--- all_files_content.txt ---

----- Contents of all_files_content.txt -----

----- Contents of Config.py -----

```
class Config:
                                                    DISCORD_TOKEN                    =
'MTI2OTM4MTE4OTA1NjMzNTk3Mw.Gihcfw.nrq0x-JiL65P0LIQTO-rTyyXq0qC-2PSSBuXr8'
    CHANNEL_ID = 1269383349278081054
    DATABASE_PASSWORD = 'postgres'
```

----- Contents of main.py -----

```
import discord
from discord.ext import commands
from boundary.BotBoundary import BotBoundary
from boundary.HelpBoundary import HelpBoundary
from boundary.AccountBoundary import AccountBoundary
from boundary.BrowserBoundary import BrowserBoundary
from boundary.LoginBoundary import LoginBoundary
from boundary.CloseBrowserBoundary import CloseBrowserBoundary
from boundary.StopBoundary import StopBoundary
from boundary.NavigationBoundary import NavigationBoundary
from boundary.PriceBoundary import PriceBoundary
from boundary.MonitorPriceBoundary import MonitorPriceBoundary
from Config import Config

# Set up the bot's intents
```

```
intents = discord.Intents.default()
intents.message_content = True  # Enable reading message content

# Initialize the bot with the correct command prefix and intents
class MyBot(commands.Bot):
    async def setup_hook(self):
        await self.add_cog(BotBoundary(self))
        await self.add_cog(HelpBoundary(self))
        await self.add_cog(AccountBoundary(self))
        await self.add_cog(BrowserBoundary(self))
        await self.add_cog(StopBoundary(self))
        await self.add_cog(LoginBoundary(self))
        await self.add_cog(CloseBrowserBoundary(self))
        await self.add_cog(NavigationBoundary(self))
        await self.add_cog(PriceBoundary(self))
        await self.add_cog(MonitorPriceBoundary(self))

# Run the bot
if __name__ == "__main__":
    bot = MyBot(command_prefix="!", intents=intents)
    print(f"Bot is starting...")
    bot.run(Config.DISCORD_TOKEN)
```

----- Contents of project_files_text.pdf -----

----- Contents of COMMIT_EDITMSG -----

going succesful with the login and getting prices and everything

----- Contents of config -----

```
[core]
 repositoryformatversion = 0
 filemode = false
 bare = false
 logallrefupdates = true
 symlinks = false
 ignorecase = true
[remote "origin"]
 url = https://github.com/oguzky7/DiscordBotProject_CISC699.git
 fetch = +refs/heads/*:refs/remotes/origin/*
[branch "develop"]
 remote = origin
 merge = refs/heads/develop
 vscode-merge-base = origin/develop
[branch "StartOverAgain"]
 remote = origin
 merge = refs/heads/StartOverAgain
 vscode-merge-base = origin/develop
[branch "StartOver"]
 remote = origin
 merge = refs/heads/StartOver
 vscode-merge-base = origin/develop
```

----- Contents of description -----

Unnamed repository; edit this file 'description' to name the repository.

----- Contents of FETCH_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69 branch 'develop' of https://github.com/oguzky7/DiscordBotProject_CISC699

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b not-for-merge branch 'StartOver' of https://github.com/oguzky7/DiscordBotProject_CISC699

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 not-for-merge branch 'StartOverAgain' of https://github.com/oguzky7/DiscordBotProject_CISC699

e6f9da804a74f224a4f50336480f1896b3142fde not-for-merge branch 'main' of https://github.com/oguzky7/DiscordBotProject_CISC699

----- Contents of HEAD -----

ref: refs/heads/StartOverAgain

----- Contents of index -----

----- Contents of ORIG_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of packed-refs -----

# pack-refs with: peeled fully-peeled sorted

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b refs/remotes/origin/StartOver

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 refs/remotes/origin/StartOverAgain

dad457b1e774179c151ce94c80cca9d1605d2b69 refs/remotes/origin/develop

e6f9da804a74f224a4f50336480f1896b3142fde refs/remotes/origin/main

----- Contents of applypatch-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.  The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:
```

----- Contents of commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message.
```

```
# Called by "git commit" with one argument, the name of the file
# that has the commit message.  The hook should exit with non-zero
# status after issuing an appropriate message if it wants to stop the
# commit.  The hook is allowed to edit the commit message file.
#
# To enable this hook, rename this file to "commit-msg".

# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"

# This example catches duplicate Signed-off-by lines.

test "" = "$(grep '^Signed-off-by: ' "$1" |
  sort | uniq -c | sed -e '/^[  ]*1[  ]/d')" || {
 echo >&2 Duplicate Signed-off-by lines.
 exit 1
}
```

----- Contents of fsmonitor-watchman.sample -----

```
#!/usr/bin/perl

use strict;
```

```
use warnings;
use IPC::Open2;

# An example hook script to integrate Watchman
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 2) and last update token
# formatted as a string and outputs to stdout a new update token and
# all files that have been modified since the update token. Paths must
# be relative to the root of the working tree and separated by a single NUL.
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $last_update_token) = @ARGV;

# Uncomment for debugging
# print STDERR "$0 $version $last_update_token\n";

# Check the hook interface version
if ($version ne 2) {
 die "Unsupported query-fsmonitor hook version '$version'.\n" .
     "Falling back to scanning...\n";
}

my $git_work_tree = get_working_dir();
```

```
my $retry = 1;

my $json_pkg;
eval {
 require JSON::XS;
 $json_pkg = "JSON::XS";
 1;
} or do {
 require JSON::PP;
 $json_pkg = "JSON::PP";
};

launch_watchman();

sub launch_watchman {
 my $o = watchman_query();
 if (is_work_tree_watched($o)) {
  output_result($o->{clock}, @{$o->{files}});
 }
}

sub output_result {
 my ($clockid, @files) = @_;

 # Uncomment for debugging watchman output
 # open (my $fh, ">", ".git/watchman-output.out");
```

```
 # binmode $fh, ":utf8";
 # print $fh "$clockid\n@files\n";
 # close $fh;

 binmode STDOUT, ":utf8";
 print $clockid;
 print "\0";
 local $, = "\0";
 print @files;
}

sub watchman_clock {
 my $response = qx/watchman clock "$git_work_tree"/;
 die "Failed to get clock id on '$git_work_tree'.\n" .
  "Falling back to scanning...\n" if $? != 0;

 return $json_pkg->new->utf8->decode($response);
}

sub watchman_query {
 my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
 or die "open2() failed: $!\n" .
 "Falling back to scanning...\n";

 # In the query expression below we're asking for names of files that
 # changed since $last_update_token but not from the .git folder.
 #
```

```
# To accomplish this, we're using the "since" generator to use the
# recency index to select candidate nodes and "fields" to limit the
# output to file names only. Then we're using the "expression" term to
# further constrain the results.
my $last_update_line = "";
if (substr($last_update_token, 0, 1) eq "c") {
 $last_update_token = "\"$last_update_token\"";
 $last_update_line = qq[\n"since": $last_update_token,];
}
my $query = <<" END";
 ["query", "$git_work_tree", {$last_update_line
  "fields": ["name"],
  "expression": ["not", ["dirname", ".git"]]
 }]
END

# Uncomment for debugging the watchman query
# open (my $fh, ">", ".git/watchman-query.json");
# print $fh $query;
# close $fh;

print CHLD_IN $query;
close CHLD_IN;
my $response = do {local $/; <CHLD_OUT>};

# Uncomment for debugging the watch response
# open ($fh, ">", ".git/watchman-response.json");
```

```
 # print $fh $response;
 # close $fh;

 die "Watchman: command returned no output.\n" .
 "Falling back to scanning...\n" if $response eq "";
 die "Watchman: command returned invalid output: $response\n" .
 "Falling back to scanning...\n" unless $response =~ /^\{/;

 return $json_pkg->new->utf8->decode($response);
}

sub is_work_tree_watched {
 my ($output) = @_;
 my $error = $output->{error};
 if ($retry > 0 and $error and $error =~ m/unable to resolve root .* directory (.*) is not watched/) {
  $retry--;
  my $response = qx/watchman watch "$git_work_tree"/;
  die "Failed to make watchman watch '$git_work_tree'.\n" .
      "Falling back to scanning...\n" if $? != 0;
  $output = $json_pkg->new->utf8->decode($response);
  $error = $output->{error};
  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  # Uncomment for debugging watchman output
  # open (my $fh, ">", ".git/watchman-output.out");
  # close $fh;
```

```
  # Watchman will always return all files on the first query so
  # return the fast "everything is dirty" flag to git and do the
  # Watchman query just to get it over with now so we won't pay
  # the cost in git to look up each individual file.
  my $o = watchman_clock();
  $error = $output->{error};

  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  output_result($o->{clock}, ("/"));
  $last_update_token = $o->{clock};

  eval { launch_watchman() };
  return 0;
 }

 die "Watchman: $error.\n" .
 "Falling back to scanning...\n" if $error;

 return 1;
}

sub get_working_dir {
 my $working_dir;
 if ($^O =~ 'msys' || $^O =~ 'cygwin') {
```

```
  $working_dir = Win32::GetCwd();
  $working_dir =~ tr/\\/\//;
 } else {
  require Cwd;
  $working_dir = Cwd::cwd();
 }

 return $working_dir;
}
```

----- Contents of post-update.sample -----

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

----- Contents of pre-applypatch.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
```

```
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:
```

----- Contents of pre-commit.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
 against=HEAD
else
 # Initial commit: diff against an empty tree object
```

```
 against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
if [ "$allownonascii" != "true" ] &&
 # Note that the use of brackets around a tr range is ok here, (it's
 # even required, for portability to Solaris 10's /usr/bin/tr), since
 # the square bracket bytes happen to fall in the designated range.
 test $(git diff --cached --name-only --diff-filter=A -z $against |
   LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
then
 cat <<\EOF
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.

To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:
```

```
  git config hooks.allownonascii true
EOF
 exit 1
fi

# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --
```

----- Contents of pre-merge-commit.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
        exec "$GIT_DIR/hooks/pre-commit"
:
```

----- Contents of pre-push.sample -----

```
#!/bin/sh

# An example hook script to verify what is about to be pushed.  Called by "git
# push" after it has checked the remote status, but before anything has been
# pushed.  If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
#   <local ref> <local oid> <remote ref> <remote oid>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')

while read local_ref local_oid remote_ref remote_oid
```

```
do
 if test "$local_oid" = "$zero"
 then
  # Handle delete
  :
 else
  if test "$remote_oid" = "$zero"
  then
   # New branch, examine all commits
   range="$local_oid"
  else
   # Update to existing branch, examine new commits
   range="$remote_oid..$local_oid"
  fi

  # Check for WIP commit
  commit=$(git rev-list -n 1 --grep '^WIP' "$range")
  if test -n "$commit"
  then
   echo >&2 "Found WIP commit in $local_ref, not pushing"
   exit 1
  fi
 fi
done

exit 0
```

----- Contents of pre-rebase.sample -----

```
#!/bin/sh
#
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.

publish=next
basebranch="$1"
if test "$#" = 2
then
 topic="refs/heads/$2"
else
 topic=`git symbolic-ref HEAD` ||
 exit 0 ;# we do not interrupt rebasing detached HEAD
```

```
fi

case "$topic" in
refs/heads/??/*)
 ;;
*)
 exit 0 ;# we do not interrupt others.
 ;;
esac

# Now we are dealing with a topic branch being rebased
# on top of master.  Is it OK to rebase it?

# Does the topic really exist?
git show-ref -q "$topic" || {
 echo >&2 "No such branch $topic"
 exit 1
}

# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
 echo >&2 "$topic is fully merged to master; better remove it."
 exit 1 ;# we could allow it, but there is no point.
fi
```

```
# Is topic ever merged to next?  If so you should not be rebasing it.
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
only_next_2=`git rev-list ^master           ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
then
 not_in_topic=`git rev-list "^$topic" master`
 if test -z "$not_in_topic"
 then
  echo >&2 "$topic is already up to date with master"
  exit 1 ;# we could allow it, but there is no point.
 else
  exit 0
 fi
else
 not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
 /usr/bin/perl -e '
  my $topic = $ARGV[0];
  my $msg = "* $topic has commits already merged to public branch:\n";
  my (%not_in_next) = map {
   /^([0-9a-f]+) /;
   ($1 => 1);
  } split(/\n/, $ARGV[1]);
  for my $elem (map {
    /^([0-9a-f]+) (.*)$/;
    [$1 => $2];
   } split(/\n/, $ARGV[2])) {
   if (!exists $not_in_next{$elem->[0]}) {
```

```
    if ($msg) {
     print STDERR $msg;
     undef $msg;
    }
    print STDERR " $elem->[1]\n";
   }
  }
 ' "$topic" "$not_in_next" "$not_in_master"
 exit 1
fi

<<\DOC_END

This sample hook safeguards topic branches that have been
published from being rewound.

The workflow assumed here is:

 * Once a topic branch forks from "master", "master" is never
   merged into it again (either directly or indirectly).

 * Once a topic branch is fully cooked and merged into "master",
   it is deleted.  If you need to build on top of it to correct
   earlier mistakes, a new topic branch is created by forking at
   the tip of the "master".  This is not strictly necessary, but
   it makes it easier to keep your history simple.
```

 * Whenever you need to test or publish your changes to topic
   branches, merge them into "next" branch.

The script, being an example, hardcodes the publish branch name
to be "next", but it is trivial to make it configurable via
$GIT_DIR/config mechanism.

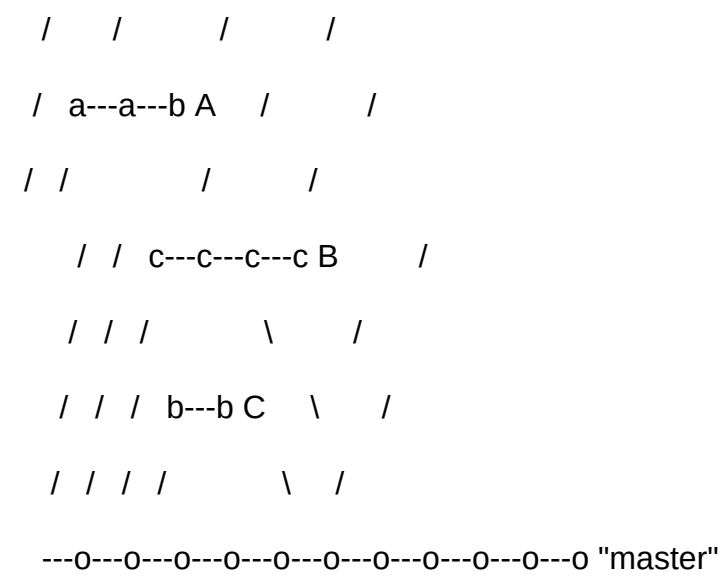
With this workflow, you would want to know:

(1) ... if a topic branch has ever been merged to "next".  Young
    topic branches can have stupid mistakes you would rather
    clean up before publishing, and things that have not been
    merged into other branches can be easily rebased without
    affecting other people.  But once it is published, you would
    not want to rewind it.

(2) ... if a topic branch has been fully merged to "master".
    Then you can delete it.  More importantly, you should not
    build on top of it -- other people may already want to
    change things related to the topic as patches against your
    "master", so if you need further changes, it is better to
    fork the topic (perhaps with the same name) afresh from the
    tip of "master".

Let's look at this example:

     o---o---o---o---o---o---o---o---o---o "next"




A, B and C are topic branches.

* A has one fix since it was merged up to "next".

* B has finished.  It has been fully merged up to "master" and "next",
  and is ready to be deleted.

* C has not merged to "next" at all.

We would want to allow C to be rebased, refuse A, and encourage
B to be deleted.

To compute (1):

```
git rev-list ^master ^topic next
git rev-list ^master        next
```

if these match, topic has not merged in next at all.

To compute (2):

git rev-list master..topic

if this is empty, it is fully merged to "master".

DOC_END

----- Contents of pre-receive.sample -----

```
#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"
then
 i=0
 while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
 do
  eval "value=\$GIT_PUSH_OPTION_$i"
  case "$value" in
```

```
  echoback=*)
   echo "echo from the pre-receive-hook: ${value#*=}" >&2
   ;;
  reject)
   exit 1
  esac
  i=$((i + 1))
 done
fi
```

----- Contents of prepare-commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source.  The hook's purpose is to edit the commit
# message file.  If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
```

```
# into the message, just before the "git status" output.  It is
# commented because it doesn't cope with --amend or with squashed
# commits.
#
# The third example adds a Signed-off-by line to the message, that can
# still be edited.  This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3

/usr/bin/perl -i.bak -ne 'print unless(m/^. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FILE"

# case "$COMMIT_SOURCE,$SHA1" in
#  ,|template,)
#    /usr/bin/perl -i.bak -pe '
#       print "\n" . `git diff --cached --name-status -r`
#   if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
#  *) ;;
# esac

# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
#   /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
```

```
# fi


----- Contents of push-to-checkout.sample -----
#!/bin/sh

# An example hook script to update a checked-out tree on a git push.
#
# This hook is invoked by git-receive-pack(1) when it reacts to git
# push and updates reference(s) in its repository, and when the push
# tries to update the branch that is currently checked out and the
# receive.denyCurrentBranch configuration variable is set to
# updateInstead.
#
# By default, such a push is refused if the working tree and the index
# of the remote repository has any difference from the currently
# checked out commit; when both the working tree and the index match
# the current commit, they are updated to match the newly pushed tip
# of the branch. This hook is to be used to override the default
# behaviour; however the code below reimplements the default behaviour
# as a starting point for convenient modification.
#
# The hook receives the commit with which the tip of the current
# branch is going to be updated:
commit=$1

# It can exit with a non-zero status to refuse the push (when it does
```

```
# so, it must not modify the index or the working tree).
die () {
 echo >&2 "$*"
 exit 1
}

# Or it can make any necessary changes to the working tree and to the
# index to bring them to the desired state when the tip of the current
# branch is updated to the new commit, and exit with a zero status.
#
# For example, the hook can simply run git read-tree -u -m HEAD "$1"
# in order to emulate git fetch that is run in the reverse direction
# with git push, as the two-tree form of git read-tree -u -m is
# essentially the same as git switch or git checkout that switches
# branches while keeping the local changes in the working tree that do
# not interfere with the difference between the branches.

# The below is a more-or-less exact translation to shell of the C code
# for the default behaviour for git's push-to-checkout hook defined in
# the push_to_deploy() function in builtin/receive-pack.c.
#
# Note that the hook will be executed from the repository directory,
# not from the working tree, so if you want to perform operations on
# the working tree, you will have to adapt your code accordingly, e.g.
# by adding "cd .." or using relative paths.

if ! git update-index -q --ignore-submodules --refresh
```

```
then
 die "Up-to-date check failed"
fi

if ! git diff-files --quiet --ignore-submodules --
then
 die "Working directory has unstaged changes"
fi

# This is a rough translation of:
#
#   head_has_history() ? "HEAD" : EMPTY_TREE_SHA1_HEX
if git cat-file -e HEAD 2>/dev/null
then
 head=HEAD
else
 head=$(git hash-object -t tree --stdin </dev/null)
fi

if ! git diff-index --quiet --cached --ignore-submodules $head --
then
 die "Working directory has staged changes"
fi

if ! git read-tree -u -m "$commit"
then
 die "Could not update working tree to new HEAD"
```

```
fi



----- Contents of update.sample -----
#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# ------
# hooks.allowunannotated
#   This boolean sets whether unannotated tags will be allowed into the
#   repository.  By default they won't be.
# hooks.allowdeletetag
#   This boolean sets whether deleting tags will be allowed in the
#   repository.  By default they won't be.
# hooks.allowmodifytag
#   This boolean sets whether a tag may be modified after creation. By default
#   it won't be.
# hooks.allowdeletebranch
#   This boolean sets whether deleting branches will be allowed in the
#   repository.  By default they won't be.
# hooks.denycreatebranch
#   This boolean sets whether remotely creating branches will be denied
```

```
#   in the repository.  By default this is allowed.
#

# --- Command line
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
 echo "Don't run this script from the command line." >&2
 echo " (if you want, you could supply GIT_DIR then run" >&2
 echo "  $0 <ref> <oldrev> <newrev>)" >&2
 exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
 echo "usage: $0 <ref> <oldrev> <newrev>" >&2
 exit 1
fi

# --- Config
allowunannotated=$(git config --type=bool hooks.allowunannotated)
allowdeletebranch=$(git config --type=bool hooks.allowdeletebranch)
denycreatebranch=$(git config --type=bool hooks.denycreatebranch)
allowdeletetag=$(git config --type=bool hooks.allowdeletetag)
allowmodifytag=$(git config --type=bool hooks.allowmodifytag)
```

```
# check for no description
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
case "$projectdesc" in
"Unnamed repository"* | "")
 echo "*** Project description file hasn't been set" >&2
 exit 1
 ;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')
if [ "$newrev" = "$zero" ]; then
 newrev_type=delete
else
 newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
 refs/tags/*,commit)
  # un-annotated tag
  short_refname=${refname##refs/tags/}
  if [ "$allowunannotated" != "true" ]; then
   echo "*** The un-annotated tag, $short_refname, is not allowed in this repository" >&2
   echo "*** Use 'git tag [ -a | -s ]' for tags you want to propagate." >&2
   exit 1
```

```
 fi
 ;;
refs/tags/*,delete)
 # delete tag
 if [ "$allowdeletetag" != "true" ]; then
  echo "*** Deleting a tag is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/tags/*,tag)
 # annotated tag
 if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
 then
  echo "*** Tag '$refname' already exists." >&2
  echo "*** Modifying a tag is not allowed in this repository." >&2
  exit 1
 fi
 ;;
refs/heads/*,commit)
 # branch
 if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
  echo "*** Creating a branch is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/heads/*,delete)
 # delete branch
```

```
  if [ "$allowdeletebranch" != "true" ]; then
   echo "*** Deleting a branch is not allowed in this repository" >&2
   exit 1
  fi
  ;;
 refs/remotes/*,commit)
  # tracking branch
  ;;
 refs/remotes/*,delete)
  # delete tracking branch
  if [ "$allowdeletebranch" != "true" ]; then
   echo "*** Deleting a tracking branch is not allowed in this repository" >&2
   exit 1
  fi
  ;;
 *)
  # Anything else (is there anything else?)
  echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
  exit 1
  ;;
esac

# --- Finished
exit 0
```

