tôi đã khảo sát các ứng dụng quản lý Web hỗ trợ thi trắc nghiệm nên ra được ưu điểm và nhược điểm của từng ứng dụng để đem lại hiệu quả cao trong quá trình nghiên cứu và xây dựng ứng dụng cho nhóm.

**4. Hướng dẫn cài đặt ứng dụng trên localhost**

**4.1 Yêu cầu phần mềm**

-DBMS Sql server 2017

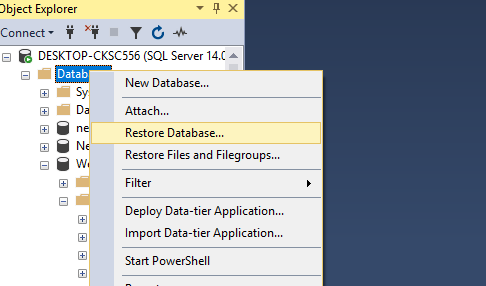
-Máy chủ Apache tomcat

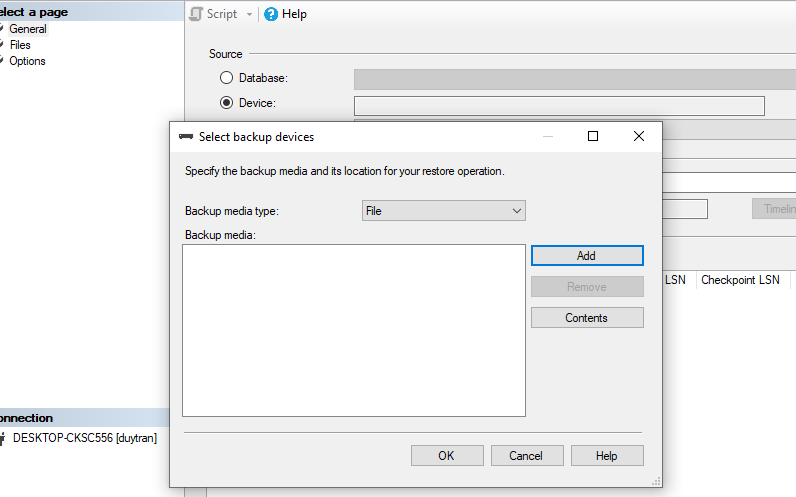
-JDK 11

-IDE Eclipse hoặc Netbean

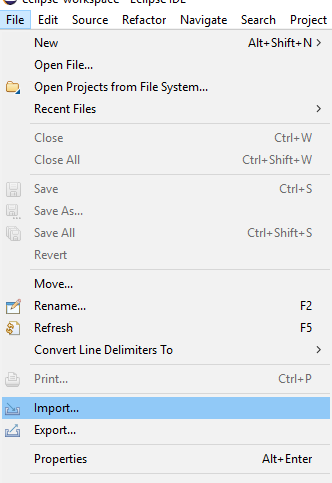
**4..2 Hướng dẫn cài đặt**

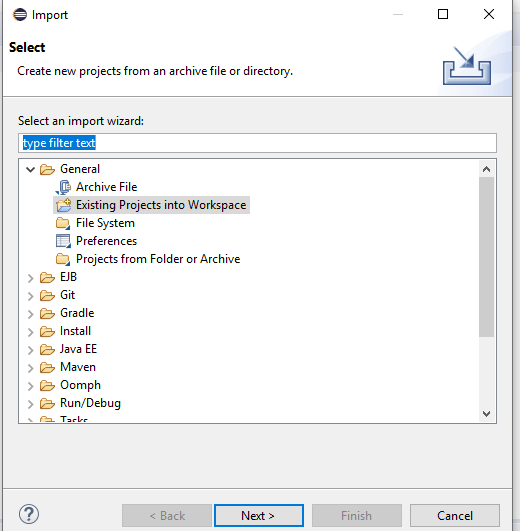
* **Tiến hành restore database bằng file .bak được gửi kèm với tài liệu này**
* Right click Database → Restore database

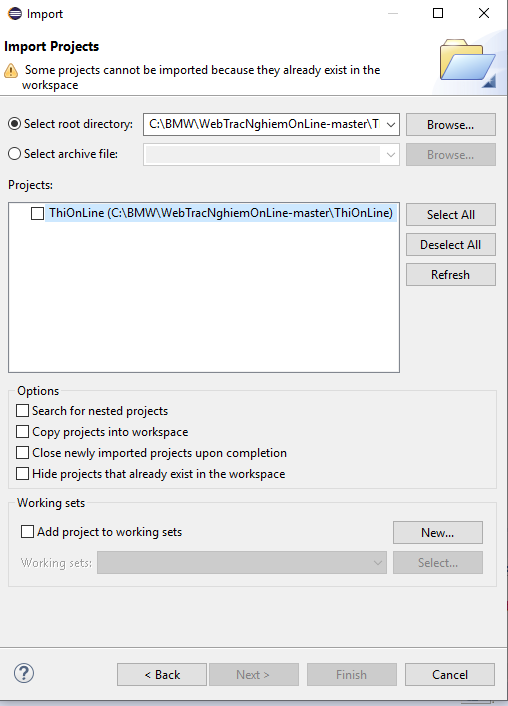
****

* Tick Device → Add → Chọn đến file .bak đã tải về****
* **Mở project trong IDE của bạn**

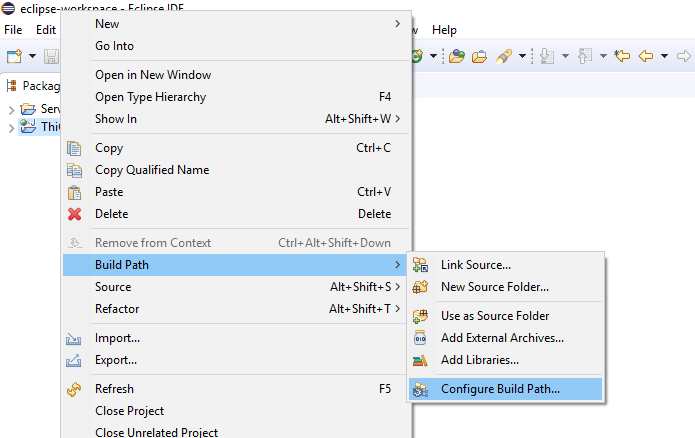
Đối với IDE eclipse, chọn File → Import → General → Existing project into workspace → Next → Chọn đến thư mục project đã tải về → Finish

****

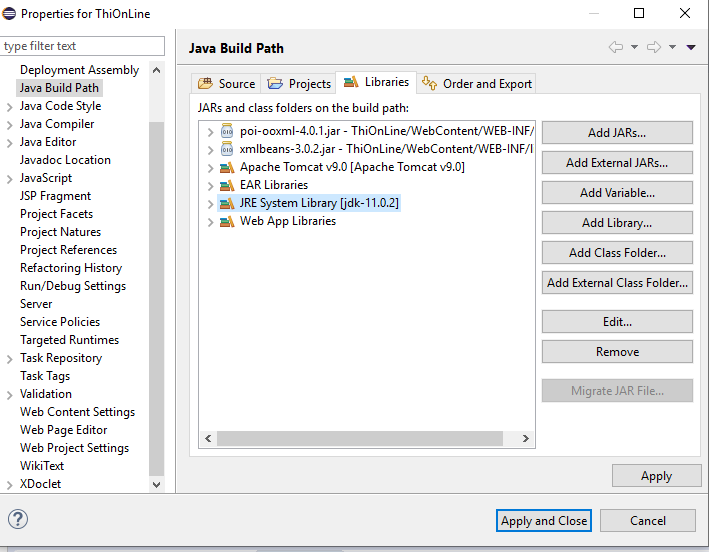
****

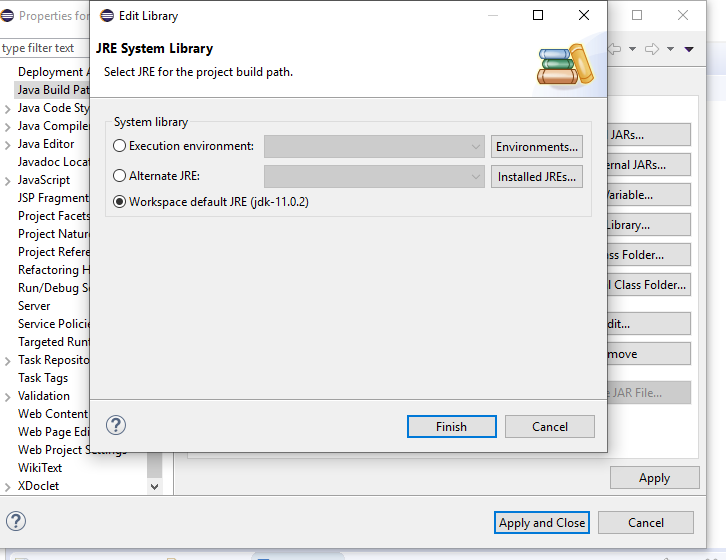
****

* **Configure build path**
* Click chuột phải vào project → Build path → configure build path

