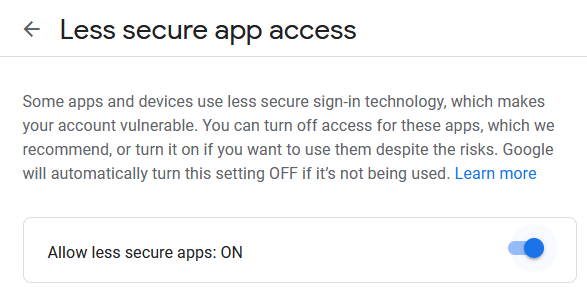
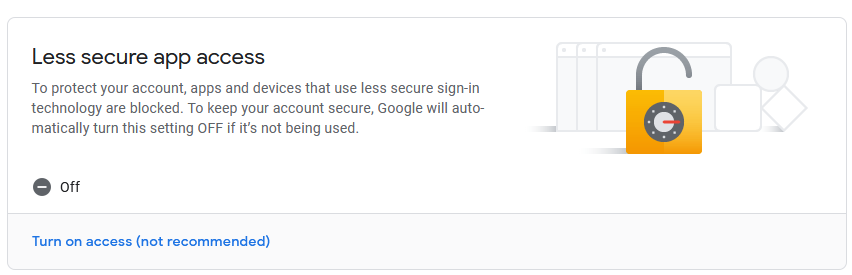
# CPRG-352 - Web Application Programming

## Lab 9

## Topic: Email

In this lab, we will extend the lecture demo one step further by implementing one of two techniques to handle a “forgotten user password”. The first technique emails the password to the user. The second technique emails the user a link which will allow the user to change their password. You can choose which technique you wish to implement.

You should first create a new Gmail account and change the credentials in web.xml. In order for your application to send emails, you will need to enable “Less secure app access” in the Google Account. Navigate to <https://myaccount.google.com/security> and “Turn on access”.



If are you unable to create a new Gmail account for some reason, please contact your instructor.

If you want to use an existing personal Gmail account, be very careful not to commit and push your password from the web.xml. Delete the repository if you make the mistake. (If you are interested in security, here is one tool for scanning public repos on GitHub for sensitive credentials: <https://github.com/techgaun/github-dorks> )

## Technique #1

In this technique, you will send the user their password.

1. Create a new email template. Here is a suggested message.

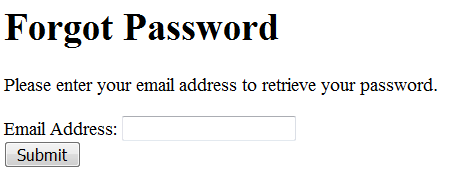
Hi {{firstname}} {{lastname}},

Here are your credentials to log back into the Notes App.

Email address: {{email}}

Password: {{password}}

1. Add a “Forgot password?” link in login.jsp to /forgot.
2. Add a servlet called ForgotPasswordServlet.java which is mapped to /forgot. Add forgot.jsp.



1. When the user submits an email address, in doPost() call a method in AccountService called forgotPassword(email, path). If forgotPassword() returns true, the email was successfully sent. Nevertheless, for security reasons you should not display a message that states whether the email was sent or not. Instead, the JSP should display the message: “If the address you entered is valid, you will receive an email very soon. Please check your email for your password.”
2. In AccountService, implement the method

boolean forgotPassword(String email, String path)

which first retrieves the user based on the email (from UserDB) and then sends the user an email.

## Technique #2

Storing unencrypted passwords is really bad security. Sending a user an email containing the password in plain text is even worse! If you want to avoid be placed on the website <http://plaintextoffenders.com/>, follow these instructions.

1. Edit the notesdb.sql script. After the `role` field in the `user` table, add a new field called reset\_password\_uuid:

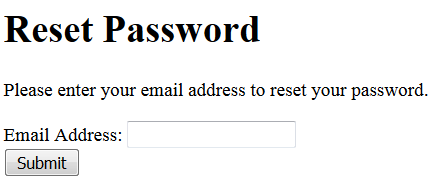
reset\_password\_uuid VARCHAR(50),

1. Regenerate your entity classes.
2. Create a new email template called resetpassword.html. Here is a suggested message.

Hi {{firstname}} {{lastname}},

To reset your password, please click this link <a href="{{link}}">{{link}}</a>

1. Add a “Forgot password” link in login.jsp to /reset
2. Add a servlet called ResetPasswordServlet.java mapped to /reset.   
   To get the URL for the link to pass to resetPassword, use:  
   String url = request.getRequestURL().toString()
3. Add reset.jsp.



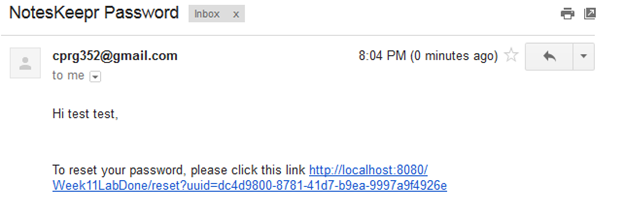
1. Add a new method to AccountService called resetPassword(String email, String path, String url). Create a UUID (universally unique ID) and store in database for the user.

String uuid = UUID.randomUUID().toString();

In the email, send the following link:

String link = url + "?uuid=" + uuid;

1. The user the receives the email and clicks on the link



1. Adjust the doGet() in ResetPasswordServlet to serve resetNewPassword.jsp if there is a UUID parameter. In the form, add a hidden variable for the UUID.



1. After the user submits the form, update the password based on the UUID and new password.
2. Create a new AccountService method called changePassword(). Ensure you clear the UUID from the database.

public boolean changePassword(String uuid, String password) {

UserDB userDB = new UserDB();

try {

User user = userDB.getByUUID(uuid);

user.setPassword(password);

user.setResetPasswordUuid(null);

userDB.update(user);

return true;

} catch (Exception ex) {

return false;

}

}