----- Contents of exclude -----

# git ls-files --others --exclude-from=.git/info/exclude

# Lines that start with '#' are comments.

# For a project mostly in C, the following would be a good set of

# exclude patterns (uncomment them if you want to use them):

# *.[oa]

# *~

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

dad457b1e774179c151ce94c80cca9d1605d2b69 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 checkout: moving from develop to StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 checkout: moving from StartOverAgain to

StartOver

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654644 -0500 checkout: moving from StartOver to StartOverAgain

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08 oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42 oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of develop -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447

-0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOver -----

0000000000000000000000000000000000000000 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 branch: Created from refs/remotes/origin/StartOver

----- Contents of StartOverAgain -----

0000000000000000000000000000000000000000 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 branch: Created from refs/remotes/origin/StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in

discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed

oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOverAgain -----

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d

oguzky7 <oguzky7@gmail.com> 1725649275 -0500 update by push

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681

oguzky7 <oguzky7@gmail.com> 1725652043 -0500 update by push

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9

oguzky7 <oguzky7@gmail.com> 1725653629 -0500 update by push

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6

oguzky7 <oguzky7@gmail.com> 1725654567 -0500 update by push

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483

oguzky7 <oguzky7@gmail.com> 1725658344 -0500 update by push

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec

oguzky7 <oguzky7@gmail.com> 1725664583 -0500 update by push

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed

oguzky7 <oguzky7@gmail.com> 1725666447 -0500 update by push

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723082 -0500 update by push

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725779 -0500 update by push

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731942 -0500 update by push

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732309 -0500 update by push

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740312 -0500 update by push

----- Contents of 6d281c79b57abc36be3229d4bbab976d1b6bf3 -----

----- Contents of 634f7e652d3fb3b816adc257a0161cfaa59896 -----

----- Contents of d30f9eeb85ec53d1d3fe51515a9471ee79af75 -----

----- Contents of da87394b264ec22cc557849312040a03c1063e -----

----- Contents of acbb56d15479249f6c46de8e3acb0c8b20cb93 -----

----- Contents of cd8b5e55c2ae7f362dc0c8436c4b89a275da90 -----

----- Contents of 56de35ec5b020a6ed5805e2520c51bac37d017 -----

----- Contents of 80ceb22ccad5017ec7c7873c96e5203198cb7b -----

----- Contents of ae88962bd7ef5ee6892e9dbd1761ec6d56b7d1 -----

----- Contents of 0d5a7309d480be697278ca6d58e46a30f02681 -----

----- Contents of 2f7fda7f6f8ecf990caa0731e5be154b2adc39 -----

----- Contents of 670af298f99b15bd9dd09da31d72f0f1d38cdd -----

----- Contents of 707c217ce7f6abc9094c93c766f29b6eab99e4 -----

----- Contents of 73684a5a9f9b5ebf1227f6801d4e478fe16d4a -----

----- Contents of d21b42548625adec2ddceedb034d770341a8c8 -----

----- Contents of d19a6ce55b82d3b656da482d1c8e6a0b54b75b -----

----- Contents of 12668a4da67180c42fa2fab9cd1de9a032b227 -----

----- Contents of 8387cc096c39bd2853df1c025f6cda162264e7 -----

----- Contents of 7bedd012010871eb4fb46ee52ec2ec4bd1c14f -----

----- Contents of f7ffd8161a083f528f9374f3a9a394498db932 -----

----- Contents of 5f314f257340ec9bfbb711c9d13737f9423489 -----

----- Contents of 691ffa09c5106ba04323f77fd727a278646d28 -----

----- Contents of b53adee5cf0e0bccf2f8ad73d337a79dc020d4 -----

----- Contents of d862acaeafd794c8d81a637c85da6be6878e7d -----

----- Contents of 52beb42325a46c9fbd5a5e3e3fa9f54432bd3b -----

----- Contents of 5795b3882003e2fb6801781de44d8bf296680c -----

----- Contents of 740f61a43018fb8b31d987eb2c5e6c674ce9fc -----

----- Contents of 29623a9d92dc8f9e81e3e27a2deade8f63d70f -----

----- Contents of 7fa5fe2e9d0be5e56ceb6c07452184fccca8ad -----

----- Contents of e571a06a385f9280d5a100f60a2d1ffba1f221 -----

----- Contents of 08c8132fd411e30656c37d89b16b6dcd8b22f4 -----

----- Contents of 0c06257cbe7ea26c5c9ce349da25220ae8d9e2 -----

----- Contents of 0dfcd25d3efecd8aac12bcc1a41befb30deaf7 -----

----- Contents of 4dd067b87b28cc9a8cca80caac89fc53036596 -----

----- Contents of c2fabce9b84c9ba79bc2e79a1ac96cc34a1003 -----

----- Contents of 7bbbe8178759121dbcda100e7c83a2c1d493e9 -----

----- Contents of 6c22f5474f34c8623c296e4b6214d003eb3b08 -----

----- Contents of 56b260e1d2b7fd86f354e6fff1b011231cd844 -----

----- Contents of c1291d130ce613ec78e688bf51a576d5efc5c3 -----

----- Contents of e689cbc21247f718c17658183158af910d05d8 -----

----- Contents of 2e280b0131be8c9e98b7222577acd8341d8fa5 -----

----- Contents of e6f8d57f1f260304e67a6b0a0286689893b8e8 -----

----- Contents of c0afdc9c0e97448eb03cd4b9dd9ce145fd8d0b -----

----- Contents of 57837b989b94d629bfc5a49ba5fc0e8261e483 -----

----- Contents of c230c6518a1ad0586efaa11fd03274ae402a27 -----

----- Contents of 2265060579174531072 2ebcce1ece57872317e -----

----- Contents of 16855907d1ddccf87830cc3220e0815d2043d6 -----

----- Contents of 7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed -----

----- Contents of 8be98f50f5ab158fff2c10db14f835a386fc99 -----

----- Contents of 5847fd8df25f2a9b603ed33703a21e91ba67c0 -----

----- Contents of 79b9a56dfb3a07e75198f466fceca1c3d8c1de -----

----- Contents of 7c13cbb1ada6eeb37aa304a9f8afb8744a47dc -----

----- Contents of 124da30b98b65e0df305d0900c2d1282b7d270 -----

----- Contents of 1cd1eadaa24a9a3f632873c244100ee7025794 -----

----- Contents of 36f14349b3a4410888ac1b43070bba46de9ede -----

----- Contents of 6ebe87b1695efcfc11786fb574e0f04bca1d18 -----

----- Contents of ddf2dd50c642602c7eba5268be402b32f66989 -----

----- Contents of 36742a002f99203e8212f1fba1a445a1bb03f6 -----

----- Contents of 623cd7312f1ff9c8abe8832fd36f20e6b84bcd -----

----- Contents of 7461d191d8a7a5de9217e88c6c140c95dd9bca -----

----- Contents of e326319a500dd5126f6173fdc28861abf9de10 -----

----- Contents of 167ddd3cb16efd9af1c2cd4539382ccd57e2fd -----

----- Contents of a951080c6be53c94cb18b588e50d8628833d3e -----

----- Contents of f3b3a6001ee55883e947e28ac1e2c7b08387bd -----

----- Contents of 2778616f503b20c4ebcb0ddd0acfaf1de01e39 -----

----- Contents of 2f14edff81d9dd2f05f2d2f8c7cd365d932bbc -----

----- Contents of 20fcb37bf59941b57c5a694a939d13d15b8900 -----

----- Contents of 98a6460ae86feb9f8049f407d2f6887424e6dc -----

----- Contents of 374847dfb20aff248fdcd59f87bdea40697b1d -----

----- Contents of f89b3afaa02365ae60cb072cf76abf1cf77ae9 -----

----- Contents of 81d6b895e5ee4ada44b133c81c73d79b001b04 -----

----- Contents of e8393441a7d2050cce36faa6564fc5cd9763ec -----

----- Contents of e434a795adc8737398a481f4c813fc380cebf3 -----

----- Contents of 7bdce5089b6b1b983d2a088b0b1a2bb2ebc56c -----

----- Contents of 790b96b9de32b415393a1c2aed46ffb49e6ae2 -----

----- Contents of e495a99648be872776623d90e76250b9cb227e -----

----- Contents of 92fdb8f360a7f817c203304d90f8e42bc44ba9 -----

----- Contents of 922e794ba24b02220cd9bb8547894d915a6e6c -----

----- Contents of bbfb724cd9bc1af776578e20073f1ec227dcfd -----

----- Contents of d010362b749bd8b0940c42084cc7192ed7d944 -----

----- Contents of 2e28edc4f64174222c015a31044054e9302f13 -----

----- Contents of 31e9454a465493fb8c05d28c52697757037d5e -----

----- Contents of c2bb829e59ca69e85259035c1e808b2c17b5b1 -----

----- Contents of 7a34d8216bffdd74b60e3e759e9ade11991cd4 -----

----- Contents of d705b2782872635fbf52906a037a130f467c0f -----

----- Contents of e8bfbb3ec7cce8b3945b1cd33efb7449fb8203 -----

----- Contents of 4c76280fb6b9eefe4f99a0dc7cc4719b85c93d -----

----- Contents of 7feec2f1765e89d736a9359a5c5e551b5f9c6d -----

----- Contents of d386fadadd939cc18d101127a21d961b148a2f -----

----- Contents of 23564225a6a0b77d1fd84d4c3b7e0800177c88 -----

----- Contents of a49e2dcd1c2d8824be05f5a7fa8d279c1c8be3 -----

----- Contents of 7422ef6e171a6dff0d231427b2d364d2dd5bb6 -----

----- Contents of b4cc5d5bf0c7aa29eb4ac829400843fcd51562 -----

----- Contents of 55b982b96b526b4e3db1ee1fe9375b73bba374 -----

----- Contents of ce2369a09426c22593285c67f91cfdcb5011ce -----

----- Contents of 4aa1ac0374fdefe7baf0c2ab6f1c65140b3190 -----

----- Contents of 5e2674b3cb3a1bed397efd15b859641dae7757 -----

----- Contents of 3184272755e64a6944c9869379c6f57ad9ba3b -----

----- Contents of f2c49d67157e02272e6521a0acb70994aff7a7 -----

----- Contents of d105aab298c832e87d7cd64886775dabf92683 -----

----- Contents of 36b1cc2ef835a0eb0f8dab3f76b700dee1ca9c -----

----- Contents of bb301025e8f69ddd490791a3c4b9fe54340715 -----

----- Contents of 3f46aa910385ec1c113d29d13a51b5933246f0 -----

----- Contents of 569d2ce7f8e2dbd43c62f90d8c1a4776fe3657 -----

----- Contents of 35ac6cdff50420307d1f6a29ae6fbce704431b -----

----- Contents of 49235dfb1543a8846b9a4317a4f3b2330af7f5 -----

----- Contents of e4df3e7aa9aa939067d082dc81129d8797e8b9 -----

----- Contents of 15d9428c470f960b9f68228b7e6e037d6fffdb -----

----- Contents of 55c1a5a11c29d9b7088fb5fae913582b80072e -----

----- Contents of 5963004ac6a7c723c76e303c6a933a7b60294e -----

----- Contents of 45a539b3b325f9d4078b26f7788fa7e4e20f4c -----

----- Contents of 6f2e8f648efc155b28e6a77f84c6d4a144f6f6 -----

----- Contents of aaf5c01d5dfff33e76d47f53e18763ae113efd -----

----- Contents of 870dde41518a8ea43fcc6a8b0c64fbf62f65f0 -----

----- Contents of 31e65729c1b72967bcf8d1bba9f4de309309ac -----

----- Contents of ffdef56d17933b2cfc82c5ad7b1fbe98641040 -----

----- Contents of 42bbb4484463cf439fba484bd82cda68230d00 -----

----- Contents of 93d2668edf7fa19ebd76cb7fd46619b412edb8 -----

----- Contents of 65f80d0fa9995fb5cd1505370703edba752087 -----

----- Contents of d12146fcaf19d62254656fde8a206024bd645e -----

----- Contents of fef29125046ebacc0b5bd63cdd89833f67d207 -----

----- Contents of 35fa7f7ad4a8914f3ff46466b98e724a942aad -----

----- Contents of 6cad680be43e263296434712ccfa7cb8683608 -----

----- Contents of 9c8e7e45537aec2443783b44ffe32a2a1d4038 -----

----- Contents of df0c2c3f663c5e0fdb705c6386fbac3c585e79 -----

----- Contents of df6da44831a6c19badb226138612e404e72fde -----

----- Contents of 95216415e1e4b423dbb14fc246d1c13f9af72c -----

----- Contents of 6048147042c058cc22c5089e6d17e8e82247e1 -----

----- Contents of cb233a35c89247d61009b08fdfbba29c952bb8 -----

----- Contents of a97486baf65dd5f97c2e2029485456bcd53875 -----

----- Contents of db4e4cb78b56559a72aca093e43364ab09e531 -----

----- Contents of dd929f339e858c99f485675a0cfd352b143313 -----

----- Contents of 134b6a845b8ff447308d5cfa26acf1abecff81 -----

----- Contents of a8744a97a8497b5a971774408b4f13efb2d36b -----

----- Contents of ce779286a458df17bc46ac6893fc66f279c068 -----

----- Contents of 1bfbc86a74447106a2a964f01541b0dc34c2b4 -----

----- Contents of 2b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb -----

----- Contents of 2f8994666a261dade3f5a5c6edb13ce71f5f05 -----

----- Contents of 4fc156385de559702c75ad60d5a7a344792805 -----

----- Contents of a3700d5e8f8bb3918aa071f60ea5ded21f77d5 -----

----- Contents of bc23c3f2b221779c8bc2c90ffe3708d2171c48 -----

----- Contents of 04f9129475b746303fc616658fc3416237821d -----

----- Contents of d073d6abfbf754fd97b9329f87cec83207ee08 -----

----- Contents of e3c2abaabea3cafb640d47d6f9dc3030186fd2 -----

----- Contents of 547b1e262f315605e9d0b4d65983e72b19449c -----

----- Contents of f3d9725de4b2121766be9d9099dd28a04183e2 -----

----- Contents of 83d1857455ca55b2a4d164f109b372f65b04d4 -----

----- Contents of dcc0a73865688f0a6a2f34ce14f06bb20b5411 -----

----- Contents of 6062c4cd437cf0f7d9498f328b3cc7c39a7aab -----

----- Contents of 00b5e53f01be9fb9c3c8d49599d76948afc966 -----

----- Contents of 871be66b6a4a8f40fd9963b283d8369781dcc2 -----

----- Contents of a99ac7ab128d7cb26d358957b6a913cb17cdce -----

----- Contents of 37d15e04101d6d5747aa61b581a89ae2847b4d -----

----- Contents of 9c78053718a20d8cdf5995cb89f455dbd6becc -----

----- Contents of bd7d94612b16f8217360592a582f490e9d348c -----

----- Contents of 2287757c4e2c0b3c1e391536abb9d9a7b606a0 -----

----- Contents of 89cb5b54aa1d718008b7cd8d4ddc73fbff1c4d -----

----- Contents of 251901266fac8881c65424cf4c60ec63098f04 -----

----- Contents of 6c9ae2c2205d4c1d6823bb667316f36832129d -----

----- Contents of 93a5648f3b5744e544ade4c54a9b2558de33df -----

----- Contents of cf8ddc0f32beaf6deb845b818a237d4e2b12bc -----

----- Contents of 5bcb586c22d55e4d9b45010159cfbe1eb97b3e -----

----- Contents of 11500072f70a22804f86198228a9bf6f4b1723 -----

----- Contents of 212707bdf961cf6127136e77ce1663ca7fa0ac -----

----- Contents of 5a2099c5e29e57c313a0d69bf170ddc1dc7773 -----

----- Contents of 5ee0932ba1d58cdcced9dbc58b5061e0cdc525 -----

----- Contents of 5fc4dc3b398fe65a45236bb26508673b60ab0a -----

----- Contents of 3b2af53524bb1eadba3659d4a3e8e3e8cef960 -----

----- Contents of 5374ec1094c5935714866aa2a07d319cabb50d -----

----- Contents of c45b719c2f6ef47b5a79a6ddeb192ec3c43c9c -----

----- Contents of 3e220c1a0abd5c31bf03f3488ee49da7ba5791 -----

----- Contents of cd6e47d89f170e14c1b089b027c2bfa32a0514 -----

----- Contents of 1019e219967d1a617e593c49eab9dbbc89302d -----

----- Contents of 1b4632f741d513f68f5c9f2147e564f8e6f44e -----

----- Contents of 1b54931841d32971dbfd232f771191f49cea62 -----

----- Contents of 4917f7466bac6768c4b2c970163f10a3da883c -----

----- Contents of d08c9e2e5e8725a91bab9594774b6acda8c235 -----

----- Contents of 83cb74283680914278ac74988973d92220e580 -----

----- Contents of 2db1c7d8b292deaabb4f080f8e287e18c3fc38 -----

----- Contents of 3a8d669dcddfa7a3373422dc3c136dd1f70025 -----

----- Contents of 07750681d74929e692538a74117db1b1779b97 -----

----- Contents of 238ce60d725df8031818872526f17bdc42af50 -----

----- Contents of 7c546782379dc48fa0d181dcffc313b3822924 -----

----- Contents of 0f48a503d03a35ed438b638a1deb8fac419f42 -----

----- Contents of 2104def8712b4ee3d0cdd2a157cf0221c9b77b -----

----- Contents of 04dbf40d154e56070b1344ba8c3cfc5f564746 -----

----- Contents of 6b2e6e73058c710b05b2893bcf4a7e98b7bbad -----

----- Contents of b55569ac58142006b2bd8a09e4e761d57805c1 -----

----- Contents of 25470382adde44f12e3109bd5934c2eecdc918 -----

----- Contents of 4685b2534290d4b5f0cf6ca0d12fabf5d79e8b -----

----- Contents of 751a6ec93ec6beea13aff04bba59d57bf0ea88 -----

----- Contents of 580bd42a3222857aea5885bed9215a1d05b7c6 -----

----- Contents of 964410b7eebddb7e50578e6ccb13c7e8ab7f27 -----

----- Contents of 4c944d46b01fc2734fe68f52ebfde0cb7998e5 -----

----- Contents of 12fb46a5acc7b60a63fc627bdf6f2791543d58 -----

----- Contents of 3237d5a3531bff37efac2e869560a125ece585 -----

----- Contents of 5eea14070cdff5e8b36d391984776aa89aab44 -----

----- Contents of 7ae1b68f35a89244d46925d6cfebd4b74e6c32 -----

----- Contents of f6136b4665613b61ce58bf20a3ceb85d188776 -----

----- Contents of 83d94a59451c4c9a239ac1ae95b06ecb5587d1 -----

----- Contents of 73fd1a7d3b7afbd81ec494aac0cf63725c97f4 -----

----- Contents of f2528590e95cab41dc7026b769871a74e85090 -----

----- Contents of 10cd67f2809bd3968b90699d6e8e7bc7ce438e -----

----- Contents of b96482d8c2ef82f77977c56260947c27516f67 -----

----- Contents of 4b722e77016ace745c5592948cbe8e0a3cb033 -----

----- Contents of ad546333c05b7217d396c913e59ca48241fcc9 -----

----- Contents of 1fd6e881aef18349e7d110677b3472dcacf2a4 -----

----- Contents of 9ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 -----

----- Contents of 41098bb377a0dfca7f862a6009ece6f6d30e4a -----

----- Contents of 78506479aeaf02315ca23e1ce3992b0af824be -----

----- Contents of bbcb0d6a2e1593e7f104095475e4d7d151d091 -----

----- Contents of 9de29bb2d1d6434b8b29ae775ad8c2e48c5391 -----

----- Contents of 09778dc77ac090a0d932499a2232daf1c94abf -----

----- Contents of 2b50c836c96adfa7f3259d99b4b4eb9f2d00e3 -----

----- Contents of 3920a9f5f3d14ccd1ce1c75d288a4e47498d64 -----

----- Contents of b48fdfb32f07250c2d917eef0d362fb6acb61a -----

----- Contents of c168dfbd74cce6f87298b1bcf0a4760533422a -----

----- Contents of 6c3d41c4698dc82ac757363e6998cba6f17c55 -----

----- Contents of b96d1f73ddca7d36a459c4fa161229ab6c68bb -----

----- Contents of db22b05f99c92e086de39ff9302bc5bd942eec -----

----- Contents of 68ca09597ba99ba5a13c4fe0f4e20ad40ea4e1 -----

----- Contents of 17e0509fe70c56442d04a203f7ccc565de0921 -----

----- Contents of 1a629ef9b315d397c4f1bf56b2bbbfc143ff49 -----

----- Contents of 872a50ce0cb03d65925b2a183db893a95ac571 -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.idx -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.pack -----

