**password reset**

\* user will enter email address

\* you will find that user based on email address from db

\* if found, you will create password reset code

\* then you will save that code in db and also send to that user's email

\* if this is the correct user, he will have access to his email

\* he can copy that code and paste in the form, enter new password then hit submit

\* on sumit, our server will receive the user's email and code

\* you will try to find the user based on the email and code

\* if found, update his old password with new one and reset code back to empty

\* done \*

+---------------------+

| User |

+---------------------+

|

| Enters email address

|

|

+---------------------+

| System |

+---------------------+

|

| Finds user based on email address

|

|

+---------------------+

| Database |

+---------------------+

|

| Creates password reset code

|

|

+---------------------+

| System |

+---------------------+

|

| Saves code in database

|

| Sends code to user's email address

|

|

+---------------------+

| User's Email |

+---------------------+

|

| User accesses email and retrieves code

|

|

+---------------------+

| User |

+---------------------+

|

| Enters code and new password

|

|

+---------------------+

| System |

+---------------------+

|

| Receives user's email and code

|

|

+---------------------+

| Database |

+---------------------+

|

| Finds user based on email and code

|

|

+---------------------+

| System |

+---------------------+

|

| Updates user's password

|

| Resets password reset code to empty

|

|

+---------------------+

| Database |

+---------------------+

|

| Confirms password has been updated

|

|

+---------------------+

| User |

+---------------------+

|

| Password reset is complete

|

|

+---------------------+

**CSRF token:**

In this sequence diagram, the user sends a request to the client that includes a CSRF token. The client then forwards this request to the server, which verifies the CSRF token against the session or cookie. If the CSRF token is valid, the server processes the request and generates a new CSRF token, which it includes in the response to the client. The client then sends a subsequent request to the server that includes the new CSRF token, and the process repeats.