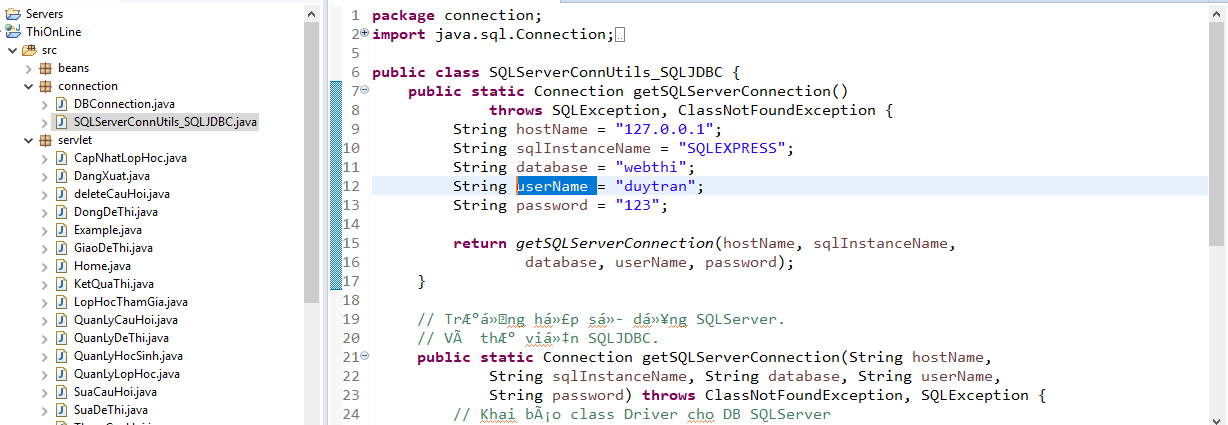
****

* Trong tab libraries , double click vào JRE system libraries → chọn JDK 11 (nếu chưa cài đặt, hãy cài đặt JDK 11) → finish → apply

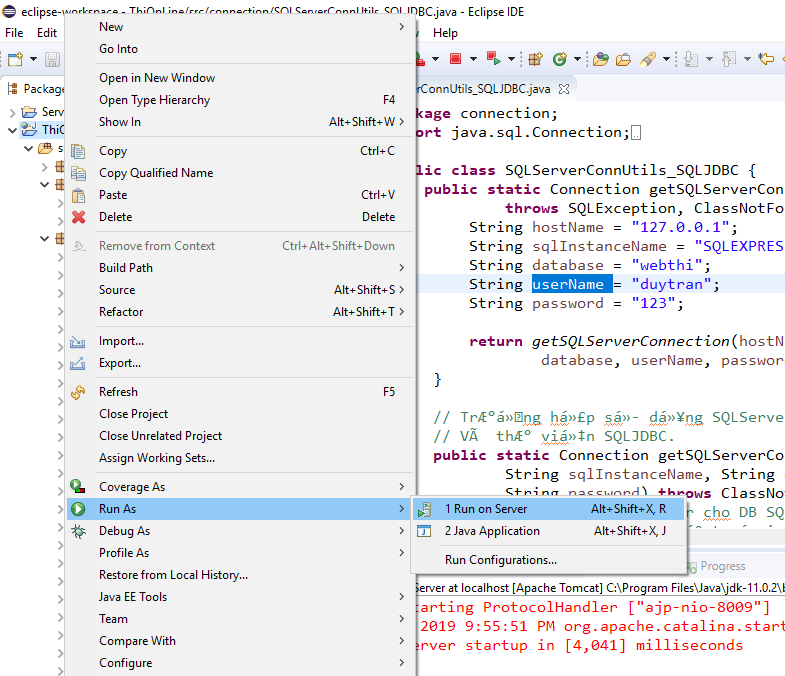
****

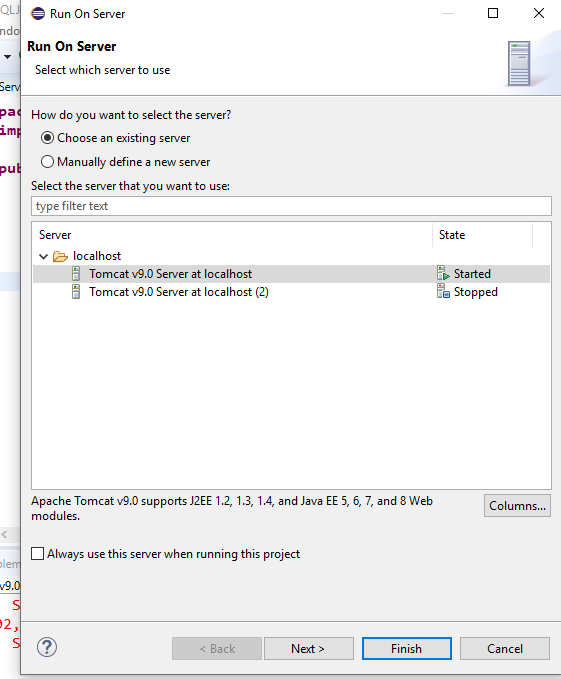
****

* **Config kết nối với database**
* Mở file SQLServerConnUtils\_SQLJDBC.java , thực hiện thay đổi sqlInstanceName (là phiên bản SQL server đang sử dụng), userName và passWord (dùng để đăng nhập vào máy chủ sql).



* Để thực hiện kết nối, cần cấu hình cho phép kết nối SQL server từ xa, xem hướng dẫn tại <https://www.youtube.com/watch?v=zoFcqt9Yufw>
* **Chạy ứng dụng với máy chủ apache tomcat**
* Click chuột phải vào project → run as → run on server → chọn đến máy chủ tomcat bạn đã cài đặt→ finish





**Để sử dụng các chức năng của website, yêu cầu người dùng phải đăng nhập hệ thống:**

* Người quản lý thí sinh và lớp học: Có thể thêm, xóa, sửa thí sinh, tạo, xóa, sửa lớp học gồm nhiều thí sinh.

Tài khoản demo: duytran, password: 123

* Người ra đề thi: có thể nhập số câu hỏi cho mỗi loại câu hỏi, phần mềm tự chọn ngâu nhiên số câu hỏi từ mỗi loại trong ngân hàng câu hỏi và tạo ra đề thi trắc nghiệm; có thể xem lại đề thi và chỉnh sửa; có thể set thời gian bắt đầu mở đề thi, và thời lượng làm bài; gán đề thi cho lớp học.

Tài khoản demo: liem , password: 123

* Người quản lý ngân hàng câu hỏi: Có thể thêm câu hỏi, xóa câu hỏi, liệt kê các câu hỏi, import danh sách câu hỏi từ file.

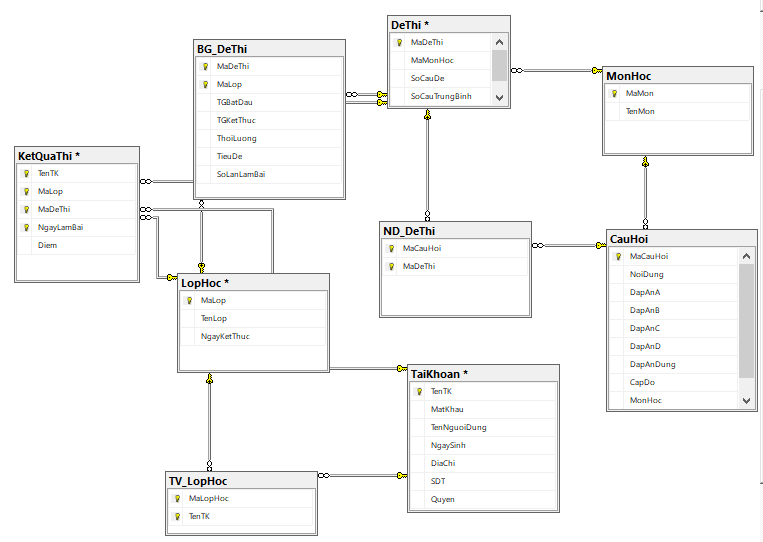
Tài khoản demo: Bao, password: 123

* Thí sinh: có thể login vào phần mềm và làm bài thi trắc nghiệm. Sau khi thi xong thì sẽ có file pdf hiển thị kết quả. Trên trang bài làm có đồng hồ đếm ngược. Khi hết giờ làm bài màn hình tự khóa và thí sinh không thể làm bài được.

Tài khoản demo: Anh, password: 123

**2. Các mẫu dùng cho thiết kế CSDL**

**2.1. Hình vẽ biễu diễn quan hệ giữa các bảng :**



**2.2 Mẫu bảng mô tả các bảng trong CSDL**

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng | Mục đích của bảng trong chương trình. |
| 1 | MonHoc | Lưu trữ các môn học. |
| 2 | DeThi | Lưu trữ thông tin đê thi (số lượng câu hỏi, môn học) |
| 3 | BG\_DeThi | Lưu trữ thông tin gán đề thi cho các lớp |
| 4 | LopHoc | Lưu trữ các lớp học. |
| 5 | TV\_LopHoc | Lưu trữ các học viên trong các lơp học. |
| 6 | TaiKhoan | Lưu trữ thông tin các nhân của từng học viên trong lớp học. |
| 7 | ND-DeThi | Tạo ra nội dung của đề thi. |
| 8 | CauHoi | Lưu trữ các câu hỏi và đáp án của câu hỏi trong các đề thi và môn học |
| 9 | KetQuaThi | Lưu trữ thông tin về ngày làm bài, điểm số của bài thi. |

**2.2.1.Bảng MonHoc**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaMon | Lưu trữ các mã môn học của các môn học. |
| 2 | TenMon | Lưu trữ các tên môn học. |

**2.2.2.Bảng DeThi**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaDeThi | Lưu trữ các mã đề thi của môn học. |
| 2 | MaMonHoc | Lưu trữ các mã môn học. |
| 3 | SoCauDe | Lưu trữ số câu dễ của đề thi |
| 4 | SoCauTrungBinh | Lưu trữ số câu trung bình của đề thi |
| 5 | SoCauKho | Lưu trữ số câu khó của bài thi |

**2.2.3.Bảng BG\_DeThi**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaDeThi | Lưu trữ mã các đề thi. |
| 2 | MaLop | Lưu trữ mã các lớp học được gán đề thi |
| 3 | TGBatDau | Lưu trữ thời gian mở đề đối với lớp |
| 4 | TGKetThuc | Lưu trữ thời gian đóng đề đối với lớp |
| 5 | ThoiLuong | Lưu trữ thời lượng làm bài thi của các bài thi đối với lớp |
| 6 | TieuDe | Lưu trữ tên của các bài thi. |
| 7 | SoLanLamBai | Lưu trữ số lần có thể làm bài của bài thi. |