----- Contents of develop -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of StartOver -----

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of HEAD -----

ref: refs/remotes/origin/develop

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of AccountBoundary.py -----

from discord.ext import commands

```
from control.AccountControl import AccountControl

class AccountBoundary(commands.Cog):

    def __init__(self, bot):
        self.bot = bot
        self.account_control = AccountControl()


    @commands.command(name='fetch_accounts')
    async def fetch_accounts(self, ctx):
        """Fetch and display all accounts."""
        accounts = self.account_control.fetch_accounts()

        # Send each account or the no accounts message to Discord
        for account in accounts:
            await ctx.send(account)



    @commands.command(name="add_account")
    async def add_account(self, ctx, username: str, password: str):
        """Add a new user account to the database."""
        result = self.account_control.add_account(username, password)
        if result:
            await ctx.send(f"Account for {username} added successfully.")
        else:
```

```
            await ctx.send(f"Failed to add account for {username}.")



    @commands.command(name="delete_account")
    async def delete_account(self, ctx, user_id: int):
        """Delete a user account from the database."""
        result = self.account_control.delete_account(user_id)
        if result:
            await ctx.send(f"Account with ID {user_id} deleted successfully.")
        else:
            await ctx.send(f"Failed to delete account with ID {user_id}.")
```

----- Contents of AvailabilityBoundary.py -----

----- Contents of BotBoundary.py -----

```
from discord.ext import commands
from control.ChatControl import ChatControl
from Config import Config

class BotBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.chat_control = ChatControl()
```

```
@commands.Cog.listener()
async def on_ready(self):
    """Bot startup message when ready."""
    print(f'Logged in as {self.bot.user.name}')
    channel = self.bot.get_channel(Config.CHANNEL_ID)
    if channel:
        await channel.send("Hi, I'm online!")

@commands.Cog.listener()
async def on_message(self, message):
    """Handle non-prefixed messages and command-prefixed messages."""
    if message.author == self.bot.user:
        return

    # Handle non-prefixed messages (like greetings)
    if not message.content.startswith('!'):
        response = self.chat_control.process_non_prefixed_message(message.content)
        await message.channel.send(response)

@commands.Cog.listener()
async def on_command_error(self, ctx, error):
    """Handle unrecognized commands."""
    if isinstance(error, commands.CommandNotFound):
        # Handle unknown command
        response = self.chat_control.handle_unrecognized_command()
        await ctx.send(response)
```

----- Contents of BrowserBoundary.py -----

```
from discord.ext import commands
from control.BrowserControl import BrowserControl


class BrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.browser_control = BrowserControl()


    @commands.command(name='launch_browser')
    async def launch_browser(self, ctx, *args):
        """Command to launch the browser."""
        incognito = "incognito" in args
        response = self.browser_control.launch_browser(ctx.author, incognito)
        await ctx.send(response)
```

----- Contents of CloseBrowserBoundary.py -----

```
from discord.ext import commands
from control.CloseBrowserControl import CloseBrowserControl


class CloseBrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.close_browser_control = CloseBrowserControl()
```

```
    @commands.command(name='close_browser')
    async def close_browser(self, ctx):
        """Command to close the browser."""
        response = self.close_browser_control.close_browser()
        await ctx.send(response)
```

----- Contents of DataExtractionBoundary.py -----

----- Contents of HelpBoundary.py -----

```
from discord.ext import commands
from control.HelpControl import HelpControl

class HelpBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.help_control = HelpControl()

    @commands.command(name='project_help')
    async def project_help(self, ctx):
        """Handles the project_help command."""
        help_message = self.help_control.get_help_message()
        await ctx.send(help_message)
```

----- Contents of LoginBoundary.py -----

```
from discord.ext import commands
from control.LoginControl import LoginControl

class LoginBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.login_control = LoginControl()

    @commands.command(name='login')
    async def login(self, ctx, site: str, *args):
        """Command to log into a website using stored credentials."""
        incognito = "incognito" in args
        retries = next((int(arg) for arg in args if arg.isdigit()), 1)
        response = await self.login_control.login(site, incognito, retries)
        await ctx.send(response)
```

----- Contents of MonitorPriceBoundary.py -----

```
from discord.ext import commands
from control.MonitorPriceControl import MonitorPriceControl

class MonitorPriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.monitor_price_control = MonitorPriceControl()

    @commands.command(name='monitor_price')
```

```
    async def monitor_price(self, ctx, url: str, frequency: int = 1):
        """Command to monitor the price at regular intervals."""
        await self.monitor_price_control.monitor_price(ctx, url, frequency)
```

----- Contents of NavigationBoundary.py -----

```
from discord.ext import commands
from control.NavigationControl import NavigationControl

class NavigationBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.navigation_control = NavigationControl()

    @commands.command(name='navigate_to_website')
    async def navigate_to_website(self, ctx, url: str):
        """Command to navigate to a specified URL."""
        response = self.navigation_control.navigate_to_url(url)
        await ctx.send(response)
```

----- Contents of NotificationBoundary.py -----

----- Contents of PriceBoundary.py -----

```
from discord.ext import commands
from control.PriceControl import PriceControl
```

```
class PriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.price_control = PriceControl()

    @commands.command(name='get_price')
    async def get_price(self, ctx, url: str):
        """Command to get the price from the given URL."""
        response = await self.price_control.get_price(ctx, url)
        await ctx.send(response)
```

----- Contents of StopBoundary.py -----

```
from discord.ext import commands
from control.BotControl import BotControl

class StopBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.bot_control = BotControl(bot)

    @commands.command(name="stop_bot")
    async def stop_bot(self, ctx):
        """Handles the stop command and gracefully shuts down the bot."""
        await ctx.send("Stopping the bot...")
        await self.bot_control.stop_bot()
```

----- Contents of StopMonitoringBoundary.py -----

----- Contents of __init__.py -----

#empty init file

----- Contents of AccountBoundary.cpython-312.pyc -----

----- Contents of BotBoundary.cpython-312.pyc -----

----- Contents of BrowserBoundary.cpython-312.pyc -----

----- Contents of ChatBoundary.cpython-312.pyc -----

----- Contents of CloseBrowserBoundary.cpython-312.pyc -----

----- Contents of HelpBoundary.cpython-312.pyc -----

----- Contents of LoginBoundary.cpython-312.pyc -----

----- Contents of MonitorPriceBoundary.cpython-312.pyc -----

----- Contents of NavigationBoundary.cpython-312.pyc -----

----- Contents of PriceBoundary.cpython-312.pyc -----

----- Contents of StopBoundary.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountControl.py -----

```
from entity.AccountEntity import AccountEntity

class AccountControl:
    def __init__(self):
        self.account_entity = AccountEntity()
```

```
    def add_account(self, username, password, webSite):
        self.account_entity.connect()
        self.account_entity.add_account(username, password, webSite)
        self.account_entity.close()

    def fetch_accounts(self):
        """Fetch all accounts and return them."""
        self.account_entity.connect()
        accounts = self.account_entity.fetch_accounts()

        if accounts:
            account_messages = []
            for account in accounts:
                message = f"ID: {account[0]}, Username: {account[1]}, Password: {account[2]}, Website:
{account[3]}"
                print(message)  # For terminal output
                account_messages.append(message)
            self.account_entity.close()
            return account_messages
        else:
            print("No accounts found.")  # For terminal output
            self.account_entity.close()
            return ["No accounts found."]

    def fetch_account_by_website(self, website):
            """Fetch the username and password where the website matches."""
```

```
            self.account_entity.connect()
            account = self.account_entity.fetch_account_by_website(website)  # Call the entity method
            self.account_entity.close()
            return account


    def delete_account(self, account_id):
        self.account_entity.connect()
        self.account_entity.delete_account(account_id)
        self.account_entity.reset_id_sequence()
        self.account_entity.close()
```

----- Contents of AvailabilityControl.py -----

----- Contents of BotControl.py -----

```
import asyncio

class BotControl:
    def __init__(self, bot):
        self.bot = bot

    async def send_greeting(self):
        """Sends a greeting when the bot comes online."""
        channel = self.bot.get_channel(self.bot.config.CHANNEL_ID)
        if channel:
```

```
            await channel.send("Hi, I'm online! type '!project_help' to see what I can do")


    async def stop_bot(self):
        """Stops the bot gracefully, ensuring all connections are closed."""
        print("Bot is stopping...")


        await self.bot.close()
```

----- Contents of BrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl  # Use AccountControl for consistency


class BrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()  # Use AccountControl to fetch credentials


    def launch_browser(self, user, incognito=False):
        return self.browser_entity.launch_browser(incognito=incognito, user=user)
```

----- Contents of ChatControl.py -----

```
# ChatControl in control/ChatControl.py
class ChatControl:
    def process_non_prefixed_message(self, message):
        """Process non-prefixed messages like 'hi', 'hello'."""
```

```
        if message.lower() in ["hi", "hello"]:
            return "Hello! How can I assist you today? Type !project_help for assistance."
        else:
            return "I didn't recognize that. Type !project_help to see available commands."

    def handle_unrecognized_command(self):
        """Handle unrecognized command from on_command_error."""
        return "I didn't recognize that command. Type !project_help for assistance."
```

----- Contents of CloseBrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity

class CloseBrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def close_browser(self):
        return self.browser_entity.close_browser()
```

----- Contents of DataExtractionControl.py -----

----- Contents of HelpControl.py -----

```
class HelpControl:
    def get_help_message(self):
        """Returns a list of available bot commands."""
```

```
        return (
            "Here are the available commands:\n"
            "!project_help - Get help on available commands.\n"
            "!chat_with_bot - Say hi to the bot.\n"
            "!login_to_website - Log in to a website.\n"
            "!launch_browser - Launch the browser.\n"
            "!close_browser - Close the browser.\n"
            "!navigate_to_website - Navigate to a website.\n"
            "!track_price - Track a product price.\n"
            "!check_price - Check the price of a product.\n"
            "!check_availability - Check the availability of a product.\n"
            "!stop_tracking - Stop tracking a product.\n"
            "!receive_notifications - Receive notifications for price changes.\n"
            "!extract_data - Export data to Excel or HTML.\n"
            "!stop - Stop the bot.\n"
        )
```

----- Contents of LoginControl.py -----

```
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl

class LoginControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()
```

```
    async def login(self, site, incognito=False, retries=1):
        # Fetch credentials using AccountControl
        account = self.account_control.fetch_account_by_website(site)
        if account:
            username, password = account
            return await self.browser_entity.login(site, username, password, incognito, retries)
        else:
            return f"No account found for website {site}"



----- Contents of MonitorPriceControl.py -----
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class MonitorPriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("MonitorPriceControl")

    async def monitor_price(self, ctx, url, frequency=1):
        """Monitor the price at a given interval."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                await ctx.send(f"Monitoring price every {frequency} minute(s).")
                previous_price = None
```

```
            while True:
                current_price = self.price_entity.get_price(url)
                if current_price:
                    if previous_price is None:
                        await ctx.send(f"Starting price monitoring. Current price is: {current_price}")
                    else:
                        if current_price > previous_price:
                                await ctx.send(f"Price went up! Current price: {current_price} (Previous:
{previous_price})")
                        elif current_price < previous_price:
                                await ctx.send(f"Price went down! Current price: {current_price} (Previous:
{previous_price})")
                        else:
                            await ctx.send(f"Price remains the same: {current_price}")
                    previous_price = current_price
                else:
                    await ctx.send("Failed to retrieve the price.")
                await asyncio.sleep(frequency * 60)  # Wait for the next check
            except Exception as e:
                self.logger.error(f"Failed to monitor price for {url}: {e}")
                await ctx.send(f"Failed to monitor price: {e}")
        else:
            await ctx.send("This command can only be used in the designated channel.")
```

----- Contents of NavigationControl.py -----

```
from entity.BrowserEntity import BrowserEntity

class NavigationControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def navigate_to_url(self, url):
        """Navigate to a specific URL."""
        return self.browser_entity.navigate_to_url(url)
```

----- Contents of NotificationControl.py -----

----- Contents of PriceControl.py -----

```
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class PriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("PriceControl")

    async def get_price(self, ctx, url):
        """Fetch the current price from the given URL."""
```

```
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                price = self.price_entity.get_price(url)
                if price:
                    return f"The current price is: {price}"
                else:
                    return "Failed to retrieve the price."
            except Exception as e:
                self.logger.error(f"Failed to get price for {url}: {e}")
                return f"Error getting price: {e}"
        else:
            return "This command can only be used in the designated channel."
```

----- Contents of StopMonitoringControl.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of AccountControl.cpython-312.pyc -----

----- Contents of BotControl.cpython-312.pyc -----

----- Contents of BrowserControl.cpython-312.pyc -----

----- Contents of ChatControl.cpython-312.pyc -----

----- Contents of CloseBrowserControl.cpython-312.pyc -----

----- Contents of HelpControl.cpython-312.pyc -----

----- Contents of LoginControl.cpython-312.pyc -----

----- Contents of MonitorPriceControl.cpython-312.pyc -----

----- Contents of NavigationControl.cpython-312.pyc -----

----- Contents of PriceControl.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountEntity.py -----

```
import psycopg2
from Config import Config


class AccountEntity:
    def __init__(self):
        self.dbname = "postgres"
        self.user = "postgres"
        self.host = "localhost"
        self.port = "5432"
        self.password = Config.DATABASE_PASSWORD


    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
                password=self.password,
                host=self.host,
                port=self.port
            )
            self.cursor = self.connection.cursor()
            print("Database Connection Established.")
```

```
        except Exception as error:
            print(f"Error connecting to the database: {error}")
            self.connection = None
            self.cursor = None

    def add_account(self, username, password, webSite):
        """Insert a new account into the accounts table."""
        try:
            if self.cursor:
                self.cursor.execute("INSERT INTO accounts (username, password, website) VALUES
(%s, %s, %s)", (username, password, webSite))
                self.connection.commit()
                print(f"Account {username} added successfully.")
        except Exception as error:
            print(f"Error inserting account: {error}")

    def fetch_accounts(self):
        """Fetch all accounts from the accounts table."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts;")
                accounts = self.cursor.fetchall()
                return accounts
        except Exception as error:
            print(f"Error fetching accounts: {error}")
            return None
```

```
    def delete_account(self, account_id):
        """Delete an account by ID."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts WHERE id = %s", (account_id,))
                account = self.cursor.fetchone()
                if account:
                    self.cursor.execute("DELETE FROM accounts WHERE id = %s", (account_id,))
                    self.connection.commit()
                    print(f"Account with ID {account_id} deleted successfully.")
                else:
                    print(f"Account with ID {account_id} not found. No deletion performed.")
        except Exception as error:
            print(f"Error deleting account: {error}")



    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        try:
                            self.cursor.execute("SELECT username, password FROM accounts WHERE
LOWER(website) = LOWER(%s)", (website,))
            return self.cursor.fetchone()  # Returns one matching account
        except Exception as error:
            print(f"Error fetching account for website {website}: {error}")
            return None
```

```
    def reset_id_sequence(self):
        """Reset the account ID sequence to the next available value."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT COALESCE(MAX(id), 0) + 1 FROM accounts")
                next_id = self.cursor.fetchone()[0]
                        self.cursor.execute("ALTER SEQUENCE accounts_id_seq RESTART WITH %s",
(next_id,))
                self.connection.commit()
                print(f"ID sequence reset to {next_id}.")
        except Exception as error:
            print(f"Error resetting ID sequence: {error}")

    def close(self):
        """Close the database connection."""
        if self.cursor:
            self.cursor.close()
        if self.connection:
            self.connection.close()
            print("Database Connection closed.")



----- Contents of BrowserEntity.py -----
import asyncio
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
```

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from utils.css_selectors import Selectors  # Import CSS selectors for the website

class BrowserEntity:
    _instance = None  # Singleton instance

    def __new__(cls, *args, **kwargs):
        if cls._instance is None:
            cls._instance = super(BrowserEntity, cls).__new__(cls)
            cls._instance.driver = None  # Initialize driver to None
        return cls._instance

    def launch_browser(self, incognito=False, user=None):
        if self.driver:
            print("Browser is already running. No need to launch a new one.")
            return "Browser is already running."

        try:
            # Special launch options as per your original implementation
            options = webdriver.ChromeOptions()

            # Add options to avoid crashing and improve performance
            options.add_argument("--remote-debugging-port=9222")
            options.add_experimental_option("excludeSwitches", ["enable-automation"])
            options.add_experimental_option('useAutomationExtension', False)
```

```
            options.add_argument("--start-maximized")
            options.add_argument("--disable-notifications")
            options.add_argument("--disable-popup-blocking")
            options.add_argument("--disable-infobars")
            options.add_argument("--disable-extensions")
            options.add_argument("--disable-webgl")
            options.add_argument("--disable-webrtc")
            options.add_argument("--disable-rtc-smoothing")

            if incognito:
                options.add_argument("--incognito")

            self.driver = webdriver.Chrome(service=Service(), options=options)
            success_message = "Chrome browser launched successfully in incognito mode." if incognito
else "Chrome browser launched successfully."
            print(f"Driver initialized: {self.driver}")  # Debug: Print the driver
            return success_message
        except Exception as e:
            error_message = f"Failed to launch browser: {e}"
            print(error_message)
            raise

    def navigate_to_url(self, url):
        if not self.driver:
            print("Driver is not initialized, launching browser first.")  # Debug
            self.launch_browser()
        try:
```

```
            self.driver.get(url)
            return f"Navigated to URL: {url}"
        except Exception as e:
            raise


def close_browser(self):
    print(f"Closing browser. Current driver: {self.driver}")  # Debug: Check the driver status
    if self.driver:
        self.driver.quit()  # Close the browser session
        self.driver = None  # Set to None after closing
        print("Browser closed successfully.")
        return "Browser closed successfully."
    else:
        print("No browser is currently open.")
        return "No browser is currently open."



async def login(self, site, username, password, incognito=False, retries=1):
    # Get the URL and selectors from css_selectors
    url = Selectors.get_selectors_for_url(site)['url']
    for attempt in range(retries):
        try:
            self.navigate_to_url(url)
            await asyncio.sleep(3)


            # Enter the email address
                                        email_field    =    self.driver.find_element(By.CSS_SELECTOR,
```

```
Selectors.get_selectors_for_url(site)['email_field'])
            email_field.click()
            email_field.send_keys(username)
            await asyncio.sleep(3)

            # Enter the password
                                    password_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['password_field'])
            password_field.click()
            password_field.send_keys(password)
            await asyncio.sleep(3)

            # Click the login button
                                    sign_in_button = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['SignIn_button'])
            sign_in_button.click()
            await asyncio.sleep(5)

            # Wait for the homepage to load after login
            WebDriverWait(self.driver, 30).until(
                                            EC.presence_of_element_located((By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['homePage'])))

            return f"Logged in to {url} successfully with username: {username}"
        except Exception as e:
            if attempt < retries - 1:
                await asyncio.sleep(3)
```

