VERIFY EMAIL API TUTORIAL
Composed by: Charlie Tan

Local Route: http://localhost:5000/api/verify_email (api/verify_email)
Deployed Route: https://kindling-lp.herokuapp.com/api/verify_email (api/verify_email)

Note Above: had to show (api/verify_email) since the blue line of the autoformat for links makes the underscore hard to see.

EXPECTED INPUT FROM FRONTEND:
{ "email_str" : some_string }

INPUT PROPERTIES EXPLAINED:
1) email_str: the email string that the server is supposed to verify after having sent a verification email to the client as part of the registration process.

EXPECTED OUTPUT FROM BACKEND:
{ "success_bool" : some_boolean }

OUTPUT PROPERTIES EXPLAINED:
1) success_bool: whether or not the verification process was successful. If 'true', the user is verified, 'false' otherwise.

EXPECTED OUTPUT ILLUSTRATED:

1)
-Case: client has a 'ready_status' code of zero before verification.
-Expected output:
{ "success_bool" : true }

-In the above case, the client's 'ready_status' code is set from zero to 1 and is considered verified.

2)
-Case: somehow, the client has a 'ready_status' code of some integer greater than zero before verification.
-Expected output:
{ "success_bool" : true }

-In the above case, the client's 'ready_status' code is left alone as it is. For example, if their 'ready_status' code was already 2 before verification, the API does nothing and leaves the

'ready_status' code as 2. The client is considered verified even though nothing was done since they had to have been verified in the past to reach a 'ready_status' code of greater than zero.

3)
-Case: a client corresponding to 'email_str' provided by the frontend input does not exist in the database.
-Expected output:
{ "success_bool" : false }

4)
-Case: a database error occurs somehow and verification can not proceed.
-Expected output:
{ "success_bool" : false }