**2.2.4.Bảng LopHoc**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaLop | Lưu trữ mã các lớp học. |
| 2 | TenLop | Lưu trữ tên của các lớp học. |
| 4 | NgayKetThuc | Lưu trữ ngày mà lớp học đó ngừng hoạt động |

**2.2.5.Bảng TV\_LopHoc**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaLopHoc | Lưu trữ các mã các lớp học. |
| 2 | TenTK | Lưu trữ các tên học viên của các lớp học. |

**2.2.6. Bảng TaiKhoan**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | TenTK | Lưu trữ tên các tài khoản (đăng nhập) của người dùng |
| 2 | MatKhau | Lưu trữ mật khẩu đănh nhập của các người dùng |
| 3 | TenNguoiDung | Lưu trữ tên thật của người dùng |
| 4 | Ngaysinh | Lưu trữ ngày sinh của người dùng |
| 5 | DiaChi | Lưu trữ địa chỉ của người dùng |
| 6 | SDT | Lưu trữ số điện thoại của người dùng |
| 7 | Quyen | Lưu trữ quyền truy cập của người dùng. |

**2.2.7. Bảng KetQuaThi**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | TenTK | Lưu trữ Tên học viên đã làm bài. |
| 2 | MaLop | Lưu trữ mã lớp của học viên đã làm bài thi. |
| 3 | MaDeThi | Lưu trữ mã đề thi của bài thi đã làm. |
| 4 | NgayLamBai | Lưu trữ thời gian bắt đầu làm bài thi |
| 6 | Diem | Lưu trữ điểm thi mà học viên đạt được. |

**2.2.8.Bảng CauHoi**

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaCauHoi | Lưu trữ mã các câu hỏi của đề thi. |
| 2 | NoiDung | Lưu trữ nội dung của câu hỏi. |
| 3 | DapAnA | Lưu trữ đáp án A của câu hỏi. |
| 4 | DapAnB | Lưu trữ đáp án B của câu hỏi. |
| 5 | DapAnC | Lưu trữ đáp án C của câu hỏi. |
| 6 | DapAnD | Lưu trữ đáp án D của câu hỏi. |
| 7 | DapAnDung | Lưu trữ đáp án đúng của câu hỏi. |
| 8 | CapDo | Lưu trữ cấp độ của câu hỏi. |
| 9 | MonHoc | Lưu trữ môn học của câu hỏi. |

**2.2.9. Bảng ND\_DeThi**

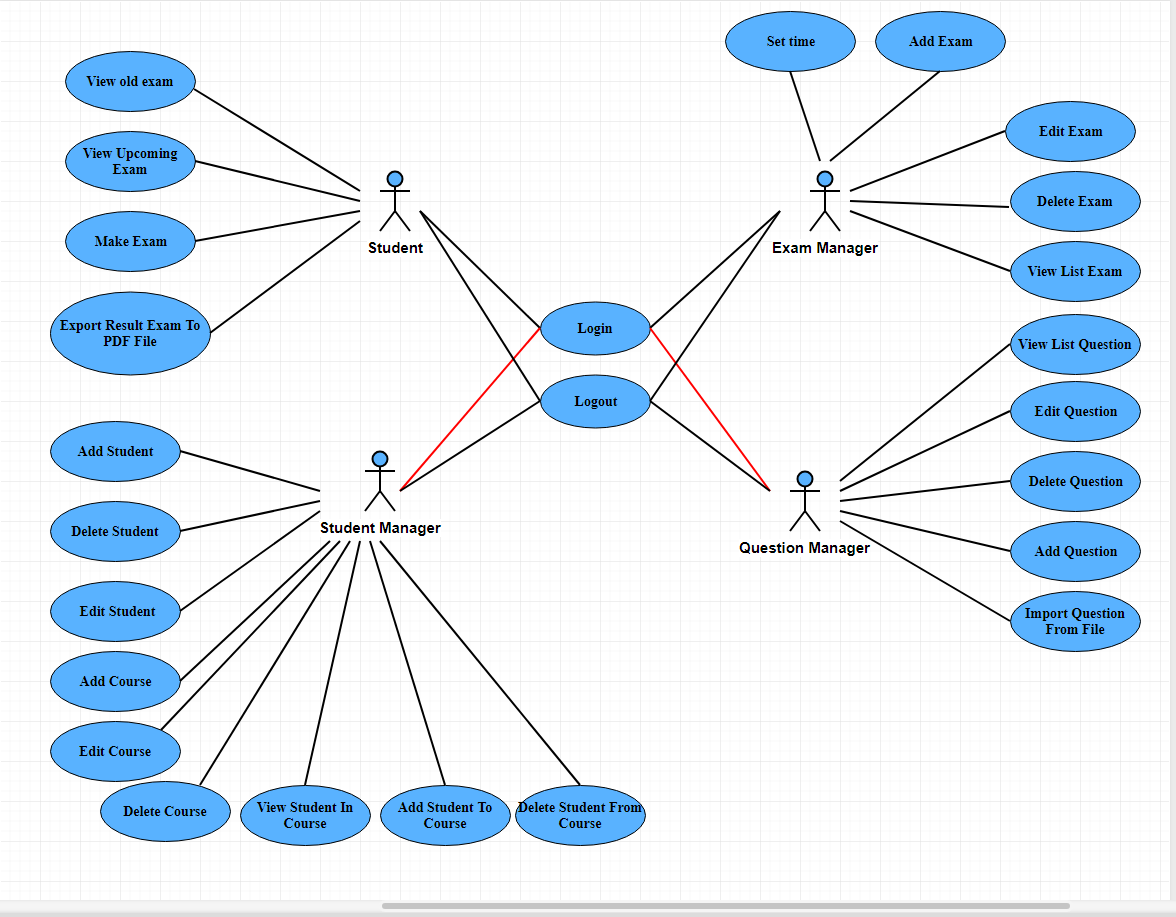
|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường |
| 1 | MaCauHoi | Lưu trữ các mã của câu hỏi trong đề thi |
| 2 | MaDeThi | Lưu trữ mã của đề thi. |

## 3. Các biểu mẫu dùng cho thiết kế giao diện

|  |  |  |
| --- | --- | --- |
| *STT* | *Màn hình/Cửa sổ/Dialog* | *Mục đích chính của màn hình* |
| 1. | Giao diện đăng nhập | Đăng nhập |
| 2 | Giao diện đăng kí tài khoản | Đăng kí tài khoản |
| 3 | Giao diện trang cá nhân | Giao diện người dùng (học sinh) sau khi đăng nhập, hiển thị thông tin cá nhân, các lớp học đã tham gia và các bài thi hiện tại. |
| 4 | Giao diện làm bài thi | Hiển thị nội dung bài thi |
| 5 | Giao diện kết quả thi | Hiển thị kết quả của bài thi vừa hoàn tất  Tùy chọn tải file pdf cho người dùng. |
| 6 | Giao diện trang chủ người quản lí học sinh | Hiển thị danh sách học sinh và các tùy chọn quản lí (Thêm, xóa, sửa) |
| 7 | Giao diện sửa thông tin học sinh | Sửa thông tin học sinh |
| 8 | Giao diên thêm một học sinh mới | Thêm một học sinh mới |
| 9 | Giao diện quản lí lớp học | Xem lại nội dung các câu hỏi của bài thi,  cung cấp chức năng xóa câu hỏi và them câu hỏi mới vào đề thi. |
| 10 | Giao diện sửa thông tin lớp học | Sưa thông tin lớp học |
| 11 | Giao diện thêm một lớp mới | Thêm một lớp học mới |
| 12 | Giao diện thêm học sinh vào lớp | Thêm học sinh vào lớp học |
| 13 | Giao diện xóa học sinh khỏi lớp | Xóa học sinh khỏi lớp |
| 14 | Giao diện quản lí đề thi | Hiển thị danh sách đề thi và các tùy chọn quản lý (them, xóa,sửa,gán) |
| 15 | Giao diện tạo đề thi | Hiển thị form tạo đề thi (môn học, số câu hỏi dễ, khó, trung bình) |
| 16 | Giao diện xem trước đề thi | Xem trước nội dung đề thi và thay đổi nó (xóa câu hỏi, them câu hỏi mới) trước khi lưu. |
| 17 | Giao diện sửa đề thi | Xem nội dung đề thi và cung cấp tùy chọn chỉnh sửa (xóa câu hỏi, đổi số lượng câu hỏi, loại câu hỏi) |
| 18 | Giao diên gán đè thi | Gán đề thi cho lớp mới, xóa và sửa (tiêu đề, thời lượng, số lần làm bài, ngày mở, ngày đóng) cho các lớp đã được gán. |
| 19 | Giao diện sửa câu hỏi | Hiển thị danh sách câu hỏi và cung cấp các tùy chọn quản lí câu hỏi (them, xóa, sửa). |
| 20 | Giao diện thêm câu hỏi | Hiển thị các trường nhập nội dung câu hỏi |
| 21 | Giao diện thêm câu hỏi từ file | Hiển thị trường import file (excel) câu hỏi |
| 22 | Giao diện xem và sửa câu hỏi | Xem lại nội dung câu hỏi và cung các cấp tùy chọn chỉnh sửa (nội dung,đáp án, môn học, cấp độ) |