```
            else:
                raise e
```

----- Contents of DateEntity.py -----

----- Contents of NotificationEntity.py -----

----- Contents of PriceEntity.py -----

```
import time
from selenium.webdriver.common.by import By
from utils.css_selectors import Selectors
from entity.BrowserEntity import BrowserEntity  # Import the browser interaction logic

class PriceEntity:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def get_price(self, url):
        """Fetch the price from the provided URL using CSS selectors."""
        selectors = Selectors.get_selectors_for_url(url)
        if not selectors:
            raise ValueError(f"No selectors found for URL: {url}")

        # Navigate to the URL using the browser entity
```

```
        self.browser_entity.navigate_to_url(url)
        time.sleep(2)  # Wait for the page to load

        try:
            # Use the CSS selector to find the price on the page
                        price_element = self.browser_entity.driver.find_element(By.CSS_SELECTOR,
selectors['price'])
            price = price_element.text
            print(f"Price found: {price}")
            return price
        except Exception as e:
            print(f"Error finding price: {e}")
            return None
```

----- Contents of PriceHistoryEntity.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of AccountEntity.cpython-312.pyc -----

----- Contents of BrowserEntity.cpython-312.pyc -----

----- Contents of PriceEntity.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of project.txt -----

DiscordBotProject_CISC699 - Project Overview

## Introduction

This project is a Discord bot designed to perform various tasks, including tracking product prices, checking availability, logging into websites, and exporting data.

The bot interacts with users via commands sent through Discord and responds based on the requested use case.

The project follows a clear structure, adhering to software engineering best practices, and separates the logic into Boundary, Control, and Entity objects to manage the flow of data and logic.

Scroll all the way down for project outline

## Objects and Their Roles

### Entity Objects

Entity objects represent the core business data and operations related to those entities. They store data and perform business logic related to that data. They do not interact directly with the user.

ProductEntity: Represents product information such as price and features. It handles product-related data (e.g., retrieving the current price).

DateEntity: Handles date and availability logic for booking or checking availability of services.

AccountEntity: Manages user login credentials for websites like BestBuy or eBay.

TrackingHistoryEntity: Stores and tracks historical data on product prices. Helps to compare past prices with current ones.

BrowserEntity: Manages the state of the browser (e.g., if the browser is running, whether it's in

incognito mode, etc.).

NotificationEntity: Handles user preferences for receiving notifications, such as when prices change or product availability is updated.

Control Objects

Control objects are responsible for handling the logic of each use case. They interact with entity objects to manage data and handle business rules. Control objects execute the steps required to fulfill a use case.

HelpControl: Provides a list of commands available to the user.

ChatControl: Handles basic user interaction, such as greetings and responses to basic phrases like "hi" or "hello."

LoginControl: Manages the process of logging into a website, including retrieving login credentials from the database and passing them to the browser.

BrowserControl: Manages the launch and setup of the browser, including handling incognito mode and configuring the browser.

CloseBrowserControl: Handles the logic for closing the browser when requested by the user.

NavigationControl: Manages the process of navigating to a specific URL in the browser.

ProductTrackingControl: Manages the tracking of a product's price over time, scheduling regular price checks.

ProductControl: Checks the current price of a product and retrieves relevant product data.

AvailabilityControl: Handles checking the availability of a product or service based on user-provided dates.

StopTrackingControl: Stops the tracking process for a product or service.

NotificationControl: Monitors for changes in tracked products and sends notifications when a price or availability change occurs.

DataExtractionControl: Manages the extraction of tracking data, exporting it to Excel or HTML files.

BotControl: Manages the overall lifecycle of the Discord bot, including starting, stopping, and managing the registration of commands.

Boundary Objects

Boundary objects serve as the bridge between the user (or external actor) and the system. They collect data from the user and forward it to the appropriate control object. Boundary objects are responsible for interacting with the actor but not for executing business logic.

HelpBoundary: Collects the user?s help request and forwards it to HelpControl.

ChatBoundary: Receives chat commands from the user and forwards them to ChatControl.

LoginBoundary: Collects login credentials from the user and forwards them to LoginControl.

BrowserBoundary: Receives commands to launch the browser and forwards them to BrowserControl.

CloseBrowserBoundary: Receives the user?s request to close the browser and forwards it to CloseBrowserControl.

NavigationBoundary: Receives URL input from the user and forwards it to NavigationControl.

ProductTrackingBoundary: Collects the user?s request to track a product and forwards it to ProductTrackingControl.

ProductBoundary: Receives the user?s request to check a product price and forwards it to ProductControl.

AvailabilityBoundary: Collects the user?s availability check request and forwards it to AvailabilityControl.

StopTrackingBoundary: Receives the user?s request to stop tracking a product and forwards it to StopTrackingControl.

NotificationBoundary: Collects user preferences for receiving notifications and forwards them to NotificationControl.

DataExtractionBoundary: Collects the user?s request to export data and forwards it to DataExtractionControl.

StopBoundary: Receives the request to stop the bot and forwards it to BotControl.

Capabilities

Here?s what the bot can do:

1. !project_help

Description: Provides a list of available commands the user can issue.

Objects Involved:

Boundary: HelpBoundary

Control: HelpControl

Interaction: HelpBoundary collects the user?s help request and forwards it to HelpControl, which responds with the list of commands.

2. !chat_with_bot

Description: Responds to simple greetings (e.g., "hi", "hello") and provides a welcome message.

Objects Involved:

Boundary: ChatBoundary

Control: ChatControl

Interaction: ChatBoundary collects chat input and forwards it to ChatControl, which sends back a predefined response.

3. !login_to_website

Description: Logs into a website using stored credentials (e.g., BestBuy).

Objects Involved:

Boundary: LoginBoundary

Control: LoginControl, BrowserControl, NavigationControl

Entity: AccountEntity

Interaction:

LoginBoundary collects login credentials and forwards them to LoginControl.

LoginControl works with BrowserControl to launch the browser.

NavigationControl navigates to the website's login page.

AccountEntity retrieves the stored credentials from the database and logs the user in.

4. !launch_browser

Description: Launches the browser, optionally in incognito mode.

Objects Involved:

Boundary: BrowserBoundary

Control: BrowserControl

Entity: BrowserEntity

Interaction: BrowserBoundary collects the user's request to launch the browser and sends it to BrowserControl. BrowserControl uses BrowserEntity to configure and launch the browser.

5. !close_browser

Description: Closes the currently open browser session.

Objects Involved:

Boundary: CloseBrowserBoundary

Control: CloseBrowserControl

Entity: BrowserEntity

Interaction: CloseBrowserBoundary forwards the user?s request to CloseBrowserControl, which then tells BrowserEntity to close the browser session.

6. !navigate_to_website

Description: Navigates to a specific website URL in the browser.

Objects Involved:

Boundary: NavigationBoundary

Control: NavigationControl

Entity: BrowserEntity

Interaction: NavigationBoundary collects the URL input from the user and forwards it to NavigationControl. NavigationControl instructs BrowserEntity to navigate to the specified URL.

7. !track_price

Description: Tracks the price of a product over time and sends notifications if the price changes.

Objects Involved:

Boundary: ProductTrackingBoundary

Control: ProductTrackingControl, ProductControl, NotificationControl

Entity: ProductEntity, TrackingHistoryEntity, NotificationEntity

Interaction:

ProductTrackingBoundary collects the product URL from the user.

ProductTrackingControl initiates price tracking and uses ProductControl to fetch the current price.

The current price is stored in TrackingHistoryEntity.

If there?s a price change, NotificationControl sends an alert via NotificationEntity.

8. !check_price

Description: Manually checks the current price of a product.

Objects Involved:

Boundary: ProductBoundary

Control: ProductControl

Entity: ProductEntity

Interaction: ProductBoundary collects the product information from the user, and ProductControl retrieves the current price using ProductEntity.

9. !check_availability

Description: Checks the availability of a product or service on a specific date.

Objects Involved:

Boundary: AvailabilityBoundary

Control: AvailabilityControl

Entity: DateEntity

Interaction: AvailabilityBoundary collects the date and product/service details. AvailabilityControl checks the availability via DateEntity.

10. !stop_tracking

Description: Stops tracking the price or availability of a product.

Objects Involved:

Boundary: StopTrackingBoundary

Control: StopTrackingControl

Entity: TrackingHistoryEntity

Interaction: StopTrackingBoundary collects the stop request from the user. StopTrackingControl stops the tracking and updates TrackingHistoryEntity.

11. !receive_notifications

Description: Sends notifications when there?s a change in price or availability for tracked products/services.

Objects Involved:

Boundary: NotificationBoundary

Control: NotificationControl

Entity: NotificationEntity, TrackingHistoryEntity

Interaction: NotificationBoundary collects the user?s preferences for receiving notifications. NotificationControl monitors for changes and uses NotificationEntity to send alerts when changes occur.

12. !extract_data

Description: Extracts the tracked product data and exports it to Excel or HTML format.

Objects Involved:

Boundary: DataExtractionBoundary

Control: DataExtractionControl

Entity: TrackingHistoryEntity

Utilities: ExcelUtils, HTMLUtils

Interaction: DataExtractionBoundary collects the user?s request for data extraction. DataExtractionControl retrieves data from TrackingHistoryEntity and uses ExcelUtils or HTMLUtils to export the data to the desired format.

13. !stop

Description: Stops the Discord bot from running.

Objects Involved:

Boundary: StopBoundary

Control: BotControl

Interaction: StopBoundary collects the stop command from the user and forwards it to BotControl, which gracefully stops the bot.

DiscordBotProject_CISC699/

?

??? boundary/

?   ??? AccountBoundary.py

?   ??? HelpBoundary.py

?   ??? ChatBoundary.py

?   ??? LoginBoundary.py

?   ??? BrowserBoundary.py

?   ??? CloseBrowserBoundary.py

?   ??? NavigationBoundary.py

?   ??? ProductTrackingBoundary.py

?   ??? ProductBoundary.py

?   ??? AvailabilityBoundary.py

?   ??? StopTrackingBoundary.py

?   ??? NotificationBoundary.py

?   ??? DataExtractionBoundary.py

?   ??? StopBoundary.py

?

??? control/

?   ??? AccountControl.py

?   ??? HelpControl.py

?   ??? ChatControl.py

?   ??? LoginControl.py

?   ??? BrowserControl.py

?   ??? CloseBrowserControl.py

?   ??? NavigationControl.py

?   ??? ProductTrackingControl.py

?   ??? ProductControl.py

?   ??? AvailabilityControl.py

?   ??? StopTrackingControl.py

?   ??? NotificationControl.py

?   ??? DataExtractionControl.py

?   ??? BotControl.py

?

??? entity/

?   ??? ProductEntity.py

?   ??? DateEntity.py

?   ??? AccountEntity.py

?   ??? TrackingHistoryEntity.py

?   ??? BrowserEntity.py

?   ??? NotificationEntity.py

?

??? utils/

?   ??? ExcelUtils.py

```
?   ??? HTMLUtils.py
?   ??? DiscordUtils.py
?
??? test/
?   ??? test_addAccount.py
?   ??? test_deleteAccount.py
?   ??? test_fetchAccounts.py
?   ??? test_excel_creation.py
?   ??? test_html_creation.py
?
??? Config.py
??? main.py
??? project.txt
```

----- Contents of project_structure.py -----

```
import os

def list_files_and_folders(directory, output_file):
    with open(output_file, 'w') as f:
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            f.write(f"Directory: {root}\n")
            for dir_name in dirs:
                f.write(f"  Folder: {dir_name}\n")
```

```
            for file_name in files:
                f.write(f"  File: {file_name}\n")

# Update the directory path to your project folder
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699"
output_file = os.path.join(project_directory, "project_structure.txt")

# Call the function to list files and save output to .txt
list_files_and_folders(project_directory, output_file)

print(f"File structure saved to {output_file}")
```

----- Contents of project_structure.txt -----

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699

  Folder: boundary

  Folder: control

  Folder: entity

  Folder: test

  Folder: utils

  File: Config.py

  File: main.py

  File: project.txt

  File: project_structure.txt

  File: temporary.py

File: Tests_URLs.txt

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\boundary

File: AccountBoundary.py

File: AvailabilityBoundary.py

File: BotBoundary.py

File: BrowserBoundary.py

File: CloseBrowserBoundary.py

File: DataExtractionBoundary.py

File: HelpBoundary.py

File: LoginBoundary.py

File: NavigationBoundary.py

File: NotificationBoundary.py

File: ProductBoundary.py

File: ProductTrackingBoundary.py

File: StopBoundary.py

File: StopTrackingBoundary.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\control

File: AccountControl.py

File: AvailabilityControl.py

File: BotControl.py

File: BrowserControl.py

File: ChatControl.py

File: CloseBrowserControl.py

File: DataExtractionControl.py

File: HelpControl.py

File: LoginControl.py

File: NavigationControl.py

File: NotificationControl.py

File: ProductControl.py

File: ProductTrackingControl.py

File: StopTrackingControl.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\entity

File: AccountEntity.py

File: BrowserEntity.py

File: DateEntity.py

File: NotificationEntity.py

File: ProductEntity.py

File: TrackingHistoryEntity.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\test

File: test_addAccount.py

File: test_deleteAccount.py

File: test_excel_creation.py

File: test_fetchAccounts.py

File: test_html_creation.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\utils

File: css_selectors.py

File: DiscordUtils.py

File: ExcelUtils.py

File: HTMLUtils.py

----- Contents of temporary.py -----

```
import os


def get_all_files_text(directory, output_file):
    with open(output_file, 'w', encoding='utf-8') as outfile:
        # Traverse the directory
        for foldername, subfolders, filenames in os.walk(directory):
            for filename in filenames:
                file_path = os.path.join(foldername, filename)
                try:
                    # Open and read each file's content
                    with open(file_path, 'r', encoding='utf-8') as infile:
                        outfile.write(f"----- Contents of {filename} -----\n")
                        outfile.write(infile.read())
                        outfile.write("\n\n")
                except Exception as e:
                    print(f"Error reading {file_path}: {e}")


if __name__ == "__main__":
            directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699"
```

```
    output_file = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\all_files_content.txt"
    get_all_files_text(directory, output_file)
```

----- Contents of Tests_URLs.txt -----

database password: postgres

Working Commands: Test commands

!project_help

!login bestbuy

!launch_browser

!close_browser

!navigate_to_website https://www.google.com/

!monitor_price https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734

&sav=true

!check_availability https://www.opentable.com/r/bar-spero-washington/

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22"

!stop_monitoring

!stop

**********

Working on it:

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22" "8:00 PM"

**********

URLs to Test:

https://www.opentable.com/r/bar-spero-washington/

https://www.ebay.com/itm/314411766963?_trkparms=amclksrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4

a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3Avlp_homepage

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

----- Contents of test_addAccount.py -----

```
import sys, os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_add_account():
    account_control = AccountControl()

    # Adding a new account
    account_control.add_account("newUser", "newPassword123", "newWebsite")

if __name__ == "__main__":
    test_add_account()
```

----- Contents of test_deleteAccount.py -----

```
import sys
```

```
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_delete_account():
    account_control = AccountControl()

    account_control.delete_account(4)

if __name__ == "__main__":
    test_delete_account()
```

----- Contents of test_excel_creation.py -----

----- Contents of test_fetchAccounts.py -----

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_fetch_accounts():
    account_control = AccountControl()
```

```
    # Fetching all accounts
    account_control.fetch_accounts()

def test_fetch_account_by_website(website):
    account_control = AccountControl()

    # Fetch the account by website directly
    account = account_control.fetch_account_by_website(website)

    if account:
        username, password = account  # Unpack the returned tuple
        print(f"Website: {website}, Username: {username}, Password: {password}")
    else:
        print(f"No account found for website: {website}")

if __name__ == "__main__":
    test_fetch_accounts()
    test_fetch_account_by_website("ebay")
```

----- Contents of test_html_creation.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of css_selectors.py -----

```
class Selectors:
    SELECTORS = {
        "trendyol": {
            "price": ".featured-prices .prc-dsc"  # Selector for Trendyol price
        },
        "ebay": {
            "url": "https://signin.ebay.com/signin/",
            "email_field": "#userid",
            "continue_button": "[data-testid*='signin-continue-btn']",
            "password_field": "#pass",
            "login_button": "#sgnBt",
            "price": ".x-price-primary span"  # CSS selector for Ebay price
        },
        "bestbuy": {
            "url": "https://www.bestbuy.com/signin/",
            "email_field": "#fld-e",
            #"continue_button": ".cia-form__controls  button",
            "password_field": "#fld-p1",
            "SignIn_button": ".cia-form__controls  button",
            "price": "[data-testid='customer-price'] span",  # CSS selector for BestBuy price
            "homePage": ".v-p-right-xxs.line-clamp"
        },
        "opentable": {
            "url": "https://www.opentable.com/",
            "date_field": "#restProfileSideBarDtpDayPicker-label",
            "time_field": "#restProfileSideBartimePickerDtpPicker",
            "find_table_button": ".find-table-button",  # Example selector for the Find Table button
```

```
        "availability_result": ".availability-result",  # Example selector for availability results
            "show_next_available_button": "button[data-test='multi-day-availability-button']",  # Show next available button
        "available_dates": "ul[data-test='time-slots'] > li",  # Available dates and times
        "no_availability": "div._8ye6OVzeOuU- span"
    }
}

@staticmethod
def get_selectors_for_url(url):
    for keyword, selectors in Selectors.SELECTORS.items():
        if keyword in url.lower():
            return selectors
    return None  # Return None if no matching selectors are found
```

----- Contents of DiscordUtils.py -----

----- Contents of ExcelUtils.py -----

----- Contents of HTMLUtils.py -----

----- Contents of css_selectors.cpython-312.pyc -----

----- Contents of HelpText.cpython-312.pyc -----

----- Contents of Config.cpython-312.pyc -----

--- Config.py ---

```
class Config:
                                                    DISCORD_TOKEN                    =
'MTI2OTM4MTE4OTA1NjMzNTk3Mw.Gihcfw.nrq0x-JiL65P0LIQTO-rTyyXq0qC-2PSSBuXr8'
    CHANNEL_ID = 1269383349278081054
    DATABASE_PASSWORD = 'postgres'
```

--- main.py ---

```
import discord
from discord.ext import commands
from boundary.BotBoundary import BotBoundary
from boundary.HelpBoundary import HelpBoundary
from boundary.AccountBoundary import AccountBoundary
from boundary.BrowserBoundary import BrowserBoundary
from boundary.LoginBoundary import LoginBoundary
from boundary.CloseBrowserBoundary import CloseBrowserBoundary
from boundary.StopBoundary import StopBoundary
from boundary.NavigationBoundary import NavigationBoundary
from boundary.PriceBoundary import PriceBoundary
from boundary.MonitorPriceBoundary import MonitorPriceBoundary
from Config import Config

# Set up the bot's intents
intents = discord.Intents.default()
intents.message_content = True  # Enable reading message content
```

```
# Initialize the bot with the correct command prefix and intents
class MyBot(commands.Bot):
    async def setup_hook(self):
        await self.add_cog(BotBoundary(self))
        await self.add_cog(HelpBoundary(self))
        await self.add_cog(AccountBoundary(self))
        await self.add_cog(BrowserBoundary(self))
        await self.add_cog(StopBoundary(self))
        await self.add_cog(LoginBoundary(self))
        await self.add_cog(CloseBrowserBoundary(self))
        await self.add_cog(NavigationBoundary(self))
        await self.add_cog(PriceBoundary(self))
        await self.add_cog(MonitorPriceBoundary(self))

# Run the bot
if __name__ == "__main__":
    bot = MyBot(command_prefix="!", intents=intents)
    print(f"Bot is starting...")
    bot.run(Config.DISCORD_TOKEN)



--- project_summary.txt ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC
699\DiscordBotProject_CISC699\all_files_content.txt ---
----- Contents of all_files_content.txt -----
```

----- Contents of Config.py -----

```
class Config:
                                                    DISCORD_TOKEN                    =
'MTI2OTM4MTE4OTA1NjMzNTk3Mw.Gihcfw.nrq0x-JiL65P0LIQTO-rTyyXq0qC-2PSSBuXr8'
    CHANNEL_ID = 1269383349278081054
    DATABASE_PASSWORD = 'postgres'
```

----- Contents of main.py -----

```
import discord
from discord.ext import commands
from boundary.BotBoundary import BotBoundary
from boundary.HelpBoundary import HelpBoundary
from boundary.AccountBoundary import AccountBoundary
from boundary.BrowserBoundary import BrowserBoundary
from boundary.LoginBoundary import LoginBoundary
from boundary.CloseBrowserBoundary import CloseBrowserBoundary
from boundary.StopBoundary import StopBoundary
from boundary.NavigationBoundary import NavigationBoundary
from boundary.PriceBoundary import PriceBoundary
from boundary.MonitorPriceBoundary import MonitorPriceBoundary
from Config import Config

# Set up the bot's intents
intents = discord.Intents.default()
intents.message_content = True  # Enable reading message content
```

```
# Initialize the bot with the correct command prefix and intents
class MyBot(commands.Bot):
    async def setup_hook(self):
        await self.add_cog(BotBoundary(self))
        await self.add_cog(HelpBoundary(self))
        await self.add_cog(AccountBoundary(self))
        await self.add_cog(BrowserBoundary(self))
        await self.add_cog(StopBoundary(self))
        await self.add_cog(LoginBoundary(self))
        await self.add_cog(CloseBrowserBoundary(self))
        await self.add_cog(NavigationBoundary(self))
        await self.add_cog(PriceBoundary(self))
        await self.add_cog(MonitorPriceBoundary(self))

# Run the bot
if __name__ == "__main__":
    bot = MyBot(command_prefix="!", intents=intents)
    print(f"Bot is starting...")
    bot.run(Config.DISCORD_TOKEN)
```

----- Contents of project_files_text.pdf -----

----- Contents of COMMIT_EDITMSG -----

going succesful with the login and getting prices and everything

----- Contents of config -----

```
[core]
 repositoryformatversion = 0
 filemode = false
 bare = false
 logallrefupdates = true
 symlinks = false
 ignorecase = true
[remote "origin"]
 url = https://github.com/oguzky7/DiscordBotProject_CISC699.git
 fetch = +refs/heads/*:refs/remotes/origin/*
[branch "develop"]
 remote = origin
 merge = refs/heads/develop
 vscode-merge-base = origin/develop
[branch "StartOverAgain"]
 remote = origin
 merge = refs/heads/StartOverAgain
 vscode-merge-base = origin/develop
[branch "StartOver"]
 remote = origin
 merge = refs/heads/StartOver
 vscode-merge-base = origin/develop
```

----- Contents of description -----

```
Unnamed repository; edit this file 'description' to name the repository.
```

----- Contents of FETCH_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69 branch 'develop' of https://github.com/oguzky7/DiscordBotProject_CISC699

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b not-for-merge branch 'StartOver' of https://github.com/oguzky7/DiscordBotProject_CISC699

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 not-for-merge branch 'StartOverAgain' of https://github.com/oguzky7/DiscordBotProject_CISC699

e6f9da804a74f224a4f50336480f1896b3142fde not-for-merge branch 'main' of https://github.com/oguzky7/DiscordBotProject_CISC699

----- Contents of HEAD -----

ref: refs/heads/StartOverAgain

----- Contents of index -----

----- Contents of ORIG_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of packed-refs -----

# pack-refs with: peeled fully-peeled sorted

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b refs/remotes/origin/StartOver

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 refs/remotes/origin/StartOverAgain

dad457b1e774179c151ce94c80cca9d1605d2b69 refs/remotes/origin/develop

e6f9da804a74f224a4f50336480f1896b3142fde refs/remotes/origin/main

----- Contents of applypatch-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.  The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:
```

----- Contents of commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message.
# Called by "git commit" with one argument, the name of the file
# that has the commit message.  The hook should exit with non-zero
```

```
# status after issuing an appropriate message if it wants to stop the
# commit.  The hook is allowed to edit the commit message file.
#
# To enable this hook, rename this file to "commit-msg".

# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"

# This example catches duplicate Signed-off-by lines.

test "" = "$(grep '^Signed-off-by: ' "$1" |
  sort | uniq -c | sed -e '/^[  ]*1[  ]/d')" || {
 echo >&2 Duplicate Signed-off-by lines.
 exit 1
}
```

----- Contents of fsmonitor-watchman.sample -----

```
#!/usr/bin/perl

use strict;
use warnings;
use IPC::Open2;
```

```
# An example hook script to integrate Watchman
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 2) and last update token
# formatted as a string and outputs to stdout a new update token and
# all files that have been modified since the update token. Paths must
# be relative to the root of the working tree and separated by a single NUL.
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $last_update_token) = @ARGV;

# Uncomment for debugging
# print STDERR "$0 $version $last_update_token\n";

# Check the hook interface version
if ($version ne 2) {
 die "Unsupported query-fsmonitor hook version '$version'.\n" .
     "Falling back to scanning...\n";
}

my $git_work_tree = get_working_dir();

my $retry = 1;
```

```
my $json_pkg;
eval {
 require JSON::XS;
 $json_pkg = "JSON::XS";
 1;
} or do {
 require JSON::PP;
 $json_pkg = "JSON::PP";
};

launch_watchman();

sub launch_watchman {
 my $o = watchman_query();
 if (is_work_tree_watched($o)) {
  output_result($o->{clock}, @{$o->{files}});
 }
}

sub output_result {
 my ($clockid, @files) = @_;

 # Uncomment for debugging watchman output
 # open (my $fh, ">", ".git/watchman-output.out");
 # binmode $fh, ":utf8";
 # print $fh "$clockid\n@files\n";
```

```
 # close $fh;

 binmode STDOUT, ":utf8";
 print $clockid;
 print "\0";
 local $, = "\0";
 print @files;
}

sub watchman_clock {
 my $response = qx/watchman clock "$git_work_tree"/;
 die "Failed to get clock id on '$git_work_tree'.\n" .
  "Falling back to scanning...\n" if $? != 0;

 return $json_pkg->new->utf8->decode($response);
}

sub watchman_query {
 my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
 or die "open2() failed: $!\n" .
 "Falling back to scanning...\n";

 # In the query expression below we're asking for names of files that
 # changed since $last_update_token but not from the .git folder.
 #
 # To accomplish this, we're using the "since" generator to use the
 # recency index to select candidate nodes and "fields" to limit the
```

```
# output to file names only. Then we're using the "expression" term to
# further constrain the results.
my $last_update_line = "";
if (substr($last_update_token, 0, 1) eq "c") {
 $last_update_token = "\"$last_update_token\"";
 $last_update_line = qq[\n"since": $last_update_token,];
}
my $query = <<" END";
 ["query", "$git_work_tree", {$last_update_line
  "fields": ["name"],
  "expression": ["not", ["dirname", ".git"]]
 }]
END

# Uncomment for debugging the watchman query
# open (my $fh, ">", ".git/watchman-query.json");
# print $fh $query;
# close $fh;

print CHLD_IN $query;
close CHLD_IN;
my $response = do {local $/; <CHLD_OUT>};

# Uncomment for debugging the watch response
# open ($fh, ">", ".git/watchman-response.json");
# print $fh $response;
# close $fh;
```

```
 die "Watchman: command returned no output.\n" .
 "Falling back to scanning...\n" if $response eq "";
 die "Watchman: command returned invalid output: $response\n" .
 "Falling back to scanning...\n" unless $response =~ /^\{/;

 return $json_pkg->new->utf8->decode($response);
}

sub is_work_tree_watched {
 my ($output) = @_;
 my $error = $output->{error};
 if ($retry > 0 and $error and $error =~ m/unable to resolve root .* directory (.*) is not watched/) {
  $retry--;
  my $response = qx/watchman watch "$git_work_tree"/;
  die "Failed to make watchman watch '$git_work_tree'.\n" .
      "Falling back to scanning...\n" if $? != 0;
  $output = $json_pkg->new->utf8->decode($response);
  $error = $output->{error};
  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  # Uncomment for debugging watchman output
  # open (my $fh, ">", ".git/watchman-output.out");
  # close $fh;

  # Watchman will always return all files on the first query so
```

```
  # return the fast "everything is dirty" flag to git and do the
  # Watchman query just to get it over with now so we won't pay
  # the cost in git to look up each individual file.
  my $o = watchman_clock();
  $error = $output->{error};

  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  output_result($o->{clock}, ("/"));
  $last_update_token = $o->{clock};

  eval { launch_watchman() };
  return 0;
 }

 die "Watchman: $error.\n" .
 "Falling back to scanning...\n" if $error;

 return 1;
}

sub get_working_dir {
 my $working_dir;
 if ($^O =~ 'msys' || $^O =~ 'cygwin') {
  $working_dir = Win32::GetCwd();
  $working_dir =~ tr/\\/\//;
```

```
 } else {
  require Cwd;
  $working_dir = Cwd::cwd();
 }

 return $working_dir;
}
```

----- Contents of post-update.sample -----

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

----- Contents of pre-applypatch.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
```

```
# appropriate message if it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:



----- Contents of pre-commit.sample -----
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
 against=HEAD
else
 # Initial commit: diff against an empty tree object
 against=$(git hash-object -t tree /dev/null)
fi
```

```
# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
if [ "$allownonascii" != "true" ] &&
 # Note that the use of brackets around a tr range is ok here, (it's
 # even required, for portability to Solaris 10's /usr/bin/tr), since
 # the square bracket bytes happen to fall in the designated range.
 test $(git diff --cached --name-only --diff-filter=A -z $against |
   LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
then
 cat <<\EOF
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.

To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:

  git config hooks.allownonascii true
```

```
EOF
 exit 1
fi

# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --
```

----- Contents of pre-merge-commit.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
        exec "$GIT_DIR/hooks/pre-commit"
:
```

----- Contents of pre-push.sample -----

```
#!/bin/sh
```

```
# An example hook script to verify what is about to be pushed.  Called by "git
# push" after it has checked the remote status, but before anything has been
# pushed.  If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
#   <local ref> <local oid> <remote ref> <remote oid>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')

while read local_ref local_oid remote_ref remote_oid
do
 if test "$local_oid" = "$zero"
```

```
 then
  # Handle delete
  :
 else
  if test "$remote_oid" = "$zero"
  then
   # New branch, examine all commits
   range="$local_oid"
  else
   # Update to existing branch, examine new commits
   range="$remote_oid..$local_oid"
  fi

  # Check for WIP commit
  commit=$(git rev-list -n 1 --grep '^WIP' "$range")
  if test -n "$commit"
  then
   echo >&2 "Found WIP commit in $local_ref, not pushing"
   exit 1
  fi
 fi
done

exit 0
```

----- Contents of pre-rebase.sample -----

```
#!/bin/sh
#
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.

publish=next
basebranch="$1"
if test "$#" = 2
then
 topic="refs/heads/$2"
else
 topic=`git symbolic-ref HEAD` ||
 exit 0 ;# we do not interrupt rebasing detached HEAD
fi
```

```
case "$topic" in
refs/heads/??/*)
 ;;
*)
 exit 0 ;# we do not interrupt others.
 ;;
esac

# Now we are dealing with a topic branch being rebased
# on top of master.  Is it OK to rebase it?

# Does the topic really exist?
git show-ref -q "$topic" || {
 echo >&2 "No such branch $topic"
 exit 1
}

# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
 echo >&2 "$topic is fully merged to master; better remove it."
 exit 1 ;# we could allow it, but there is no point.
fi

# Is topic ever merged to next?  If so you should not be rebasing it.
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
```

```
only_next_2=`git rev-list ^master           ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
then
 not_in_topic=`git rev-list "^$topic" master`
 if test -z "$not_in_topic"
 then
  echo >&2 "$topic is already up to date with master"
  exit 1 ;# we could allow it, but there is no point.
 else
  exit 0
 fi
else
 not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
 /usr/bin/perl -e '
  my $topic = $ARGV[0];
  my $msg = "* $topic has commits already merged to public branch:\n";
  my (%not_in_next) = map {
   /^([0-9a-f]+) /;
   ($1 => 1);
  } split(/\n/, $ARGV[1]);
  for my $elem (map {
    /^([0-9a-f]+) (.*)$/;
    [$1 => $2];
   } split(/\n/, $ARGV[2])) {
   if (!exists $not_in_next{$elem->[0]}) {
    if ($msg) {
     print STDERR $msg;
```

```
    undef $msg;
   }
   print STDERR " $elem->[1]\n";
  }
 }
' "$topic" "$not_in_next" "$not_in_master"
 exit 1
fi

<<\DOC_END

This sample hook safeguards topic branches that have been
published from being rewound.

The workflow assumed here is:

 * Once a topic branch forks from "master", "master" is never
   merged into it again (either directly or indirectly).

 * Once a topic branch is fully cooked and merged into "master",
   it is deleted.  If you need to build on top of it to correct
   earlier mistakes, a new topic branch is created by forking at
   the tip of the "master".  This is not strictly necessary, but
   it makes it easier to keep your history simple.

 * Whenever you need to test or publish your changes to topic
   branches, merge them into "next" branch.
```

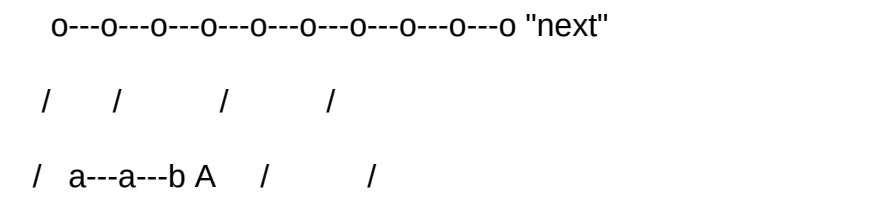
The script, being an example, hardcodes the publish branch name
to be "next", but it is trivial to make it configurable via
$GIT_DIR/config mechanism.

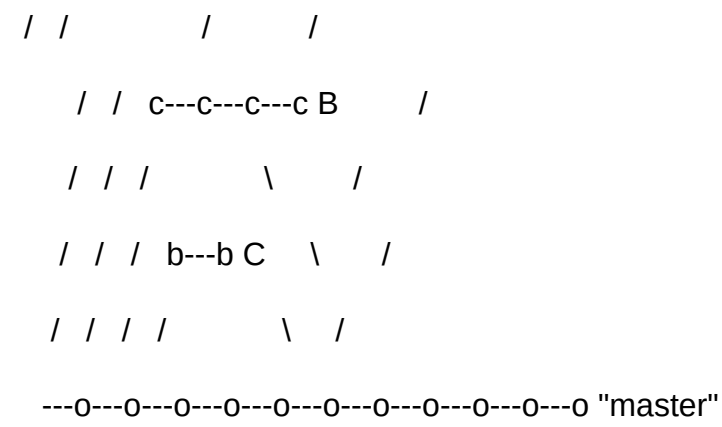
With this workflow, you would want to know:

(1) ... if a topic branch has ever been merged to "next". Young
topic branches can have stupid mistakes you would rather
clean up before publishing, and things that have not been
merged into other branches can be easily rebased without
affecting other people. But once it is published, you would
not want to rewind it.

(2) ... if a topic branch has been fully merged to "master".
Then you can delete it. More importantly, you should not
build on top of it -- other people may already want to
change things related to the topic as patches against your
"master", so if you need further changes, it is better to
fork the topic (perhaps with the same name) afresh from the
tip of "master".

Let's look at this example:






A, B and C are topic branches.

* A has one fix since it was merged up to "next".

* B has finished. It has been fully merged up to "master" and "next",
and is ready to be deleted.

* C has not merged to "next" at all.

We would want to allow C to be rebased, refuse A, and encourage
B to be deleted.

To compute (1):

git rev-list ^master ^topic next
git rev-list ^master next

if these match, topic has not merged in next at all.

To compute (2):

```
 git rev-list master..topic
```

```
 if this is empty, it is fully merged to "master".
```

```
DOC_END
```

----- Contents of pre-receive.sample -----

```
#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"
then
 i=0
 while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
 do
  eval "value=\$GIT_PUSH_OPTION_$i"
  case "$value" in
  echoback=*)
   echo "echo from the pre-receive-hook: ${value#*=}" >&2
```

```
    ;;
  reject)
    exit 1
  esac
  i=$((i + 1))
 done
fi
```

----- Contents of prepare-commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source.  The hook's purpose is to edit the commit
# message file.  If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
# into the message, just before the "git status" output.  It is
# commented because it doesn't cope with --amend or with squashed
```

```
# commits.
#
# The third example adds a Signed-off-by line to the message, that can
# still be edited.  This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3

/usr/bin/perl -i.bak -ne 'print unless(m/^. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FILE"

# case "$COMMIT_SOURCE,$SHA1" in
#  ,|template,)
#    /usr/bin/perl -i.bak -pe '
#       print "\n" . `git diff --cached --name-status -r`
#   if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
#  *) ;;
# esac

# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
#   /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
# fi
```

----- Contents of push-to-checkout.sample -----

```
#!/bin/sh

# An example hook script to update a checked-out tree on a git push.
#
# This hook is invoked by git-receive-pack(1) when it reacts to git
# push and updates reference(s) in its repository, and when the push
# tries to update the branch that is currently checked out and the
# receive.denyCurrentBranch configuration variable is set to
# updateInstead.
#
# By default, such a push is refused if the working tree and the index
# of the remote repository has any difference from the currently
# checked out commit; when both the working tree and the index match
# the current commit, they are updated to match the newly pushed tip
# of the branch. This hook is to be used to override the default
# behaviour; however the code below reimplements the default behaviour
# as a starting point for convenient modification.
#
# The hook receives the commit with which the tip of the current
# branch is going to be updated:
commit=$1

# It can exit with a non-zero status to refuse the push (when it does
# so, it must not modify the index or the working tree).
die () {
```

```
 echo >&2 "$*"
 exit 1
}

# Or it can make any necessary changes to the working tree and to the
# index to bring them to the desired state when the tip of the current
# branch is updated to the new commit, and exit with a zero status.
#
# For example, the hook can simply run git read-tree -u -m HEAD "$1"
# in order to emulate git fetch that is run in the reverse direction
# with git push, as the two-tree form of git read-tree -u -m is
# essentially the same as git switch or git checkout that switches
# branches while keeping the local changes in the working tree that do
# not interfere with the difference between the branches.

# The below is a more-or-less exact translation to shell of the C code
# for the default behaviour for git's push-to-checkout hook defined in
# the push_to_deploy() function in builtin/receive-pack.c.
#
# Note that the hook will be executed from the repository directory,
# not from the working tree, so if you want to perform operations on
# the working tree, you will have to adapt your code accordingly, e.g.
# by adding "cd .." or using relative paths.

if ! git update-index -q --ignore-submodules --refresh
then
 die "Up-to-date check failed"
```

```
fi

if ! git diff-files --quiet --ignore-submodules --
then
 die "Working directory has unstaged changes"
fi

# This is a rough translation of:
#
#   head_has_history() ? "HEAD" : EMPTY_TREE_SHA1_HEX
if git cat-file -e HEAD 2>/dev/null
then
 head=HEAD
else
 head=$(git hash-object -t tree --stdin </dev/null)
fi

if ! git diff-index --quiet --cached --ignore-submodules $head --
then
 die "Working directory has staged changes"
fi

if ! git read-tree -u -m "$commit"
then
 die "Could not update working tree to new HEAD"
fi
```

----- Contents of update.sample -----

```
#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# ------
# hooks.allowunannotated
#   This boolean sets whether unannotated tags will be allowed into the
#   repository.  By default they won't be.
# hooks.allowdeletetag
#   This boolean sets whether deleting tags will be allowed in the
#   repository.  By default they won't be.
# hooks.allowmodifytag
#   This boolean sets whether a tag may be modified after creation. By default
#   it won't be.
# hooks.allowdeletebranch
#   This boolean sets whether deleting branches will be allowed in the
#   repository.  By default they won't be.
# hooks.denycreatebranch
#   This boolean sets whether remotely creating branches will be denied
#   in the repository.  By default this is allowed.
#
```

```
# --- Command line
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
 echo "Don't run this script from the command line." >&2
 echo " (if you want, you could supply GIT_DIR then run" >&2
 echo "  $0 <ref> <oldrev> <newrev>)" >&2
 exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
 echo "usage: $0 <ref> <oldrev> <newrev>" >&2
 exit 1
fi

# --- Config
allowunannotated=$(git config --type=bool hooks.allowunannotated)
allowdeletebranch=$(git config --type=bool hooks.allowdeletebranch)
denycreatebranch=$(git config --type=bool hooks.denycreatebranch)
allowdeletetag=$(git config --type=bool hooks.allowdeletetag)
allowmodifytag=$(git config --type=bool hooks.allowmodifytag)

# check for no description
```

```
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
case "$projectdesc" in
"Unnamed repository"* | "")
 echo "*** Project description file hasn't been set" >&2
 exit 1
 ;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')
if [ "$newrev" = "$zero" ]; then
 newrev_type=delete
else
 newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
 refs/tags/*,commit)
  # un-annotated tag
  short_refname=${refname##refs/tags/}
  if [ "$allowunannotated" != "true" ]; then
   echo "*** The un-annotated tag, $short_refname, is not allowed in this repository" >&2
   echo "*** Use 'git tag [ -a | -s ]' for tags you want to propagate." >&2
   exit 1
  fi
  ;;
```