**4. Các biểu mẫu dùng cho đặc tả phần mềm**

**4.1 Use case**



**4.2 Bảng mô tả các actor**

|  |  |  |
| --- | --- | --- |
| TT | Tên actor | Ý nghĩa |
| 1 | Student | Sử dụng Web để làm bài thi trắc nghiệm. |
| 2 | Question Manager | Sử dụng Web để quản lí kho câu hỏi, có trong bài thi thông qua các hình thức thêm, xóa, sửa câu hỏi vào kho câu hỏi. |
| 3 | Exam Manager | Sử dụng Web để quản lí hình thức bài thi cho các thí sinh thông các hình thức tạo đề thi,cài đặt và chỉnh sửa thời gian thi, số câu hỏi của bài thi. |
| 4 | Student Manager | Sử dụng Web để quản lí các thí sinh và lớp học tham gia vào bài thi thông qua các hình thức thêm, xóa, sửa các thi sinh và lớp học |

**4.3 Bảng mô tả các use case**

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
| 1 | Export Result Exam To PDF File | Xuất kết quả thi ra file PDF. |
| 2 | View Old Exam | Xem lại các bài thi cũ đã thi. |
| 3 | View Upcoming Exam | Xem các bài thi đang trong thời gian có thể làm bài. |
| 4 | Make Exam | Làm bài thi online. |
| 5 | Set time | Cài đặt thời gian làm bài thi. |
| 6 | View List Exam | Xem danh sách đề thi. |
| 7 | Edit Exam | Sửa đề thi. |
| 8 | Delete Exam | Xóa đề thi |
| 9 | Add Exam | Thêm đề thi |
| 10 | View List Question | Xem danh sách câu hỏi |
| 11 | Edit Question | Sửa câu hỏi |
| 12 | Delete Question | Xóa câu hỏi |
| 13 | Add Question | Thêm câu hỏi |
| 14 | Import Question From File | Thêm danh sách câu hỏi từ file excel |
| 15 | View List Student | Xem danh sách học sinh |
| 16 | Edit Student | Sửa thông tin sinh viên |
| 17 | Delete Student | Xóa thông tin sinh viên |
| 18 | Add Student | Thêm thông tin sinh viên |
| 19 | Delete student from course | Xóa sinh viên khỏi lớp |
| 20 | Add Student To Course | Thêm sinh viên vào lớp học |
| 21 | View Student In Course | Xem danh sách sinh viên vào lớp học |
| 22 | Edit Course | Sửa lớp học |
| 23 | Delete Course | Xóa lớp học |
| 24 | Edit Course | Sửa lớp học |
| 25 | Add Course | Thêm lớp học |
| 26 | Logout | Đăng xuất khỏi hệ thống |
| 27 | Login | Đăng nhập vào hệ thống |