```
refs/tags/*,delete)
 # delete tag
 if [ "$allowdeletetag" != "true" ]; then
  echo "*** Deleting a tag is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/tags/*,tag)
 # annotated tag
 if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
 then
  echo "*** Tag '$refname' already exists." >&2
  echo "*** Modifying a tag is not allowed in this repository." >&2
  exit 1
 fi
 ;;
refs/heads/*,commit)
 # branch
 if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
  echo "*** Creating a branch is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/heads/*,delete)
 # delete branch
 if [ "$allowdeletebranch" != "true" ]; then
  echo "*** Deleting a branch is not allowed in this repository" >&2
```

```
   exit 1
  fi
  ;;
 refs/remotes/*,commit)
  # tracking branch
  ;;
 refs/remotes/*,delete)
  # delete tracking branch
  if [ "$allowdeletebranch" != "true" ]; then
   echo "*** Deleting a tracking branch is not allowed in this repository" >&2
   exit 1
  fi
  ;;
 *)
  # Anything else (is there anything else?)
  echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
  exit 1
  ;;
esac

# --- Finished
exit 0
```

```
----- Contents of exclude -----
# git ls-files --others --exclude-from=.git/info/exclude
# Lines that start with '#' are comments.
```

# For a project mostly in C, the following would be a good set of

# exclude patterns (uncomment them if you want to use them):

# *.[oa]

# *~

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

dad457b1e774179c151ce94c80cca9d1605d2b69 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 checkout: moving from develop to StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 checkout: moving from StartOverAgain to StartOver

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6

oguzky7 <oguzky7@gmail.com> 1725654644 -0500 checkout: moving from StartOver to StartOverAgain

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08 oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42 oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of develop -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOver -----

0000000000000000000000000000000000000000 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 branch: Created from refs/remotes/origin/StartOver

----- Contents of StartOverAgain -----

0000000000000000000000000000000000000000 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 branch: Created from refs/remotes/origin/StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed

oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOverAgain -----

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d

oguzky7 <oguzky7@gmail.com> 1725649275 -0500 update by push

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681

oguzky7 <oguzky7@gmail.com> 1725652043 -0500 update by push

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9

oguzky7 <oguzky7@gmail.com> 1725653629 -0500 update by push

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6

oguzky7 <oguzky7@gmail.com> 1725654567 -0500 update by push

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483

oguzky7 <oguzky7@gmail.com> 1725658344 -0500 update by push

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec

oguzky7 <oguzky7@gmail.com> 1725664583 -0500 update by push

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed

oguzky7 <oguzky7@gmail.com> 1725666447 -0500 update by push

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723082 -0500 update by push

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725779 -0500 update by push

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731942 -0500 update by push

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732309 -0500 update by push

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740312 -0500 update by push

----- Contents of 6d281c79b57abc36be3229d4bbab976d1b6bf3 -----

----- Contents of 634f7e652d3fb3b816adc257a0161cfaa59896 -----

----- Contents of d30f9eeb85ec53d1d3fe51515a9471ee79af75 -----

----- Contents of da87394b264ec22cc557849312040a03c1063e -----

----- Contents of acbb56d15479249f6c46de8e3acb0c8b20cb93 -----

----- Contents of cd8b5e55c2ae7f362dc0c8436c4b89a275da90 -----

----- Contents of 56de35ec5b020a6ed5805e2520c51bac37d017 -----

----- Contents of 80ceb22ccad5017ec7c7873c96e5203198cb7b -----

----- Contents of ae88962bd7ef5ee6892e9dbd1761ec6d56b7d1 -----

----- Contents of 0d5a7309d480be697278ca6d58e46a30f02681 -----

----- Contents of 2f7fda7f6f8ecf990caa0731e5be154b2adc39 -----

----- Contents of 670af298f99b15bd9dd09da31d72f0f1d38cdd -----

----- Contents of 707c217ce7f6abc9094c93c766f29b6eab99e4 -----

----- Contents of 73684a5a9f9b5ebf1227f6801d4e478fe16d4a -----

----- Contents of d21b42548625adec2ddceedb034d770341a8c8 -----

----- Contents of d19a6ce55b82d3b656da482d1c8e6a0b54b75b -----

----- Contents of 12668a4da67180c42fa2fab9cd1de9a032b227 -----

----- Contents of 8387cc096c39bd2853df1c025f6cda162264e7 -----

----- Contents of 7bedd012010871eb4fb46ee52ec2ec4bd1c14f -----

----- Contents of f7ffd8161a083f528f9374f3a9a394498db932 -----

----- Contents of 5f314f257340ec9bfbb711c9d13737f9423489 -----

----- Contents of 691ffa09c5106ba04323f77fd727a278646d28 -----

----- Contents of b53adee5cf0e0bccf2f8ad73d337a79dc020d4 -----

----- Contents of d862acaeafd794c8d81a637c85da6be6878e7d -----

----- Contents of 52beb42325a46c9fbd5a5e3e3fa9f54432bd3b -----

----- Contents of 5795b3882003e2fb6801781de44d8bf296680c -----

----- Contents of 740f61a43018fb8b31d987eb2c5e6c674ce9fc -----

----- Contents of 29623a9d92dc8f9e81e3e27a2deade8f63d70f -----

----- Contents of 7fa5fe2e9d0be5e56ceb6c07452184fccca8ad -----

----- Contents of e571a06a385f9280d5a100f60a2d1ffba1f221 -----

----- Contents of 08c8132fd411e30656c37d89b16b6dcd8b22f4 -----

----- Contents of 0c06257cbe7ea26c5c9ce349da25220ae8d9e2 -----

----- Contents of 0dfcd25d3efecd8aac12bcc1a41befb30deaf7 -----

----- Contents of 4dd067b87b28cc9a8cca80caac89fc53036596 -----

----- Contents of c2fabce9b84c9ba79bc2e79a1ac96cc34a1003 -----

----- Contents of 7bbbe8178759121dbcda100e7c83a2c1d493e9 -----

----- Contents of 6c22f5474f34c8623c296e4b6214d003eb3b08 -----

----- Contents of 56b260e1d2b7fd86f354e6fff1b011231cd844 -----

----- Contents of c1291d130ce613ec78e688bf51a576d5efc5c3 -----

----- Contents of e689cbc21247f718c17658183158af910d05d8 -----

----- Contents of 2e280b0131be8c9e98b7222577acd8341d8fa5 -----

----- Contents of e6f8d57f1f260304e67a6b0a0286689893b8e8 -----

----- Contents of c0afdc9c0e97448eb03cd4b9dd9ce145fd8d0b -----

----- Contents of 57837b989b94d629bfc5a49ba5fc0e8261e483 -----

----- Contents of c230c6518a1ad0586efaa11fd03274ae402a27 -----

----- Contents of 22650605791745310722ebcce1ece57872317e -----

----- Contents of 16855907d1ddccf87830cc3220e0815d2043d6 -----

----- Contents of 7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed -----

----- Contents of 8be98f50f5ab158fff2c10db14f835a386fc99 -----

----- Contents of 5847fd8df25f2a9b603ed33703a21e91ba67c0 -----

----- Contents of 79b9a56dfb3a07e75198f466fceca1c3d8c1de -----

----- Contents of 7c13cbb1ada6eeb37aa304a9f8afb8744a47dc -----

----- Contents of 124da30b98b65e0df305d0900c2d1282b7d270 -----

----- Contents of 1cd1eadaa24a9a3f632873c244100ee7025794 -----

----- Contents of 36f14349b3a4410888ac1b43070bba46de9ede -----

----- Contents of 6ebe87b1695efcfc11786fb574e0f04bca1d18 -----

----- Contents of ddf2dd50c642602c7eba5268be402b32f66989 -----

----- Contents of 36742a002f99203e8212f1fba1a445a1bb03f6 -----

----- Contents of 623cd7312f1ff9c8abe8832fd36f20e6b84bcd -----

----- Contents of 7461d191d8a7a5de9217e88c6c140c95dd9bca -----

----- Contents of e326319a500dd5126f6173fdc28861abf9de10 -----

----- Contents of 167ddd3cb16efd9af1c2cd4539382ccd57e2fd -----

----- Contents of a951080c6be53c94cb18b588e50d8628833d3e -----

----- Contents of f3b3a6001ee55883e947e28ac1e2c7b08387bd -----

----- Contents of 2778616f503b20c4ebcb0ddd0acfaf1de01e39 -----

----- Contents of 2f14edff81d9dd2f05f2d2f8c7cd365d932bbc -----

----- Contents of 20fcb37bf59941b57c5a694a939d13d15b8900 -----

----- Contents of 98a6460ae86feb9f8049f407d2f6887424e6dc -----

----- Contents of 374847dfb20aff248fdcd59f87bdea40697b1d -----

----- Contents of f89b3afaa02365ae60cb072cf76abf1cf77ae9 -----

----- Contents of 81d6b895e5ee4ada44b133c81c73d79b001b04 -----

----- Contents of e8393441a7d2050cce36faa6564fc5cd9763ec -----

----- Contents of e434a795adc8737398a481f4c813fc380cebf3 -----

----- Contents of 7bdce5089b6b1b983d2a088b0b1a2bb2ebc56c -----

----- Contents of 790b96b9de32b415393a1c2aed46ffb49e6ae2 -----

----- Contents of e495a99648be872776623d90e76250b9cb227e -----

----- Contents of 92fdb8f360a7f817c203304d90f8e42bc44ba9 -----

----- Contents of 922e794ba24b02220cd9bb8547894d915a6e6c -----

----- Contents of bbfb724cd9bc1af776578e20073f1ec227dcfd -----

----- Contents of d010362b749bd8b0940c42084cc7192ed7d944 -----

----- Contents of 2e28edc4f64174222c015a31044054e9302f13 -----

----- Contents of 31e9454a465493fb8c05d28c52697757037d5e -----

----- Contents of c2bb829e59ca69e85259035c1e808b2c17b5b1 -----

----- Contents of 7a34d8216bffdd74b60e3e759e9ade11991cd4 -----

----- Contents of d705b2782872635fbf52906a037a130f467c0f -----

----- Contents of e8bfbb3ec7cce8b3945b1cd33efb7449fb8203 -----

----- Contents of 4c76280fb6b9eefe4f99a0dc7cc4719b85c93d -----

----- Contents of 7feec2f1765e89d736a9359a5c5e551b5f9c6d -----

----- Contents of d386fadadd939cc18d101127a21d961b148a2f -----

----- Contents of 23564225a6a0b77d1fd84d4c3b7e0800177c88 -----

----- Contents of a49e2dcd1c2d8824be05f5a7fa8d279c1c8be3 -----

----- Contents of 7422ef6e171a6dff0d231427b2d364d2dd5bb6 -----

----- Contents of b4cc5d5bf0c7aa29eb4ac829400843fcd51562 -----

----- Contents of 55b982b96b526b4e3db1ee1fe9375b73bba374 -----

----- Contents of ce2369a09426c22593285c67f91cfdcb5011ce -----

----- Contents of 4aa1ac0374fdefe7baf0c2ab6f1c65140b3190 -----

----- Contents of 5e2674b3cb3a1bed397efd15b859641dae7757 -----

----- Contents of 3184272755e64a6944c9869379c6f57ad9ba3b -----

----- Contents of f2c49d67157e02272e6521a0acb70994aff7a7 -----

----- Contents of d105aab298c832e87d7cd64886775dabf92683 -----

----- Contents of 36b1cc2ef835a0eb0f8dab3f76b700dee1ca9c -----

----- Contents of bb301025e8f69ddd490791a3c4b9fe54340715 -----

----- Contents of 3f46aa910385ec1c113d29d13a51b5933246f0 -----

----- Contents of 569d2ce7f8e2dbd43c62f90d8c1a4776fe3657 -----

----- Contents of 35ac6cdff50420307d1f6a29ae6fbce704431b -----

----- Contents of 49235dfb1543a8846b9a4317a4f3b2330af7f5 -----

----- Contents of e4df3e7aa9aa939067d082dc81129d8797e8b9 -----

----- Contents of 15d9428c470f960b9f68228b7e6e037d6fffdb -----

----- Contents of 55c1a5a11c29d9b7088fb5fae913582b80072e -----

----- Contents of 5963004ac6a7c723c76e303c6a933a7b60294e -----

----- Contents of 45a539b3b325f9d4078b26f7788fa7e4e20f4c -----

----- Contents of 6f2e8f648efc155b28e6a77f84c6d4a144f6f6 -----

----- Contents of aaf5c01d5dfff33e76d47f53e18763ae113efd -----

----- Contents of 870dde41518a8ea43fcc6a8b0c64fbf62f65f0 -----

----- Contents of 31e65729c1b72967bcf8d1bba9f4de309309ac -----

----- Contents of ffdef56d17933b2cfc82c5ad7b1fbe98641040 -----

----- Contents of 42bbb4484463cf439fba484bd82cda68230d00 -----

----- Contents of 93d2668edf7fa19ebd76cb7fd46619b412edb8 -----

----- Contents of 65f80d0fa9995fb5cd1505370703edba752087 -----

----- Contents of d12146fcaf19d62254656fde8a206024bd645e -----

----- Contents of fef29125046ebacc0b5bd63cdd89833f67d207 -----

----- Contents of 35fa7f7ad4a8914f3ff46466b98e724a942aad -----

----- Contents of 6cad680be43e263296434712ccfa7cb8683608 -----

----- Contents of 9c8e7e45537aec2443783b44ffe32a2a1d4038 -----

----- Contents of df0c2c3f663c5e0fdb705c6386fbac3c585e79 -----

----- Contents of df6da44831a6c19badb226138612e404e72fde -----

----- Contents of 95216415e1e4b423dbb14fc246d1c13f9af72c -----

----- Contents of 6048147042c058cc22c5089e6d17e8e82247e1 -----

----- Contents of cb233a35c89247d61009b08fdfbba29c952bb8 -----

----- Contents of a97486baf65dd5f97c2e2029485456bcd53875 -----

----- Contents of db4e4cb78b56559a72aca093e43364ab09e531 -----

----- Contents of dd929f339e858c99f485675a0cfd352b143313 -----

----- Contents of 134b6a845b8ff447308d5cfa26acf1abecff81 -----

----- Contents of a8744a97a8497b5a971774408b4f13efb2d36b -----

----- Contents of ce779286a458df17bc46ac6893fc66f279c068 -----

----- Contents of 1bfbc86a74447106a2a964f01541b0dc34c2b4 -----

----- Contents of 2b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb -----

----- Contents of 2f8994666a261dade3f5a5c6edb13ce71f5f05 -----

----- Contents of 4fc156385de559702c75ad60d5a7a344792805 -----

----- Contents of a3700d5e8f8bb3918aa071f60ea5ded21f77d5 -----

----- Contents of bc23c3f2b221779c8bc2c90ffe3708d2171c48 -----

----- Contents of 04f9129475b746303fc616658fc3416237821d -----

----- Contents of d073d6abfbf754fd97b9329f87cec83207ee08 -----

----- Contents of e3c2abaabea3cafb640d47d6f9dc3030186fd2 -----

----- Contents of 547b1e262f315605e9d0b4d65983e72b19449c -----

----- Contents of f3d9725de4b2121766be9d9099dd28a04183e2 -----

----- Contents of 83d1857455ca55b2a4d164f109b372f65b04d4 -----

----- Contents of dcc0a73865688f0a6a2f34ce14f06bb20b5411 -----

----- Contents of 6062c4cd437cf0f7d9498f328b3cc7c39a7aab -----

----- Contents of 00b5e53f01be9fb9c3c8d49599d76948afc966 -----

----- Contents of 871be66b6a4a8f40fd9963b283d8369781dcc2 -----

----- Contents of a99ac7ab128d7cb26d358957b6a913cb17cdce -----

----- Contents of 37d15e04101d6d5747aa61b581a89ae2847b4d -----

----- Contents of 9c78053718a20d8cdf5995cb89f455dbd6becc -----

----- Contents of bd7d94612b16f8217360592a582f490e9d348c -----

----- Contents of 2287757c4e2c0b3c1e391536abb9d9a7b606a0 -----

----- Contents of 89cb5b54aa1d718008b7cd8d4ddc73fbff1c4d -----

----- Contents of 251901266fac8881c65424cf4c60ec63098f04 -----

----- Contents of 6c9ae2c2205d4c1d6823bb667316f36832129d -----

----- Contents of 93a5648f3b5744e544ade4c54a9b2558de33df -----

----- Contents of cf8ddc0f32beaf6deb845b818a237d4e2b12bc -----

----- Contents of 5bcb586c22d55e4d9b45010159cfbe1eb97b3e -----

----- Contents of 11500072f70a22804f86198228a9bf6f4b1723 -----

----- Contents of 212707bdf961cf6127136e77ce1663ca7fa0ac -----

----- Contents of 5a2099c5e29e57c313a0d69bf170ddc1dc7773 -----

----- Contents of 5ee0932ba1d58cdcced9dbc58b5061e0cdc525 -----

----- Contents of 5fc4dc3b398fe65a45236bb26508673b60ab0a -----

----- Contents of 3b2af53524bb1eadba3659d4a3e8e3e8cef960 -----

----- Contents of 5374ec1094c5935714866aa2a07d319cabb50d -----

----- Contents of c45b719c2f6ef47b5a79a6ddeb192ec3c43c9c -----

----- Contents of 3e220c1a0abd5c31bf03f3488ee49da7ba5791 -----

----- Contents of cd6e47d89f170e14c1b089b027c2bfa32a0514 -----

----- Contents of 1019e219967d1a617e593c49eab9dbbc89302d -----

----- Contents of 1b4632f741d513f68f5c9f2147e564f8e6f44e -----

----- Contents of 1b54931841d32971dbfd232f771191f49cea62 -----

----- Contents of 4917f7466bac6768c4b2c970163f10a3da883c -----

----- Contents of d08c9e2e5e8725a91bab9594774b6acda8c235 -----

----- Contents of 83cb74283680914278ac74988973d92220e580 -----

----- Contents of 2db1c7d8b292deaabb4f080f8e287e18c3fc38 -----

----- Contents of 3a8d669dcddfa7a3373422dc3c136dd1f70025 -----

----- Contents of 07750681d74929e692538a74117db1b1779b97 -----

----- Contents of 238ce60d725df8031818872526f17bdc42af50 -----

----- Contents of 7c546782379dc48fa0d181dcffc313b3822924 -----

----- Contents of 0f48a503d03a35ed438b638a1deb8fac419f42 -----

----- Contents of 2104def8712b4ee3d0cdd2a157cf0221c9b77b -----

----- Contents of 04dbf40d154e56070b1344ba8c3cfc5f564746 -----

----- Contents of 6b2e6e73058c710b05b2893bcf4a7e98b7bbad -----

----- Contents of b55569ac58142006b2bd8a09e4e761d57805c1 -----

----- Contents of 25470382adde44f12e3109bd5934c2eecdc918 -----

----- Contents of 4685b2534290d4b5f0cf6ca0d12fabf5d79e8b -----

----- Contents of 751a6ec93ec6beea13aff04bba59d57bf0ea88 -----

----- Contents of 580bd42a3222857aea5885bed9215a1d05b7c6 -----

----- Contents of 964410b7eebddb7e50578e6ccb13c7e8ab7f27 -----

----- Contents of 4c944d46b01fc2734fe68f52ebfde0cb7998e5 -----

----- Contents of 12fb46a5acc7b60a63fc627bdf6f2791543d58 -----

----- Contents of 3237d5a3531bff37efac2e869560a125ece585 -----

----- Contents of 5eea14070cdff5e8b36d391984776aa89aab44 -----

----- Contents of 7ae1b68f35a89244d46925d6cfebd4b74e6c32 -----

----- Contents of f6136b4665613b61ce58bf20a3ceb85d188776 -----

----- Contents of 83d94a59451c4c9a239ac1ae95b06ecb5587d1 -----

----- Contents of 73fd1a7d3b7afbd81ec494aac0cf63725c97f4 -----

----- Contents of f2528590e95cab41dc7026b769871a74e85090 -----

----- Contents of 10cd67f2809bd3968b90699d6e8e7bc7ce438e -----

----- Contents of b96482d8c2ef82f77977c56260947c27516f67 -----

----- Contents of 4b722e77016ace745c5592948cbe8e0a3cb033 -----

----- Contents of ad546333c05b7217d396c913e59ca48241fcc9 -----

----- Contents of 1fd6e881aef18349e7d110677b3472dcacf2a4 -----

----- Contents of 9ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 -----

----- Contents of 41098bb377a0dfca7f862a6009ece6f6d30e4a -----

----- Contents of 78506479aeaf02315ca23e1ce3992b0af824be -----

----- Contents of bbcb0d6a2e1593e7f104095475e4d7d151d091 -----

----- Contents of 9de29bb2d1d6434b8b29ae775ad8c2e48c5391 -----

----- Contents of 09778dc77ac090a0d932499a2232daf1c94abf -----

----- Contents of 2b50c836c96adfa7f3259d99b4b4eb9f2d00e3 -----

----- Contents of 3920a9f5f3d14ccd1ce1c75d288a4e47498d64 -----

----- Contents of b48fdfb32f07250c2d917eef0d362fb6acb61a -----

----- Contents of c168dfbd74cce6f87298b1bcf0a4760533422a -----

----- Contents of 6c3d41c4698dc82ac757363e6998cba6f17c55 -----

----- Contents of b96d1f73ddca7d36a459c4fa161229ab6c68bb -----

----- Contents of db22b05f99c92e086de39ff9302bc5bd942eec -----

----- Contents of 68ca09597ba99ba5a13c4fe0f4e20ad40ea4e1 -----

----- Contents of 17e0509fe70c56442d04a203f7ccc565de0921 -----

----- Contents of 1a629ef9b315d397c4f1bf56b2bbbfc143ff49 -----

----- Contents of 872a50ce0cb03d65925b2a183db893a95ac571 -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.idx -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.pack -----

----- Contents of develop -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of StartOver -----

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of HEAD -----

ref: refs/remotes/origin/develop

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of AccountBoundary.py -----

from discord.ext import commands

from control.AccountControl import AccountControl

```
class AccountBoundary(commands.Cog):

    def __init__(self, bot):
        self.bot = bot
        self.account_control = AccountControl()



    @commands.command(name='fetch_accounts')
    async def fetch_accounts(self, ctx):
        """Fetch and display all accounts."""
        accounts = self.account_control.fetch_accounts()

        # Send each account or the no accounts message to Discord
        for account in accounts:
            await ctx.send(account)



    @commands.command(name="add_account")
    async def add_account(self, ctx, username: str, password: str):
        """Add a new user account to the database."""
        result = self.account_control.add_account(username, password)
        if result:
            await ctx.send(f"Account for {username} added successfully.")
        else:
            await ctx.send(f"Failed to add account for {username}.")
```

```
    @commands.command(name="delete_account")
    async def delete_account(self, ctx, user_id: int):
        """Delete a user account from the database."""
        result = self.account_control.delete_account(user_id)
        if result:
            await ctx.send(f"Account with ID {user_id} deleted successfully.")
        else:
            await ctx.send(f"Failed to delete account with ID {user_id}.")
```

----- Contents of AvailabilityBoundary.py -----

----- Contents of BotBoundary.py -----

```
from discord.ext import commands
from control.ChatControl import ChatControl
from Config import Config

class BotBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.chat_control = ChatControl()

    @commands.Cog.listener()
    async def on_ready(self):
```

```
        """Bot startup message when ready."""
        print(f'Logged in as {self.bot.user.name}')
        channel = self.bot.get_channel(Config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online!")