**4.4 Bảng mô tả các chức năng**

​

|  |  |  |  |
| --- | --- | --- | --- |
| Tên chức năng và sinh viên phụ trách thực hiện | Các hàm giúp thực hiện chức năng | Nhiệm vụ của từng hàm | Trích đoạn code chính và tên file |
| Đăng nhập | KiemTraDangNhap  (File DBUltis) | Kiểm tra một cặp tài khoản, mật khẩu có thể dăng nhập | **public** **static** TaiKhoan KiemTraDangNhap(Connection conn, String tenTK, String matKhau) **throws** SQLException {  String sql = "select TenTK, TenNguoiDung, Quyen from TaiKhoan where TenTK = ? and MatKhau = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, tenTK);  pstm.setString(2, matKhau);  ResultSet rs = pstm.executeQuery();  **if** (rs.next()) {  TaiKhoan tk = **new** TaiKhoan();  tk.setTenTK(tenTK);  String tenNguoiDung = rs.getString("TenNguoiDung");  **int** quyen = rs.getInt("Quyen");  tk.setTenNguoiDung(tenNguoiDung);  tk.setQuyen(quyen);  **return** tk;  }  **return** **null**;  } |
|  | SetTaiKhoanDangNhap  (file MyUltis) | Lưu tài khoản đăng nhập vào session | **public** **static** **void** SetTaiKhoanDangNhap(HttpSession session, TaiKhoan loginedUser) {  session.setAttribute("loginedUser", loginedUser);  } |
|  | GetTaiKhoanDangNhap  (file MyUltis) | Lấy tài khoản đã đăng nhập từ session | **public** **static** TaiKhoan getTaiKhoanDangNhap(HttpSession session) {TaiKhoan loginedUser = (TaiKhoan) session.getAttribute("loginedUser");  **return** loginedUser;  } |
|  | StoreUserCookie  (file MyUltis) | Lưu tài khoản đã đăng nhập vào cookie (chức năng remember me) | **public** **static** **void** storeUserCookie(HttpServletResponse response, TaiKhoan user) {  Cookie cookieUserName = **new** Cookie(***ATT\_NAME\_USER\_NAME***, user.getTenTK()); cookieUserName.setMaxAge(24 \* 60 \* 60); response.addCookie(cookieUserName);  } |
|  | GetUserNameInCookie  (file MyUltis) | Lấy tài khoản đã đăng nhập từ cookie | **public** **static** String getUserNameInCookie(HttpServletRequest request) {  Cookie[] cookies = request.getCookies();  **if** (cookies != **null**) {  **for** (Cookie cookie : cookies) {  **if** (***ATT\_NAME\_USER\_NAME***.equals(cookie.getName())) {  **return** cookie.getValue();  }  }  }  **return** **null**;  } |
|  | HashMD5  (Nguồn: stack overflow)  (file MathFunction) | Mã hóa mật khẩu thành chuỗi MD5 | **public** **static** String HashMD5(String md5) {  **try** { java.security.MessageDigest md = java.security.MessageDigest.*getInstance*("MD5");  **byte**[] array = md.digest(md5.getBytes());  StringBuffer sb = **new** StringBuffer();  **for** (**int** i = 0; i < array.length; ++i) {  sb.append(Integer.*toHexString*((array[i] & 0xFF) | 0x100).substring(1, 3));  }  **return** sb.toString();  } **catch** (java.security.NoSuchAlgorithmException e) {  }  **return** **null**;  } |
| Hiển thị thông tin cá nhân | LayThongTinHocSinh | Lấy thông tin cảu một học sinh | **public** **static** ThongTinTK LayThongTin(Connection conn, String TenTK) **throws** SQLException {  String sql = "select TenNguoiDung,NgaySinh,DiaChi,SDT from TaiKhoan where TenTK = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, TenTK);  ResultSet rs = pstm.executeQuery();  ThongTinTK tk = **new** ThongTinTK();  **if** (rs.next()) {  String name = rs.getString("TenNguoiDung");  String ngaysinh = rs.getString("NgaySinh");  String diachi = rs.getString("DiaChi");  String sdt = rs.getString("SDT");tk.setTenTK(TenTK);  tk.setTenNguoiDung(name); tk.setNgaySinh(ngaysinh);  tk.setDiaChi(diachi);  tk.setSDT(sdt);  }  **return** tk;} |
|  | LayDSLopHoc | Lấy danh sách lớp học của học sinh | **public** **static** List<LopHoc> LayDSLopHoc(Connection conn, String TenTK) **throws** SQLException {  String sql = "execute pr\_DSLopHoc'" + TenTK + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<LopHoc> list = **new** ArrayList<LopHoc>();  **while** (rs.next()) {  String malop = rs.getString("MaLop");  String tenlop = rs.getString("TenLop");  LopHoc lh = **new** LopHoc();  lh.setMaLop(malop);  lh.setTenLop(tenlop);  list.add(lh);  }  **return** list;  } |
| Thi | KiemTraDuocPhepThi | Kiểm tra một học sinh có quyền thi hay không khi họ có được Url bài thi | **public** **static** LuotThi KiemTraDuocPhepThi(Connection conn, String tenTK, String maDe, String maLop)  **throws** SQLException {  String sql = "pr\_KiemTraDuocPhepThi '" + tenTK + "','" + maDe + "','" + maLop + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  **if** (rs.next()) {  LuotThi lt = **new** LuotThi();  lt.setMaDe(maDe);  lt.setMaLop(maLop);  String tieuDe = rs.getString("TieuDe");  **int** thoiLuong = rs.getInt("ThoiLuong");  Timestamp batDau = rs.getTimestamp("BatDau");  lt.setTieuDe(tieuDe);  lt.setThoiLuong(thoiLuong);  lt.setBatDau(batDau);  **return** lt;  }  **return** **null**;  } |
|  | LayNoiDungDeThi | Lấy toàn bộ nội dung của đề thi | **public** **static** List<ND\_DeThi> LayDeThi(Connection conn, String MaDe) **throws** SQLException {  String sql = "execute pr\_LayDeThi '" + MaDe + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<ND\_DeThi> list = **new** ArrayList<ND\_DeThi>();  **while** (rs.next()) {  **int** maCauHoi = rs.getInt("MaCauHoi");  String noiDung = rs.getString("NoiDung");  String dapAnA = rs.getString("DapAnA");  String dapAnB = rs.getString("DapAnB");  String dapAnC = rs.getString("DapAnC");  String dapAnD = rs.getString("DapAnD");  String dapAnDung = rs.getString("DapAnDung");  ND\_DeThi ch = **new** ND\_DeThi();  ch.setMaCauHoi(maCauHoi);  ch.setNoiDung(noiDung);  ch.setDapAnA(dapAnA);  ch.setDapAnB(dapAnB);  ch.setDapAnC(dapAnC);  ch.setDapAnD(dapAnD);  ch.setDapAnDung(dapAnDung);  list.add(ch);}  **return** list;  } |
|  | LayDapAn | Lấy đáp án của đề thi | **public** **static** String LayDapAn(Connection conn, String MaDe) **throws** SQLException {  String sql = "execute pr\_LayDapAn '" + MaDe + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  String dapAn = "";  **while** (rs.next()) {  dapAn = rs.getString("DapAn");  }  **return** dapAn;  } |
|  | ChamDiem | Chấm điểm bài thi | **public** **static** **float** ChamDiem(String baiThi, String dapAn) {  baiThi = baiThi.toUpperCase();  dapAn = dapAn.toUpperCase();  **float** diem = 0;  **float** dem = 0;  **float** dodaidapan = dapAn.length();  **float** dodaibaithi = baiThi.length();  **if** (dodaidapan == dodaibaithi) {  **for** (**int** i = 0; i < dodaibaithi; i++)  **if** (baiThi.charAt(i) == dapAn.charAt(i))  dem++;  } **else**  **return** -1;  diem = (dem / dodaibaithi) \* 10;  **return** diem;  } |
|  | LuuKetQuaThiTamThoi | Lưu tạm thời một kết quả bài thi ngay sau khi học sinh bấm làm bài, nếu học sinh cố tình không nộp bài (Bằng cách tắt máy hoặc tắt mạng) họ sẽ nhận điểm 0 | **public** **static** **void** LuuKetQuaThiTamThoi (Connection conn, String tenTk, String lop, String deThi, Timestamp ngayLam ) **throws** SQLException  {  String sql = "insert into KetQuaThi values(?,?,?,?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, tenTk);  pstm.setString(2, lop);  pstm.setString(3, deThi);  pstm.setTimestamp(4, ngayLam);  pstm.setFloat(5, 0);  pstm.executeUpdate();  } |
|  | LuuKetQuaThi | Lưu lai kết quả bài thi sau khi chấm điểm | **public** **static** **void** LuuKetQuaThi(Connection conn, String tenTk, String lop, String deThi, Timestamp ngayLam,**float** diem) **throws** SQLException  {  String sql = "update KetQuaThi set Diem = ?\r\n" +  "where TenTK = ? and MaLop = ? and MaDeThi = ? and NgayLamBai = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setFloat(1, diem);  pstm.setString(2, tenTk);  pstm.setString(3, lop);  pstm.setString(4, deThi);  pstm.setTimestamp(5, ngayLam);  pstm.setFloat(5, 0);  pstm.executeUpdate();  } |
| Đăng xuất | doGet file DangXuat.java | Hủy bỏ phiên hiện tại và điều hướng về trang chủ | **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  HttpSession session = request.getSession();  session.invalidate();  response.sendRedirect(request.getContextPath() + "/Home");  } |
| Quản lí học sinh | LayDanhSachHocSinh (file HOCSINH\_DAO.java) | Lấy danh sách gồm n học sinh (theo trang) | **public** **static** List<ThongTinTK> LayDanhSachHocSinh(Connection conn, **int** page) **throws** SQLException {  String sql = "execute pr\_LayDanhSachHocSinh";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<ThongTinTK> list = **new** ArrayList<ThongTinTK>();  **int** max = page \* *np*;  **int** min = (page - 1) \* *np*;  **int** i = 0;  **while** (rs.next() && i < max) {  **if** (i >= min) {  String tentk = rs.getString("TenTK");  String name = rs.getString("TenNguoiDung");  String ngaysinh = rs.getString("NgaySinh");  String diachi = rs.getString("DiaChi");  String sdt = rs.getString("SDT");  **int** soluonglop = rs.getInt("SoLuong");  ThongTinTK tk = **new** ThongTinTK();  tk.setTenTK(tentk);  tk.setTenNguoiDung(name);  tk.setNgaySinh(ngaysinh);  tk.setDiaChi(diachi);  tk.setSDT(sdt);  tk.setSoLuongLopHoc(soluonglop);  list.add(tk);  }  i++;  }  **return** list;  } |
|  | TimKiemHocSinh | Tìm kiếm 1 học sinh theo từ khóa | **public** **static** List<ThongTinTK> TimKiemHocSinh(Connection conn, String key) **throws** SQLException {  String sql = "exec pr\_TimKiemTaiKhoan'" + key + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<ThongTinTK> list = **new** ArrayList<ThongTinTK>();  **while**(rs.next())  {  String tentk = rs.getString("TenTK");  String name = rs.getString("TenNguoiDung");  String ngaysinh = rs.getString("NgaySinh");  String diachi = rs.getString("DiaChi");  String sdt = rs.getString("SDT");  **int** soluonglop = rs.getInt("SoLuong");  ThongTinTK tk = **new** ThongTinTK();  tk.setTenTK(tentk);  tk.setTenNguoiDung(name);  tk.setNgaySinh(ngaysinh);  tk.setDiaChi(diachi);  tk.setSDT(sdt);  tk.setSoLuongLopHoc(soluonglop);  list.add(tk);  }  **return** list;  } |
|  | LayDSHocSinhDaVaoLop | Lấy danh sách thành viên hiện tại của một lớp | **public** **static** List<HocSinh> LayDSHocSinhDaVaoLop(Connection conn, String maLop) **throws** SQLException {  String sql = "exec pr\_HocSinhDaVaoLop'" + maLop + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<HocSinh> list = **new** ArrayList<>();  **while** (rs.next()) {  String tenTK = rs.getString("TenTK");  String tenHocSinh = rs.getString("TenNguoiDung");  HocSinh hs = **new** HocSinh();  hs.setTenTK(tenTK);  hs.setTenHocSinh(tenHocSinh);  list.add(hs);  }  **return** list;  } |
|  | LayDSHocSinhChuaVaoLop | Lấy danh sách học sinh chưa phải là thành viên của một lớp (dùng cho chức năng them hoc sinh vào lớp) | **public** **static** List<HocSinh> LayDSHocSinhDaVaoLop(Connection conn, String maLop) **throws** SQLException {  String sql = "exec pr\_HocSinhDaVaoLop'" + maLop + "'";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<HocSinh> list = **new** ArrayList<>();  **while** (rs.next()) {  String tenTK = rs.getString("TenTK");  String tenHocSinh = rs.getString("TenNguoiDung");  HocSinh hs = **new** HocSinh();  hs.setTenTK(tenTK);  hs.setTenHocSinh(tenHocSinh);  list.add(hs);  }  **return** list;  } |
|  | SuaTaiKhoan | Sửa thông tin của một tài khoản | **public** **static** **void** SuaTaiKhoan(Connection conn,String tenTK, String tenHS,String ngaySinh, String diaChi, String sdt) **throws** SQLException  {  String sql = "Update TaiKhoan \r\n" +  "set TenNguoiDung = ?, NgaySinh = ?, DiaChi = ?, SDT = ?\r\n" +  "where TenTK = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, tenHS);  pstm.setString(2, ngaySinh);  pstm.setString(3, diaChi);  pstm.setString(4, sdt);  pstm.setString(5, tenTK);  pstm.executeLargeUpdate();  } |
|  | ThemHocSinh | Thêm một học sinh | **public** **static** **void** ThemHocSinh(Connection conn, ThongTinTK tk) **throws** SQLException {  String sql = "INSERT INTO TaiKhoan VALUES (?,?,?,?,?,?,1)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, tk.getTenTK());  pstm.setString(2, tk.getMatKhau());  pstm.setString(3, tk.getTenNguoiDung());  pstm.setString(4, tk.getNgaySinh());  pstm.setString(5, tk.getDiaChi());  pstm.setString(6, tk.getSDT());  pstm.executeUpdate();  } |
|  | XoaHocSinh | Xóa tài khoản của một học sinh | **public** **static** **void** XoaTaiKhoan(Connection conn, String tenTK) **throws** SQLException {  String sql = "DeLete from TaiKhoan where tenTK = (?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, tenTK);  pstm.executeUpdate();  } |
| Quản Lý Lớp Học  Các hàm chính nằm ở file LOPHOC\_DAO.java | LayDSLopHoc | Lấy danh sách của toàn bô lớp học | **public** **static** List<LopHoc> LayDSLopHoc(Connection conn) **throws** SQLException {  String sql = "exec pr\_LayDSlopHoc";  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  List<LopHoc> list = **new** ArrayList<LopHoc>();  **while** (rs.next()) {  String malop = rs.getString("MaLop");  String tenlop = rs.getString("TenLop");  Date ngayKetThuc = rs.getDate("NgayKetThuc");  **int** soHocSinh = rs.getInt("SoLuong");  LopHoc lh = **new** LopHoc();  lh.setMaLop(malop);  lh.setTenLop(tenlop);  lh.setNgayKetThuc(ngayKetThuc);  lh.setSoHocSinh(soHocSinh);  list.add(lh);  }  **return** list;  } |
|  | XoaLopHoc | Xóa một lớp học ra khỏi danh sách | **public** **static** **void** XoaLopHoc(Connection conn, String maLop) **throws** SQLException {  String sql = "DeLete from LopHoc where MaLop = (?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maLop);  pstm.executeUpdate();  } |
|  | ThemLopHoc | Thêm một lớp học mới | **public** **static** **void** ThemLopHoc(Connection conn, String tenLop, String ngayKetThuc)  **throws** SQLException, ParseException {  String maLop = "LH" + Long.*toString*(System.*currentTimeMillis*());  String sql = "insert into LopHoc values(?,?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maLop);  pstm.setString(2, tenLop);  pstm.setString(3, ngayKetThuc);  pstm.executeUpdate();  } |
|  | ThemVaoLop | Thêm học sinh vào lớp học | **public** **static** **void** ThemVaoLop(Connection conn, String maLop, String tenTK) **throws** SQLException {  String sql = "Insert into TV\_LopHoc values (?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maLop);  pstm.setString(2, tenTK);  pstm.executeUpdate(); |
|  | XoaKhoiLop | Xóa một học sinh khỏi lớp học | **public** **static** **void** ThemVaoLop(Connection conn, String maLop, String tenTK) **throws** SQLException {  String sql = "Insert into TV\_LopHoc values (?,?)";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maLop);  pstm.setString(2, tenTK);  pstm.executeUpdate();  } |
| Quản Lý đề thi  (DETHI\_PLUS\_DAO.java) | LayDSDeThi | Lấy danh sách đề thi | **public** **static** List<DeThi> LayDSDeThi(Connection conn,**int** page) **throws** SQLException {  String sql = "select \* from DeThi";  List<DeThi> list = **new** ArrayList<>();  PreparedStatement pstm = conn.prepareStatement(sql);  ResultSet rs = pstm.executeQuery();  **int** max = page \* *np*;  **int** min = (page-1)\**np*;  **int** i=0;  **while** (rs.next()&&i<max) {  **if**(i>=min) {  String maDeThi = rs.getString("MaDeThi");  String maMonHoc = rs.getString("maMonHoc");  **int** soCauDe = rs.getInt("SoCauDe");  **int** soCauKho = rs.getInt("SoCauKho");  **int** soCauTrungBinh = rs.getInt("SoCauTrungBinh");  DeThi dt = **new** DeThi();  dt.setMaDeThi(maDeThi);  dt.setMaMonHoc(maMonHoc);  dt.setSoCauDe(soCauDe);  dt.setSoCauKho(soCauKho);  dt.setSoCauTrungBinh(soCauTrungBinh);  list.add(dt);  }  i++;  }  **return** list;  } |
|  | TaoDeThi | Tạo ra một bộ câu hỏi ngẫu nhiên từ một môn học có chung cấp độ | **private** **static** List<ND\_DeThi> TaoDeThi(Connection conn, **int** capDo, **int** soLuong, String maMonHoc)  **throws** SQLException {  List<ND\_DeThi> rsList = **new** ArrayList<>();  **if** (soLuong > 0) {  Random rd = **new** Random();  String sql = "select count(\*) as SoLuong from CauHoi where CapDo = ? and MonHoc = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setInt(1, capDo);  pstm.setString(2, maMonHoc);  ResultSet rs = pstm.executeQuery();  **if** (rs.next()) {  **int** soLuongSanCo = rs.getInt("SoLuong");  **if** (soLuongSanCo < soLuong) {  **if** (capDo == 1)  **throw** **new** ArithmeticException(  "Số câu hỏi dễ bạn chọn vượt quá mức cung câp, số lượng tối đa hiện tại: "  + soLuongSanCo);  **else** **if** (capDo == 2)  **throw** **new** ArithmeticException(  "Số câu hỏi trung bình bạn chọn vượt quá mức cung câp, số lượng tối đa hiện tại: "  + soLuongSanCo);  **else** **if** (capDo == 3)  **throw** **new** ArithmeticException(  "Số câu hỏi khó bạn chọn vượt quá mức cung câp, số lượng tối đa hiện tại: "  + soLuongSanCo);  } **else** {  sql = "select MaCauHoi,NoiDung,DapAnA,DapAnB,DapAnC,DapAnD,DapAnDung from CauHoi where CapDo = ? and CauHoi.MonHoc = ?";  pstm = conn.prepareStatement(sql);  pstm.setInt(1, capDo);  pstm.setString(2, maMonHoc);  rs = pstm.executeQuery();  List<ND\_DeThi> tmpList = **new** ArrayList<>();  **while** (rs.next()) {  **int** maCauHoi = rs.getInt("MaCauHoi");  String noiDung = rs.getString("NoiDung");  String dapAnA = rs.getString("DapAnA");  String dapAnB = rs.getString("DapAnB");  String dapAnC = rs.getString("DapAnC");  String dapAnD = rs.getString("DapAnD");  String dapAnDug = rs.getString("DapAnDung");  ND\_DeThi ch = **new** ND\_DeThi();  ch.setMaCauHoi(maCauHoi);  ch.setNoiDung(noiDung);  ch.setDapAnA(dapAnA);  ch.setDapAnB(dapAnB);  ch.setDapAnC(dapAnC);  ch.setDapAnD(dapAnD);  ch.setDapAnDung(dapAnDug);  tmpList.add(ch);  }  **int** i;  **while** (rsList.size() < soLuong) {  i = rd.nextInt(tmpList.size());  rsList.add(tmpList.get(i));  tmpList.remove(i);  }  }  }  }  **return** rsList;  } |
|  | TaoDeThi  (hàm cùng tên) | Tạo một đề vợi các cấp độ câu hỏi | **public** **static** List<ND\_DeThi> TaoDeThi(Connection conn, String maMonHoc, **int** soCauDe, **int** soCauTb, **int** soCauKho)  **throws** SQLException {  List<ND\_DeThi> rsList = **new** ArrayList<>();  rsList.addAll(*TaoDeThi*(conn, 1, soCauDe, maMonHoc));  rsList.addAll(*TaoDeThi*(conn, 2, soCauTb, maMonHoc));  rsList.addAll(*TaoDeThi*(conn, 3, soCauKho, maMonHoc));  **return** rsList;  } |
|  | LayThongTinDeThi | Lấy thông tin đề thi | **public** **static** DeThi LayThongTinDeThi(Connection conn, String maDe) **throws** SQLException {  String sql = "select \* from DeThi where MaDeThi = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  ResultSet rs = pstm.executeQuery();  **if** (rs.next()) {  DeThi dt = **new** DeThi();  dt.setMaMonHoc(rs.getString("MaMonHoc"));  dt.setSoCauDe(rs.getInt("SoCauDe"));  dt.setSoCauTrungBinh(rs.getInt("SoCauTrungBinh"));  dt.setSoCauKho(rs.getInt("SoCauKho"));  **return** dt;  }  **return** **null**;  } |
|  | LayNoiDungDeThi | Lấy nội dung của đề thi | Đã có ở chức năng Thi |
|  | LayDSLopDaGanDe | Lấy danh sách những lớp đã được gán đề (bao gồm các thông tin về thời lượng làm bài, số lần làm bài, ngày đóng, mở) | **public** **static** List<BG\_DeThi> LayDSLopDaGanDeThi(Connection conn, String maDe) **throws** SQLException {  List<BG\_DeThi> list = **new** ArrayList<>();  String sql = "select LopHoc.MaLop, TenLop, ThoiLuong, TieuDe, TGBatDau, TGKetThuc,SoLanLamBai\r\n"  + "from BG\_DeThi, LopHoc \r\n" + "where MaDeThi = ? and LopHoc.MaLop = BG\_DeThi.MaLop";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, maDe);  ResultSet rs = pstm.executeQuery();  **while** (rs.next()) {  String tieuDe = rs.getString("TieuDe");  String maLop = rs.getString("MaLop");  String tenLop = rs.getString("TenLop");  **int** thoiLuong = rs.getInt("ThoiLuong");  String tGBatDau = rs.getString("TGBatDau");  String tGKetThuc = rs.getString("TGketThuc");  **int** soLanLamBai = rs.getInt("soLanLamBai");  BG\_DeThi bg = **new** BG\_DeThi();  bg.setTieuDe(tieuDe);  bg.setMaLop(maLop);  bg.setTenLop(tenLop);  bg.setThoiLuong(thoiLuong);  bg.setTGBatDau(tGBatDau);  bg.setTGKetThuc(tGKetThuc);  bg.setSoLanLamBai(soLanLamBai);  list.add(bg);  }  **return** list;  } |
|  | GiaoDeThi | Gán đề thi cho một lớp | **public** **static** **void** GiaoDeThi(Connection conn, BG\_DeThi bg) **throws** SQLException, ParseException {  String sql = "INSERT INTO BG\_DeThi VALUES (?,?,?,?,?,?,?)";  DateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd'T'HH:mm");  java.util.Date tmp1 = dateFormat.parse(bg.getTGBatDau());  java.util.Date tmp2 = dateFormat.parse(bg.getTGKetThuc());  Timestamp bd = **new** Timestamp(tmp1.getTime());  Timestamp kt = **new** Timestamp(tmp2.getTime());  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, bg.getMaDeThi());  pstm.setString(2, bg.getMaLop());  pstm.setTimestamp(3, bd);  pstm.setTimestamp(4, kt);  pstm.setInt(5, bg.getThoiLuong());  pstm.setString(6, bg.getTieuDe());  pstm.setInt(7, bg.getSoLanLamBai());  pstm.executeUpdate();  } |
|  | GiaoLaiDeThi | Cập nhật việc gán đề thi cho một lớp (sửa thời gian làm bài, số làn làm bài, ngày đóng, mở) | **public** **static** **void** GiaoLaiDeThi(Connection conn, BG\_DeThi bg) **throws** SQLException, ParseException{  String sql = "Update BG\_DeThi\r\n" +  "set TieuDe=?, ThoiLuong = ?,SoLanLamBai=?, TGBatDau = ?, TGKetThuc=?\r\n" +  "where MaLop = ? and MaDeThi =?";  DateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd'T'HH:mm");  java.util.Date tmp1 = dateFormat.parse(bg.getTGBatDau());  java.util.Date tmp2 = dateFormat.parse(bg.getTGKetThuc());  Timestamp bd = **new** Timestamp(tmp1.getTime());  Timestamp kt = **new** Timestamp(tmp2.getTime());  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, bg.getTieuDe());  pstm.setInt(2, bg.getThoiLuong());  pstm.setInt(3, bg.getSoLanLamBai());  pstm.setTimestamp(4, bd);  pstm.setTimestamp(5, kt);  pstm.setString(6, bg.getMaLop());  pstm.setString(7, bg.getMaDeThi());  pstm.executeUpdate();  } |
|  | DongDeThi | Đóng đề thi đối với một lớp | **public** **static** **void** DongDeThi(Connection conn, String deThi, String lop) **throws** SQLException  {  String sql = "delete from BG\_DeThi where MaDeThi = ? and MaLop = ?";  PreparedStatement pstm = conn.prepareStatement(sql);  pstm.setString(1, deThi);  pstm.setString(2, lop);  pstm.executeUpdate();  } |
| Quản lý câu hỏi  CAUHOI\_DAO.java | xemDSCauHoi | Lấy nội danh sách câu hỏi theo từng trang | **public** List<CauHoi> xemDSCauHoi(**int** page) **throws** SQLException {  List<CauHoi> ds = **new** ArrayList<CauHoi>();  Statement st = conn.createStatement();  String sql = "select \* from CauHoi";  ResultSet rs = st.executeQuery(sql);  **int** max = page \* np;  **int** min = (page-1)\*np;  **int** i=0;  **while** (rs.next() && i < max) {  **if**(i>=min)  {  CauHoi ch = **new** CauHoi();  ch.setMaCauHoi(rs.getInt("MaCauHoi"));  ch.setNoiDung(rs.getString("NoiDung"));  ch.setDapAnA(rs.getString("DapAnA"));  ch.setDapAnB(rs.getString("DapAnB"));  ch.setDapAnC(rs.getString("DapAnC"));  ch.setDapAnD(rs.getString("DapAnD"));  ch.setDapAnDung(rs.getString("DapAnDung"));  ch.setCapDo(rs.getInt("CapDo"));  ch.setMonHoc(rs.getString("MonHoc"));  ds.add(ch);  }  i++;  }  **return** ds;  } |
|  | suaCauHoi | Sửa nội dung, đáp án, cấp độ, môn học, của câu hỏi | **public** **void** suaCauHoi(String MaCauHoi, String NoiDung, String DapAnA, String DapAnB, String DapAnC, String DapAnD,  String DapAnDung, **int** CapDo, String MonHoc) **throws** SQLException {  String sql = "update CauHoi set NoiDung = ?, DapAnA = ?, DapAnB = ?, DapAnC = ?, DapAnD =?,\n"  + "DapAnDung = ?,CapDo = ?, MonHoc = ? where MaCauHoi = ?";  CallableStatement cstm = conn.prepareCall(sql);  cstm.setString(1, NoiDung);  cstm.setString(2, DapAnA);  cstm.setString(3, DapAnB);  cstm.setString(4, DapAnC);  cstm.setString(5, DapAnD);  cstm.setString(6, DapAnDung);  cstm.setInt(7, CapDo);  cstm.setString(8, MonHoc);  **int** maCauHoi = Integer.*parseInt*(MaCauHoi);  cstm.setInt(9, maCauHoi);  cstm.executeUpdate();  } |
|  | xoaCauHoi | Xóa một câu hỏi | **public** **void** xoaCauHoi(**int** MaCauHoi) **throws** SQLException {  String sql = "delete from CauHoi where MaCauHoi = ?";  CallableStatement cstm = conn.prepareCall(sql);  cstm.setInt(1, MaCauHoi);  cstm.executeUpdate();  } |
|  | ThemCauHoi | Thêm một câu hỏi | **public** **void** themCauHoi(String NoiDung, String DapAnA, String DapAnB, String DapAnC, String DapAnD, String DapAnDung,  **int** CapDo, String MonHoc) **throws** SQLException {  String sql = "insert into CauHoi values (?,?,?,?,?,?,?,?)";  CallableStatement cstm = conn.prepareCall(sql);  cstm.setString(1, NoiDung);  cstm.setString(2, DapAnA);  cstm.setString(3, DapAnB);  cstm.setString(4, DapAnC);  cstm.setString(5, DapAnD);  cstm.setString(6, DapAnDung);  cstm.setInt(7, CapDo);  cstm.setString(8, MonHoc);  cstm.executeUpdate();  } |