    @commands.Cog.listener()
    async def on_message(self, message):
        """Handle non-prefixed messages and command-prefixed messages."""
        if message.author == self.bot.user:
            return

        # Handle non-prefixed messages (like greetings)
        if not message.content.startswith('!'):
            response = self.chat_control.process_non_prefixed_message(message.content)
            await message.channel.send(response)

    @commands.Cog.listener()
    async def on_command_error(self, ctx, error):
        """Handle unrecognized commands."""
        if isinstance(error, commands.CommandNotFound):
            # Handle unknown command
            response = self.chat_control.handle_unrecognized_command()
            await ctx.send(response)
```

----- Contents of BrowserBoundary.py -----

```
from discord.ext import commands
from control.BrowserControl import BrowserControl

class BrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.browser_control = BrowserControl()

    @commands.command(name='launch_browser')
    async def launch_browser(self, ctx, *args):
        """Command to launch the browser."""
        incognito = "incognito" in args
        response = self.browser_control.launch_browser(ctx.author, incognito)
        await ctx.send(response)
```

----- Contents of CloseBrowserBoundary.py -----

```
from discord.ext import commands
from control.CloseBrowserControl import CloseBrowserControl

class CloseBrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.close_browser_control = CloseBrowserControl()

    @commands.command(name='close_browser')
    async def close_browser(self, ctx):
```

```
        """Command to close the browser."""
        response = self.close_browser_control.close_browser()
        await ctx.send(response)


----- Contents of DataExtractionBoundary.py -----


----- Contents of HelpBoundary.py -----
from discord.ext import commands
from control.HelpControl import HelpControl

class HelpBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.help_control = HelpControl()

    @commands.command(name='project_help')
    async def project_help(self, ctx):
        """Handles the project_help command."""
        help_message = self.help_control.get_help_message()
        await ctx.send(help_message)


----- Contents of LoginBoundary.py -----
from discord.ext import commands
from control.LoginControl import LoginControl
```

```
class LoginBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.login_control = LoginControl()

    @commands.command(name='login')
    async def login(self, ctx, site: str, *args):
        """Command to log into a website using stored credentials."""
        incognito = "incognito" in args
        retries = next((int(arg) for arg in args if arg.isdigit()), 1)
        response = await self.login_control.login(site, incognito, retries)
        await ctx.send(response)
```

----- Contents of MonitorPriceBoundary.py -----

```
from discord.ext import commands
from control.MonitorPriceControl import MonitorPriceControl

class MonitorPriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.monitor_price_control = MonitorPriceControl()

    @commands.command(name='monitor_price')
    async def monitor_price(self, ctx, url: str, frequency: int = 1):
        """Command to monitor the price at regular intervals."""
```

```
        await self.monitor_price_control.monitor_price(ctx, url, frequency)
```

----- Contents of NavigationBoundary.py -----

```
from discord.ext import commands
from control.NavigationControl import NavigationControl

class NavigationBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.navigation_control = NavigationControl()

    @commands.command(name='navigate_to_website')
    async def navigate_to_website(self, ctx, url: str):
        """Command to navigate to a specified URL."""
        response = self.navigation_control.navigate_to_url(url)
        await ctx.send(response)
```

----- Contents of NotificationBoundary.py -----

----- Contents of PriceBoundary.py -----

```
from discord.ext import commands
from control.PriceControl import PriceControl

class PriceBoundary(commands.Cog):
```

```
    def __init__(self, bot):
        self.bot = bot
        self.price_control = PriceControl()

    @commands.command(name='get_price')
    async def get_price(self, ctx, url: str):
        """Command to get the price from the given URL."""
        response = await self.price_control.get_price(ctx, url)
        await ctx.send(response)



----- Contents of StopBoundary.py -----
from discord.ext import commands
from control.BotControl import BotControl

class StopBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.bot_control = BotControl(bot)

    @commands.command(name="stop_bot")
    async def stop_bot(self, ctx):
        """Handles the stop command and gracefully shuts down the bot."""
        await ctx.send("Stopping the bot...")
        await self.bot_control.stop_bot()
```

----- Contents of StopMonitoringBoundary.py -----

----- Contents of __init__.py -----

#empty init file

----- Contents of AccountBoundary.cpython-312.pyc -----

----- Contents of BotBoundary.cpython-312.pyc -----

----- Contents of BrowserBoundary.cpython-312.pyc -----

----- Contents of ChatBoundary.cpython-312.pyc -----

----- Contents of CloseBrowserBoundary.cpython-312.pyc -----

----- Contents of HelpBoundary.cpython-312.pyc -----

----- Contents of LoginBoundary.cpython-312.pyc -----

----- Contents of MonitorPriceBoundary.cpython-312.pyc -----

----- Contents of NavigationBoundary.cpython-312.pyc -----

----- Contents of PriceBoundary.cpython-312.pyc -----

----- Contents of StopBoundary.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountControl.py -----

```
from entity.AccountEntity import AccountEntity

class AccountControl:
    def __init__(self):
        self.account_entity = AccountEntity()

    def add_account(self, username, password, webSite):
        self.account_entity.connect()
```

```
        self.account_entity.add_account(username, password, webSite)
        self.account_entity.close()

    def fetch_accounts(self):
        """Fetch all accounts and return them."""
        self.account_entity.connect()
        accounts = self.account_entity.fetch_accounts()

        if accounts:
            account_messages = []
            for account in accounts:
                message = f"ID: {account[0]}, Username: {account[1]}, Password: {account[2]}, Website:
{account[3]}"
                print(message)  # For terminal output
                account_messages.append(message)
            self.account_entity.close()
            return account_messages
        else:
            print("No accounts found.")  # For terminal output
            self.account_entity.close()
            return ["No accounts found."]


    def fetch_account_by_website(self, website):
            """Fetch the username and password where the website matches."""
            self.account_entity.connect()
            account = self.account_entity.fetch_account_by_website(website)  # Call the entity method
```

```
            self.account_entity.close()
            return account


    def delete_account(self, account_id):
        self.account_entity.connect()
        self.account_entity.delete_account(account_id)
        self.account_entity.reset_id_sequence()
        self.account_entity.close()
```

----- Contents of AvailabilityControl.py -----

----- Contents of BotControl.py -----

```
import asyncio

class BotControl:
    def __init__(self, bot):
        self.bot = bot

    async def send_greeting(self):
        """Sends a greeting when the bot comes online."""
        channel = self.bot.get_channel(self.bot.config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online! type '!project_help' to see what I can do")
```

```
    async def stop_bot(self):
        """Stops the bot gracefully, ensuring all connections are closed."""
        print("Bot is stopping...")

        await self.bot.close()
```

----- Contents of BrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl  # Use AccountControl for consistency

class BrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()  # Use AccountControl to fetch credentials

    def launch_browser(self, user, incognito=False):
        return self.browser_entity.launch_browser(incognito=incognito, user=user)
```

----- Contents of ChatControl.py -----

```
# ChatControl in control/ChatControl.py
class ChatControl:
    def process_non_prefixed_message(self, message):
        """Process non-prefixed messages like 'hi', 'hello'."""
        if message.lower() in ["hi", "hello"]:
            return "Hello! How can I assist you today? Type !project_help for assistance."
```

```
        else:
            return "I didn't recognize that. Type !project_help to see available commands."

    def handle_unrecognized_command(self):
        """Handle unrecognized command from on_command_error."""
        return "I didn't recognize that command. Type !project_help for assistance."
```

----- Contents of CloseBrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity

class CloseBrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def close_browser(self):
        return self.browser_entity.close_browser()
```

----- Contents of DataExtractionControl.py -----

----- Contents of HelpControl.py -----

```
class HelpControl:
    def get_help_message(self):
        """Returns a list of available bot commands."""
        return (
            "Here are the available commands:\n"
```

```
            "!project_help - Get help on available commands.\n"
            "!chat_with_bot - Say hi to the bot.\n"
            "!login_to_website - Log in to a website.\n"
            "!launch_browser - Launch the browser.\n"
            "!close_browser - Close the browser.\n"
            "!navigate_to_website - Navigate to a website.\n"
            "!track_price - Track a product price.\n"
            "!check_price - Check the price of a product.\n"
            "!check_availability - Check the availability of a product.\n"
            "!stop_tracking - Stop tracking a product.\n"
            "!receive_notifications - Receive notifications for price changes.\n"
            "!extract_data - Export data to Excel or HTML.\n"
            "!stop - Stop the bot.\n"
        )



----- Contents of LoginControl.py -----
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl

class LoginControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()

    async def login(self, site, incognito=False, retries=1):
        # Fetch credentials using AccountControl
```

```
        account = self.account_control.fetch_account_by_website(site)
        if account:
            username, password = account
            return await self.browser_entity.login(site, username, password, incognito, retries)
        else:
            return f"No account found for website {site}"



----- Contents of MonitorPriceControl.py -----
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class MonitorPriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("MonitorPriceControl")

    async def monitor_price(self, ctx, url, frequency=1):
        """Monitor the price at a given interval."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                await ctx.send(f"Monitoring price every {frequency} minute(s).")
                previous_price = None

                while True:
```

```
                current_price = self.price_entity.get_price(url)
                if current_price:
                    if previous_price is None:
                        await ctx.send(f"Starting price monitoring. Current price is: {current_price}")
                    else:
                        if current_price > previous_price:
                                await ctx.send(f"Price went up! Current price: {current_price} (Previous:
{previous_price})")
                        elif current_price < previous_price:
                                await ctx.send(f"Price went down! Current price: {current_price} (Previous:
{previous_price})")
                        else:
                            await ctx.send(f"Price remains the same: {current_price}")
                    previous_price = current_price
                else:
                    await ctx.send("Failed to retrieve the price.")
                await asyncio.sleep(frequency * 60)  # Wait for the next check
            except Exception as e:
                self.logger.error(f"Failed to monitor price for {url}: {e}")
                await ctx.send(f"Failed to monitor price: {e}")
        else:
            await ctx.send("This command can only be used in the designated channel.")
```

----- Contents of NavigationControl.py -----

```
from entity.BrowserEntity import BrowserEntity
```

```
class NavigationControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def navigate_to_url(self, url):
        """Navigate to a specific URL."""
        return self.browser_entity.navigate_to_url(url)
```

----- Contents of NotificationControl.py -----

----- Contents of PriceControl.py -----

```
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class PriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("PriceControl")

    async def get_price(self, ctx, url):
        """Fetch the current price from the given URL."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
```

```
                price = self.price_entity.get_price(url)
                if price:
                    return f"The current price is: {price}"
                else:
                    return "Failed to retrieve the price."
            except Exception as e:
                self.logger.error(f"Failed to get price for {url}: {e}")
                return f"Error getting price: {e}"
        else:
            return "This command can only be used in the designated channel."
```

----- Contents of StopMonitoringControl.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of AccountControl.cpython-312.pyc -----

----- Contents of BotControl.cpython-312.pyc -----

----- Contents of BrowserControl.cpython-312.pyc -----

----- Contents of ChatControl.cpython-312.pyc -----

----- Contents of CloseBrowserControl.cpython-312.pyc -----

----- Contents of HelpControl.cpython-312.pyc -----

----- Contents of LoginControl.cpython-312.pyc -----

----- Contents of MonitorPriceControl.cpython-312.pyc -----

----- Contents of NavigationControl.cpython-312.pyc -----

----- Contents of PriceControl.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountEntity.py -----

```
import psycopg2
from Config import Config


class AccountEntity:
    def __init__(self):
        self.dbname = "postgres"
        self.user = "postgres"
        self.host = "localhost"
        self.port = "5432"
        self.password = Config.DATABASE_PASSWORD


    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
                password=self.password,
                host=self.host,
                port=self.port
            )
            self.cursor = self.connection.cursor()
            print("Database Connection Established.")
        except Exception as error:
            print(f"Error connecting to the database: {error}")
```

```
            self.connection = None
            self.cursor = None

    def add_account(self, username, password, webSite):
        """Insert a new account into the accounts table."""
        try:
            if self.cursor:
                    self.cursor.execute("INSERT INTO accounts (username, password, website) VALUES
(%s, %s, %s)", (username, password, webSite))
                self.connection.commit()
                print(f"Account {username} added successfully.")
        except Exception as error:
            print(f"Error inserting account: {error}")

    def fetch_accounts(self):
        """Fetch all accounts from the accounts table."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts;")
                accounts = self.cursor.fetchall()
                return accounts
        except Exception as error:
            print(f"Error fetching accounts: {error}")
            return None

    def delete_account(self, account_id):
        """Delete an account by ID."""
```

```
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts WHERE id = %s", (account_id,))
                account = self.cursor.fetchone()
                if account:
                    self.cursor.execute("DELETE FROM accounts WHERE id = %s", (account_id,))
                    self.connection.commit()
                    print(f"Account with ID {account_id} deleted successfully.")
                else:
                    print(f"Account with ID {account_id} not found. No deletion performed.")
        except Exception as error:
            print(f"Error deleting account: {error}")



    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        try:
                        self.cursor.execute("SELECT username, password FROM accounts WHERE
LOWER(website) = LOWER(%s)", (website,))
            return self.cursor.fetchone()  # Returns one matching account
        except Exception as error:
            print(f"Error fetching account for website {website}: {error}")
            return None




    def reset_id_sequence(self):
```

```
        """Reset the account ID sequence to the next available value."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT COALESCE(MAX(id), 0) + 1 FROM accounts")
                next_id = self.cursor.fetchone()[0]
                        self.cursor.execute("ALTER SEQUENCE accounts_id_seq RESTART WITH %s",
(next_id,))
                self.connection.commit()
                print(f"ID sequence reset to {next_id}.")
        except Exception as error:
            print(f"Error resetting ID sequence: {error}")

    def close(self):
        """Close the database connection."""
        if self.cursor:
            self.cursor.close()
        if self.connection:
            self.connection.close()
            print("Database Connection closed.")



----- Contents of BrowserEntity.py -----
import asyncio
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
```

```
from selenium.webdriver.support import expected_conditions as EC
from utils.css_selectors import Selectors  # Import CSS selectors for the website

class BrowserEntity:
    _instance = None  # Singleton instance

    def __new__(cls, *args, **kwargs):
        if cls._instance is None:
            cls._instance = super(BrowserEntity, cls).__new__(cls)
            cls._instance.driver = None  # Initialize driver to None
        return cls._instance

    def launch_browser(self, incognito=False, user=None):
        if self.driver:
            print("Browser is already running. No need to launch a new one.")
            return "Browser is already running."

        try:
            # Special launch options as per your original implementation
            options = webdriver.ChromeOptions()

            # Add options to avoid crashing and improve performance
            options.add_argument("--remote-debugging-port=9222")
            options.add_experimental_option("excludeSwitches", ["enable-automation"])
            options.add_experimental_option('useAutomationExtension', False)
            options.add_argument("--start-maximized")
            options.add_argument("--disable-notifications")
```

```
            options.add_argument("--disable-popup-blocking")
            options.add_argument("--disable-infobars")
            options.add_argument("--disable-extensions")
            options.add_argument("--disable-webgl")
            options.add_argument("--disable-webrtc")
            options.add_argument("--disable-rtc-smoothing")

            if incognito:
                options.add_argument("--incognito")

            self.driver = webdriver.Chrome(service=Service(), options=options)
            success_message = "Chrome browser launched successfully in incognito mode." if incognito
else "Chrome browser launched successfully."
            print(f"Driver initialized: {self.driver}")  # Debug: Print the driver
            return success_message
        except Exception as e:
            error_message = f"Failed to launch browser: {e}"
            print(error_message)
            raise

    def navigate_to_url(self, url):
        if not self.driver:
            print("Driver is not initialized, launching browser first.")  # Debug
            self.launch_browser()
        try:
            self.driver.get(url)
            return f"Navigated to URL: {url}"
```

```
        except Exception as e:
            raise


    def close_browser(self):
        print(f"Closing browser. Current driver: {self.driver}")  # Debug: Check the driver status
        if self.driver:
            self.driver.quit()  # Close the browser session
            self.driver = None  # Set to None after closing
            print("Browser closed successfully.")
            return "Browser closed successfully."
        else:
            print("No browser is currently open.")
            return "No browser is currently open."



    async def login(self, site, username, password, incognito=False, retries=1):
        # Get the URL and selectors from css_selectors
        url = Selectors.get_selectors_for_url(site)['url']
        for attempt in range(retries):
            try:
                self.navigate_to_url(url)
                await asyncio.sleep(3)

                # Enter the email address
                                                email_field    =    self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['email_field'])
                email_field.click()
```

```
            email_field.send_keys(username)
            await asyncio.sleep(3)

            # Enter the password
                                    password_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['password_field'])
            password_field.click()
            password_field.send_keys(password)
            await asyncio.sleep(3)

            # Click the login button
                                    sign_in_button = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['SignIn_button'])
            sign_in_button.click()
            await asyncio.sleep(5)

            # Wait for the homepage to load after login
            WebDriverWait(self.driver, 30).until(
                                            EC.presence_of_element_located((By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['homePage'])))

            return f"Logged in to {url} successfully with username: {username}"
        except Exception as e:
            if attempt < retries - 1:
                await asyncio.sleep(3)
            else:
                raise e
```

----- Contents of DateEntity.py -----

----- Contents of NotificationEntity.py -----

----- Contents of PriceEntity.py -----

```
import time
from selenium.webdriver.common.by import By
from utils.css_selectors import Selectors
from entity.BrowserEntity import BrowserEntity  # Import the browser interaction logic

class PriceEntity:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def get_price(self, url):
        """Fetch the price from the provided URL using CSS selectors."""
        selectors = Selectors.get_selectors_for_url(url)
        if not selectors:
            raise ValueError(f"No selectors found for URL: {url}")

        # Navigate to the URL using the browser entity
        self.browser_entity.navigate_to_url(url)
        time.sleep(2)  # Wait for the page to load
```