Chương 3. Lỗi và biện pháp khắc phục (Nháp)

1. XXS

**1.1 Giới thiệu**

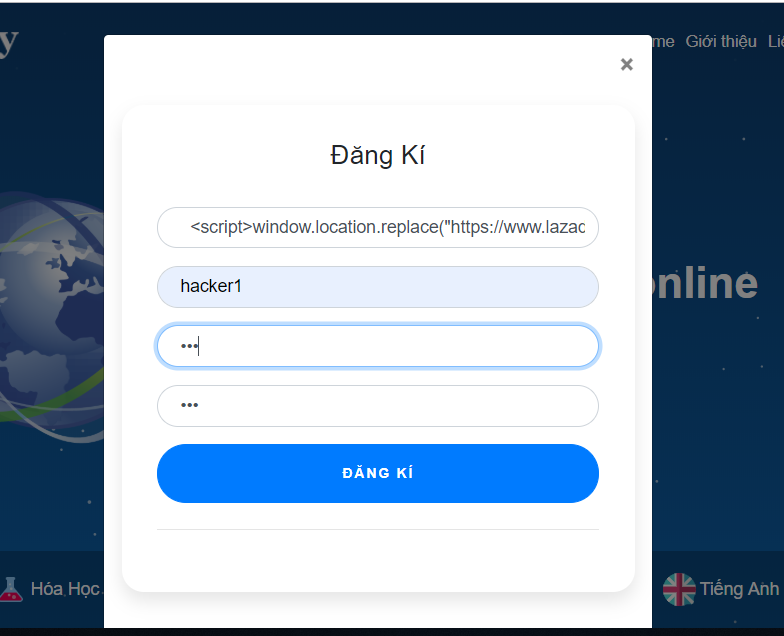
Cross-Site Scripting hay còn được gọi tắt là XSS là một kĩ thuật tấn công bằng cách chèn vào các website động (ASP, PHP, CGI, JSP ...) những thẻ HTML hay những đoạn mã script nguy hiểm có thể gây nguy hại cho những nạn nhân sử dụng.

Lỗi này xảy ra khi ứng dụng web thu nhận các dữ liệu nguy hiểm được nhập từ hacker. Một website thường chứa các link, thông qua các link này hacker có thể chèn các đoạn code vào và khi người dùng nào đó sử dụng link này thì coi như 99% là chết, hacker có thể thông qua lỗi này để chèn code vào site hay link để lấy các thông tin quan trọng từ nạn nhân

Phụ thuộc vào mục đích của hacker, những đoạn Javascript được chèn vào để lấy những thông tin như:

* Cookie: hacker có thể lấy được cookie của người dùng và dùng những thông tin trong cookie để giả mạo phiên truy cập hoặc lấy những thông tin nhạy cảm khác được lưu trong cookie.
* Keylogging: hacker có thể ghi lại những thao tác gõ phím của người dùng bằng cách sử dụng sự kiện trong Javascript và gửi tất cả những thao tác gõ phím đó về cho hắn để thực hiện những mục đích như đánh cắp các thông tin nhạy cảm, lấy mật khẩu truy cập website hoặc mã số thẻ tín dụng...
* Phishing: hacker có thể thay đổi giao diện của website bằng cách thay đổi cấu trúc HTML trong trang web để đánh lừa người dùng. Hacker có thể tạo ra những form đăng nhập giả nhằm lừa người dùng đăng nhập vào để đánh cắp mật khẩu.
  1. **XSS trong website Thi Trắc Nghiệm**

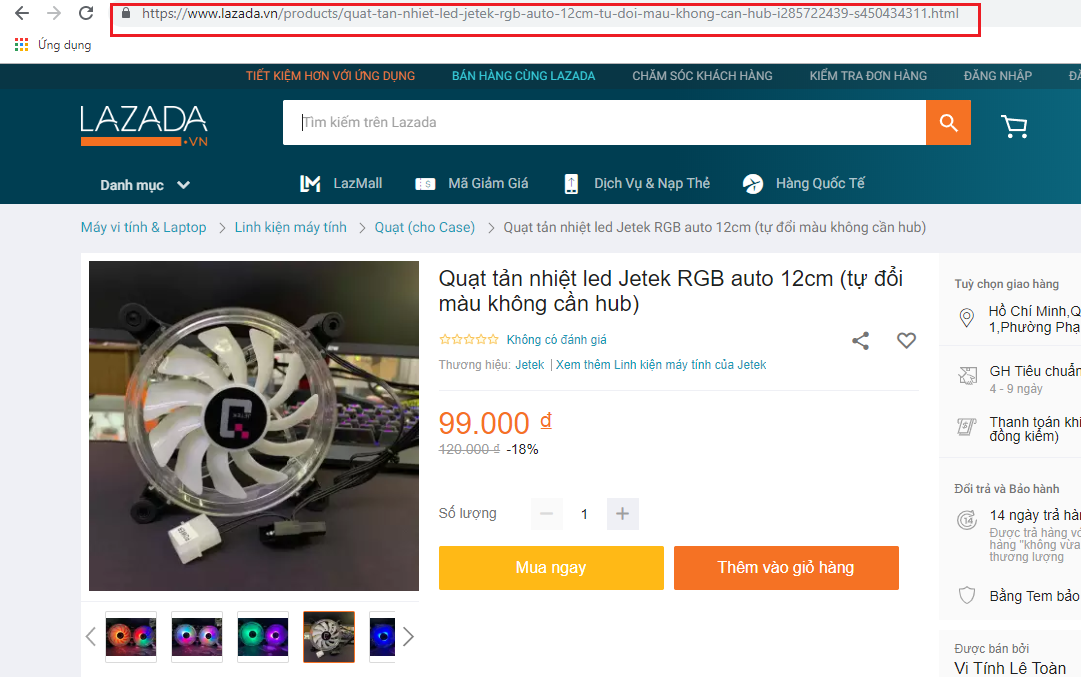
Tại trang đăng kí, ở trường Tên của bạn, hacker có thể chèn các đoạn Script, các đoạn script này sẽ được lưu trữ trong CDSL và trả về cho người dùng khác khi họ truy cập vào trang web.



Ở đây, người dùng có tên đăng nhập hacker1, đã tiêm đoạn script

<script>window.location.replace("https://www.lazada.vn/products/quat-tan-nhietb-i285722439-s450434311.html");</script>

Mã javascipt này khiến người dùng khác (Người quản lí học sinh) tự động chuyển hướng trang sang <https://www.lazada.vn/products/quat-tan-nhiet-led-jetek-rgb-auto-12cm-tu-doi-mau-khong-can-hub-i285722439-s450434311.html> - một trang bán hàng của kẻ tấn công



Tương tự như vậy, hacker có thể chèn các đoạn script vào các trường nhập liêu của website,

Các trường có thể bị tấn công bao gồm: Trường nhập tên đăng kí, tạo câu hỏi, tạo đề thi, lớp học... bla bla.

* 1. **Cách phòng chống**

Loại bỏ các thẻ HTML ra khỏi dữ liệu đầu vào của người dùng

Loại bỏ các thẻ HTML trong các dữ liệu chuỗi trước khi trả về cho người dùng

1. SQL injection

**2.1 Giới thiệu**

* 1. **SQL injection trong website Thi Trắc Nghiệm**

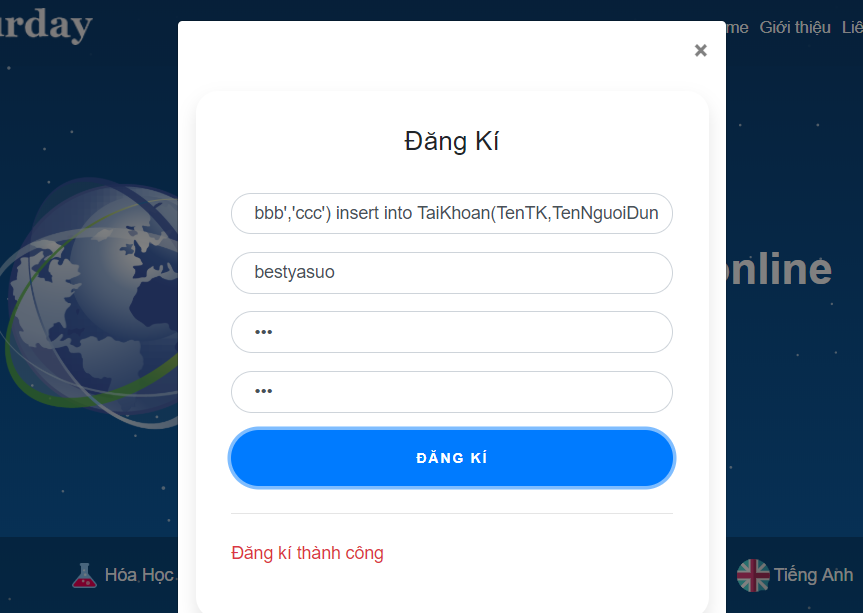
Tại hàm xử lí việc đăng kí tài khoản, sau khi nhận dữ liệu từ người dùn, máy chủ sẽ thực hiện đoạn truy vấn tạo tài khoản mới như sau:

"Insert into TaiKhoan (TenTK,TenNguoiDung,MatKhau) values ('" + tenTk + "','" + ten +"','" + matKhau + "')";

Trong đó, tenTK, ten, matKhau ứng với dữ liệu được nhập

từ người dùng ở các trường “Tên tài khoản”, “Tên của bạn” và “Mật khẩu”,

hacker có thé tiêm các đoạn mã SQL vào một trong các trường trên:



Ở đây, hacker đã nhập đoạn mã sau vào trường tên của bạn:

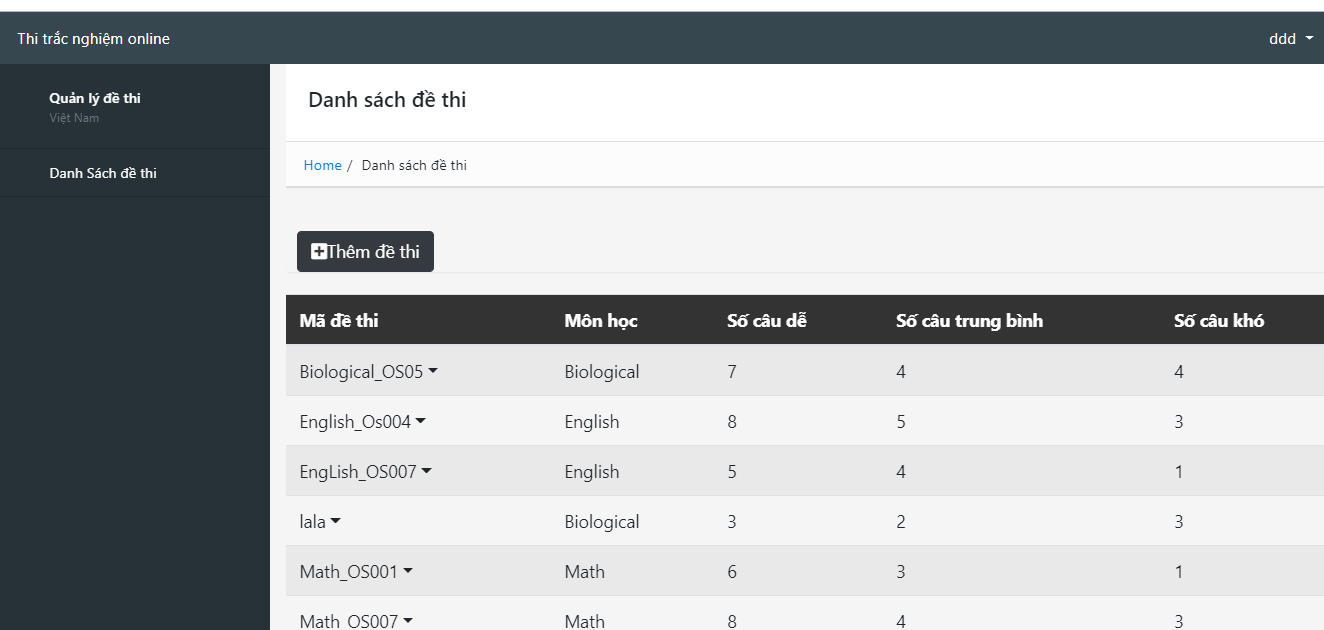
bbb','ccc') insert into TaiKhoan(TenTK,TenNguoiDung,Quyen,MatKhau) values ('hacker2','ddd','3

khi đó, chuỗi truy vấn máy chủ thực hiện sẽ trở thành:

Insert into TaiKhoan (TenTK,TenNguoiDung,MatKhau) values ('bestyasuo','bbb','ccc')

insert into TaiKhoan(TenTK,TenNguoiDung,Quyen,MatKhau) values ('hacker2','ddd','3','123')

chuỗi truy vấn trên sẽ tiến hành tạo 2 tài khoản mới, một tài khoản với tên đăng nhập là bestyasuo với quyền là 1 (học sinh – mặc đinh), và một tài khoản có tên đăng nhập là hacker2 với quyền là 3 (người quản lí đề thi),

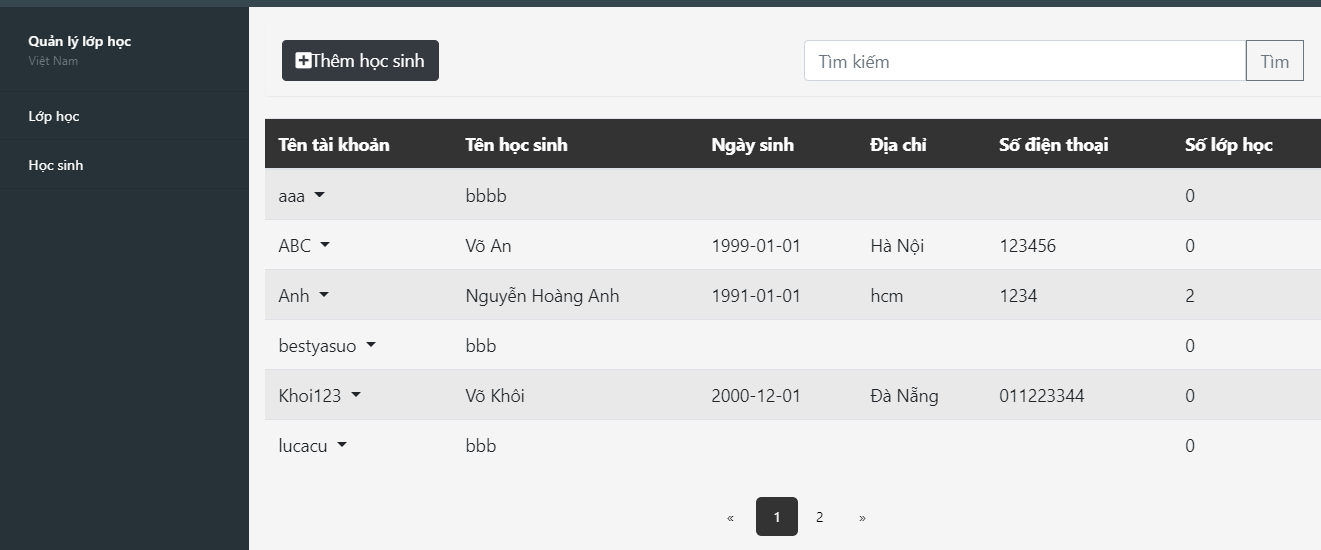


Như vậy hacker đã có được một tài khoản với quyền vượt mức cho phép,

1. Access Control

3.1 Giới thiệu

3.2 Lỗi access control trong website thi trắc nghiện



Hệ thống chỉ cho phép người quản lí học sinh có quyền xem,sửa, thêm, xóa học sinh, và lớp học.

tuy nhiên bằng cách thay đổi các Parameter trong request, người quản lí học sinh có thể thay đổi tài khoản của các quản lí khác (Người quan lí đề thi, người quản lí câu hỏi).

Ví dụ: Để xóa người quản lí đề thi có tên đăng nhập hacker2 (đã tạo ở phần trước), người quản lí học sinh chỉ cần thay đổi Url từ:

<http://localhost:8080/ThiOnLine/XoaHocSinh?tenTK=hocsinh1>

thành

<http://localhost:8080/ThiOnLine/XoaHocSinh?tenTK=hacker2>