```
        try:
            # Use the CSS selector to find the price on the page
                        price_element = self.browser_entity.driver.find_element(By.CSS_SELECTOR,
selectors['price'])
            price = price_element.text
            print(f"Price found: {price}")
            return price
        except Exception as e:
            print(f"Error finding price: {e}")
            return None
```

----- Contents of PriceHistoryEntity.py -----

----- Contents of __init__.py -----

#empty init file

----- Contents of AccountEntity.cpython-312.pyc -----

----- Contents of BrowserEntity.cpython-312.pyc -----

----- Contents of PriceEntity.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of project.txt -----

DiscordBotProject_CISC699 - Project Overview

Introduction

This project is a Discord bot designed to perform various tasks, including tracking product prices, checking availability, logging into websites, and exporting data.

The bot interacts with users via commands sent through Discord and responds based on the requested use case.

The project follows a clear structure, adhering to software engineering best practices, and separates the logic into Boundary, Control, and Entity objects to manage the flow of data and logic.

Scroll all the way down for project outline

Objects and Their Roles

Entity Objects

Entity objects represent the core business data and operations related to those entities. They store data and perform business logic related to that data. They do not interact directly with the user.

ProductEntity: Represents product information such as price and features. It handles product-related data (e.g., retrieving the current price).

DateEntity: Handles date and availability logic for booking or checking availability of services.

AccountEntity: Manages user login credentials for websites like BestBuy or eBay.

TrackingHistoryEntity: Stores and tracks historical data on product prices. Helps to compare past prices with current ones.

BrowserEntity: Manages the state of the browser (e.g., if the browser is running, whether it's in incognito mode, etc.).

NotificationEntity: Handles user preferences for receiving notifications, such as when prices change or product availability is updated.

Control Objects

Control objects are responsible for handling the logic of each use case. They interact with entity objects to manage data and handle business rules. Control objects execute the steps required to fulfill a use case.

HelpControl: Provides a list of commands available to the user.

ChatControl: Handles basic user interaction, such as greetings and responses to basic phrases like "hi" or "hello."

LoginControl: Manages the process of logging into a website, including retrieving login credentials from the database and passing them to the browser.

BrowserControl: Manages the launch and setup of the browser, including handling incognito mode and configuring the browser.

CloseBrowserControl: Handles the logic for closing the browser when requested by the user.

NavigationControl: Manages the process of navigating to a specific URL in the browser.

ProductTrackingControl: Manages the tracking of a product's price over time, scheduling regular price checks.

ProductControl: Checks the current price of a product and retrieves relevant product data.

AvailabilityControl: Handles checking the availability of a product or service based on user-provided dates.

StopTrackingControl: Stops the tracking process for a product or service.

NotificationControl: Monitors for changes in tracked products and sends notifications when a price or availability change occurs.

DataExtractionControl: Manages the extraction of tracking data, exporting it to Excel or HTML files.

BotControl: Manages the overall lifecycle of the Discord bot, including starting, stopping, and managing the registration of commands.

Boundary Objects

Boundary objects serve as the bridge between the user (or external actor) and the system. They collect data from the user and forward it to the appropriate control object. Boundary objects are responsible for interacting with the actor but not for executing business logic.

HelpBoundary: Collects the user?s help request and forwards it to HelpControl.

ChatBoundary: Receives chat commands from the user and forwards them to ChatControl.

LoginBoundary: Collects login credentials from the user and forwards them to LoginControl.

BrowserBoundary: Receives commands to launch the browser and forwards them to BrowserControl.

CloseBrowserBoundary: Receives the user?s request to close the browser and forwards it to

CloseBrowserControl.

NavigationBoundary: Receives URL input from the user and forwards it to NavigationControl.

ProductTrackingBoundary: Collects the user?s request to track a product and forwards it to ProductTrackingControl.

ProductBoundary: Receives the user?s request to check a product price and forwards it to ProductControl.

AvailabilityBoundary: Collects the user?s availability check request and forwards it to AvailabilityControl.

StopTrackingBoundary: Receives the user?s request to stop tracking a product and forwards it to StopTrackingControl.

NotificationBoundary: Collects user preferences for receiving notifications and forwards them to NotificationControl.

DataExtractionBoundary: Collects the user?s request to export data and forwards it to DataExtractionControl.

StopBoundary: Receives the request to stop the bot and forwards it to BotControl.

Capabilities

Here?s what the bot can do:

1. !project_help

Description: Provides a list of available commands the user can issue.

Objects Involved:

Boundary: HelpBoundary

Control: HelpControl

Interaction: HelpBoundary collects the user?s help request and forwards it to HelpControl, which responds with the list of commands.

2. !chat_with_bot

Description: Responds to simple greetings (e.g., "hi", "hello") and provides a welcome message.

Objects Involved:

Boundary: ChatBoundary

Control: ChatControl

Interaction: ChatBoundary collects chat input and forwards it to ChatControl, which sends back a predefined response.

3. !login_to_website

Description: Logs into a website using stored credentials (e.g., BestBuy).

Objects Involved:

Boundary: LoginBoundary

Control: LoginControl, BrowserControl, NavigationControl

Entity: AccountEntity

Interaction:

LoginBoundary collects login credentials and forwards them to LoginControl.

LoginControl works with BrowserControl to launch the browser.

NavigationControl navigates to the website's login page.

AccountEntity retrieves the stored credentials from the database and logs the user in.

4. !launch_browser

Description: Launches the browser, optionally in incognito mode.

Objects Involved:

Boundary: BrowserBoundary

Control: BrowserControl

Entity: BrowserEntity

Interaction: BrowserBoundary collects the user's request to launch the browser and sends it to BrowserControl. BrowserControl uses BrowserEntity to configure and launch the browser.

5. !close_browser

Description: Closes the currently open browser session.

Objects Involved:

Boundary: CloseBrowserBoundary

Control: CloseBrowserControl

Entity: BrowserEntity

Interaction: CloseBrowserBoundary forwards the user?s request to CloseBrowserControl, which then tells BrowserEntity to close the browser session.

6. !navigate_to_website

Description: Navigates to a specific website URL in the browser.

Objects Involved:

Boundary: NavigationBoundary

Control: NavigationControl

Entity: BrowserEntity

Interaction: NavigationBoundary collects the URL input from the user and forwards it to NavigationControl. NavigationControl instructs BrowserEntity to navigate to the specified URL.

7. !track_price

Description: Tracks the price of a product over time and sends notifications if the price changes.

Objects Involved:

Boundary: ProductTrackingBoundary

Control: ProductTrackingControl, ProductControl, NotificationControl

Entity: ProductEntity, TrackingHistoryEntity, NotificationEntity

Interaction:

ProductTrackingBoundary collects the product URL from the user.

ProductTrackingControl initiates price tracking and uses ProductControl to fetch the current price.

The current price is stored in TrackingHistoryEntity.

If there?s a price change, NotificationControl sends an alert via NotificationEntity.

8. !check_price

Description: Manually checks the current price of a product.

Objects Involved:

Boundary: ProductBoundary

Control: ProductControl

Entity: ProductEntity

Interaction: ProductBoundary collects the product information from the user, and ProductControl retrieves the current price using ProductEntity.

9. !check_availability

Description: Checks the availability of a product or service on a specific date.

Objects Involved:

Boundary: AvailabilityBoundary

Control: AvailabilityControl

Entity: DateEntity

Interaction: AvailabilityBoundary collects the date and product/service details. AvailabilityControl checks the availability via DateEntity.

10. !stop_tracking

Description: Stops tracking the price or availability of a product.

Objects Involved:

Boundary: StopTrackingBoundary

Control: StopTrackingControl

Entity: TrackingHistoryEntity

Interaction: StopTrackingBoundary collects the stop request from the user. StopTrackingControl stops the tracking and updates TrackingHistoryEntity.

11. !receive_notifications

Description: Sends notifications when there?s a change in price or availability for tracked products/services.

Objects Involved:

Boundary: NotificationBoundary

Control: NotificationControl

Entity: NotificationEntity, TrackingHistoryEntity

Interaction: NotificationBoundary collects the user?s preferences for receiving notifications. NotificationControl monitors for changes and uses NotificationEntity to send alerts when changes occur.

12. !extract_data

Description: Extracts the tracked product data and exports it to Excel or HTML format.

Objects Involved:

Boundary: DataExtractionBoundary

Control: DataExtractionControl

Entity: TrackingHistoryEntity

Utilities: ExcelUtils, HTMLUtils

Interaction: DataExtractionBoundary collects the user?s request for data extraction. DataExtractionControl retrieves data from TrackingHistoryEntity and uses ExcelUtils or HTMLUtils to export the data to the desired format.

13. !stop

Description: Stops the Discord bot from running.

Objects Involved:

Boundary: StopBoundary

Control: BotControl

Interaction: StopBoundary collects the stop command from the user and forwards it to BotControl, which gracefully stops the bot.

DiscordBotProject_CISC699/

?

??? boundary/

?   ??? AccountBoundary.py

?   ??? HelpBoundary.py

?   ??? ChatBoundary.py

?   ??? LoginBoundary.py

?   ??? BrowserBoundary.py

?   ??? CloseBrowserBoundary.py

?   ??? NavigationBoundary.py

?   ??? ProductTrackingBoundary.py

?   ??? ProductBoundary.py

?   ??? AvailabilityBoundary.py

?   ??? StopTrackingBoundary.py

?   ??? NotificationBoundary.py

?   ??? DataExtractionBoundary.py

?   ??? StopBoundary.py

?

??? control/

?   ??? AccountControl.py

?   ??? HelpControl.py

?   ??? ChatControl.py

?   ??? LoginControl.py

?   ??? BrowserControl.py

?   ??? CloseBrowserControl.py

?   ??? NavigationControl.py

?   ??? ProductTrackingControl.py

?   ??? ProductControl.py

?   ??? AvailabilityControl.py

?   ??? StopTrackingControl.py

?   ??? NotificationControl.py

?   ??? DataExtractionControl.py

?   ??? BotControl.py

?

??? entity/

?   ??? ProductEntity.py

?   ??? DateEntity.py

?   ??? AccountEntity.py

?   ??? TrackingHistoryEntity.py

?   ??? BrowserEntity.py

?   ??? NotificationEntity.py

?

??? utils/

?   ??? ExcelUtils.py

?   ??? HTMLUtils.py

?   ??? DiscordUtils.py

```
?
??? test/
?   ??? test_addAccount.py
?   ??? test_deleteAccount.py
?   ??? test_fetchAccounts.py
?   ??? test_excel_creation.py
?   ??? test_html_creation.py
?
??? Config.py
??? main.py
??? project.txt



----- Contents of project_structure.py -----
import os

def list_files_and_folders(directory, output_file):
    with open(output_file, 'w') as f:
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            f.write(f"Directory: {root}\n")
            for dir_name in dirs:
                f.write(f"  Folder: {dir_name}\n")
            for file_name in files:
                f.write(f"  File: {file_name}\n")
```

```
# Update the directory path to your project folder
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699"
output_file = os.path.join(project_directory, "project_structure.txt")

# Call the function to list files and save output to .txt
list_files_and_folders(project_directory, output_file)

print(f"File structure saved to {output_file}")
```

----- Contents of project_structure.txt -----

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699

- Folder: boundary
- Folder: control
- Folder: entity
- Folder: test
- Folder: utils
- File: Config.py
- File: main.py
- File: project.txt
- File: project_structure.txt
- File: temporary.py
- File: Tests_URLs.txt

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC

699/DiscordBotProject_CISC699\boundary

File: AccountBoundary.py

File: AvailabilityBoundary.py

File: BotBoundary.py

File: BrowserBoundary.py

File: CloseBrowserBoundary.py

File: DataExtractionBoundary.py

File: HelpBoundary.py

File: LoginBoundary.py

File: NavigationBoundary.py

File: NotificationBoundary.py

File: ProductBoundary.py

File: ProductTrackingBoundary.py

File: StopBoundary.py

File: StopTrackingBoundary.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\control

File: AccountControl.py

File: AvailabilityControl.py

File: BotControl.py

File: BrowserControl.py

File: ChatControl.py

File: CloseBrowserControl.py

File: DataExtractionControl.py

File: HelpControl.py

File: LoginControl.py

File: NavigationControl.py

File: NotificationControl.py

File: ProductControl.py

File: ProductTrackingControl.py

File: StopTrackingControl.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\entity

File: AccountEntity.py

File: BrowserEntity.py

File: DateEntity.py

File: NotificationEntity.py

File: ProductEntity.py

File: TrackingHistoryEntity.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\test

File: test_addAccount.py

File: test_deleteAccount.py

File: test_excel_creation.py

File: test_fetchAccounts.py

File: test_html_creation.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\utils

File: css_selectors.py

File: DiscordUtils.py

File: ExcelUtils.py

File: HTMLUtils.py

----- Contents of temporary.py -----

```
import os

def get_all_files_text(directory, output_file):
    with open(output_file, 'w', encoding='utf-8') as outfile:
        # Traverse the directory
        for foldername, subfolders, filenames in os.walk(directory):
            for filename in filenames:
                file_path = os.path.join(foldername, filename)
                try:
                    # Open and read each file's content
                    with open(file_path, 'r', encoding='utf-8') as infile:
                        outfile.write(f"----- Contents of {filename} -----\n")
                        outfile.write(infile.read())
                        outfile.write("\n\n")
                except Exception as e:
                    print(f"Error reading {file_path}: {e}")

if __name__ == "__main__":
        directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699"
        output_file = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\all_files_content.txt"
```

get_all_files_text(directory, output_file)

----- Contents of Tests_URLs.txt -----

database password: postgres

Working Commands: Test commands

!project_help

!login bestbuy

!launch_browser

!close_browser

!navigate_to_website https://www.google.com/

!monitor_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

!check_availability https://www.opentable.com/r/bar-spero-washington/

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22"

!stop_monitoring

!stop

**********

Working on it:

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22" "8:00 PM"

**********

URLs to Test:

https://www.opentable.com/r/bar-spero-washington/

https://www.ebay.com/itm/314411766963?_trkparms=amclksrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSi

mpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f81
9a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3A
vlp_homepage

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734
&sav=true

----- Contents of test_addAccount.py -----

```
import sys, os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_add_account():
    account_control = AccountControl()

    # Adding a new account
    account_control.add_account("newUser", "newPassword123", "newWebsite")

if __name__ == "__main__":
    test_add_account()
```

----- Contents of test_deleteAccount.py -----

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
```

```
from control.AccountControl import AccountControl

def test_delete_account():
    account_control = AccountControl()

    account_control.delete_account(4)

if __name__ == "__main__":
    test_delete_account()
```

----- Contents of test_excel_creation.py -----

----- Contents of test_fetchAccounts.py -----

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_fetch_accounts():
    account_control = AccountControl()

    # Fetching all accounts
    account_control.fetch_accounts()
```

```
def test_fetch_account_by_website(website):
    account_control = AccountControl()

    # Fetch the account by website directly
    account = account_control.fetch_account_by_website(website)

    if account:
        username, password = account  # Unpack the returned tuple
        print(f"Website: {website}, Username: {username}, Password: {password}")
    else:
        print(f"No account found for website: {website}")

if __name__ == "__main__":
    test_fetch_accounts()
    test_fetch_account_by_website("ebay")
```

----- Contents of test_html_creation.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of css_selectors.py -----

```
class Selectors:
    SELECTORS = {
```

```
"trendyol": {
    "price": ".featured-prices .prc-dsc"  # Selector for Trendyol price
},
"ebay": {
    "url": "https://signin.ebay.com/signin/",
    "email_field": "#userid",
    "continue_button": "[data-testid*='signin-continue-btn']",
    "password_field": "#pass",
    "login_button": "#sgnBt",
    "price": ".x-price-primary span"  # CSS selector for Ebay price
},
"bestbuy": {
    "url": "https://www.bestbuy.com/signin/",
    "email_field": "#fld-e",
    #"continue_button": ".cia-form__controls  button",
    "password_field": "#fld-p1",
    "SignIn_button": ".cia-form__controls  button",
    "price": "[data-testid='customer-price'] span",  # CSS selector for BestBuy price
    "homePage": ".v-p-right-xxs.line-clamp"
},
"opentable": {
    "url": "https://www.opentable.com/",
    "date_field": "#restProfileSideBarDtpDayPicker-label",
    "time_field": "#restProfileSideBartimePickerDtpPicker",
    "find_table_button": ".find-table-button",  # Example selector for the Find Table button
    "availability_result": ".availability-result",  # Example selector for availability results
        "show_next_available_button": "button[data-test='multi-day-availability-button']",  # Show
```

```
next available button
            "available_dates": "ul[data-test='time-slots'] > li",  # Available dates and times
            "no_availability": "div._8ye6OVzeOuU- span"
        }
    }

    @staticmethod
    def get_selectors_for_url(url):
        for keyword, selectors in Selectors.SELECTORS.items():
            if keyword in url.lower():
                return selectors
        return None  # Return None if no matching selectors are found
```

----- Contents of DiscordUtils.py -----

----- Contents of ExcelUtils.py -----

----- Contents of HTMLUtils.py -----

----- Contents of css_selectors.cpython-312.pyc -----

----- Contents of HelpText.cpython-312.pyc -----

----- Contents of Config.cpython-312.pyc -----

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC

699\DiscordBotProject_CISC699\Config.py ---

```
class Config:
    DISCORD_TOKEN = 'MTI2OTM4MTE4OTA1NjMzNTk3Mw.Gihcfw.nrq0x-JiL65P0LIQTO-rTyyXq0qC-2PSSBuXr8'
    CHANNEL_ID = 1269383349278081054
    DATABASE_PASSWORD = 'postgres'
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\main.py ---

```
import discord
from discord.ext import commands
from boundary.BotBoundary import BotBoundary
from boundary.HelpBoundary import HelpBoundary
from boundary.AccountBoundary import AccountBoundary
from boundary.BrowserBoundary import BrowserBoundary
from boundary.LoginBoundary import LoginBoundary
from boundary.CloseBrowserBoundary import CloseBrowserBoundary
from boundary.StopBoundary import StopBoundary
from boundary.NavigationBoundary import NavigationBoundary
from boundary.PriceBoundary import PriceBoundary
from boundary.MonitorPriceBoundary import MonitorPriceBoundary
from Config import Config

# Set up the bot's intents
intents = discord.Intents.default()
intents.message_content = True  # Enable reading message content

# Initialize the bot with the correct command prefix and intents
```

```
class MyBot(commands.Bot):
    async def setup_hook(self):
        await self.add_cog(BotBoundary(self))
        await self.add_cog(HelpBoundary(self))
        await self.add_cog(AccountBoundary(self))
        await self.add_cog(BrowserBoundary(self))
        await self.add_cog(StopBoundary(self))
        await self.add_cog(LoginBoundary(self))
        await self.add_cog(CloseBrowserBoundary(self))
        await self.add_cog(NavigationBoundary(self))
        await self.add_cog(PriceBoundary(self))
        await self.add_cog(MonitorPriceBoundary(self))

# Run the bot
if __name__ == "__main__":
    bot = MyBot(command_prefix="!", intents=intents)
    print(f"Bot is starting...")
    bot.run(Config.DISCORD_TOKEN)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\project_summary.txt ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\all_files_content.txt ---

----- Contents of all_files_content.txt -----

----- Contents of Config.py -----

```
class Config:
                                                    DISCORD_TOKEN                    =
'MTI2OTM4MTE4OTA1NjMzNTk3Mw.Gihcfw.nrq0x-JiL65P0LIQTO-rTyyXq0qC-2PSSBuXr8'
    CHANNEL_ID = 1269383349278081054
    DATABASE_PASSWORD = 'postgres'
```

----- Contents of main.py -----

```
import discord
from discord.ext import commands
from boundary.BotBoundary import BotBoundary
from boundary.HelpBoundary import HelpBoundary
from boundary.AccountBoundary import AccountBoundary
from boundary.BrowserBoundary import BrowserBoundary
from boundary.LoginBoundary import LoginBoundary
from boundary.CloseBrowserBoundary import CloseBrowserBoundary
from boundary.StopBoundary import StopBoundary
from boundary.NavigationBoundary import NavigationBoundary
from boundary.PriceBoundary import PriceBoundary
from boundary.MonitorPriceBoundary import MonitorPriceBoundary
from Config import Config

# Set up the bot's intents
intents = discord.Intents.default()
intents.message_content = True  # Enable reading message content

# Initialize the bot with the correct command prefix and intents
```

```
class MyBot(commands.Bot):
    async def setup_hook(self):
        await self.add_cog(BotBoundary(self))
        await self.add_cog(HelpBoundary(self))
        await self.add_cog(AccountBoundary(self))
        await self.add_cog(BrowserBoundary(self))
        await self.add_cog(StopBoundary(self))
        await self.add_cog(LoginBoundary(self))
        await self.add_cog(CloseBrowserBoundary(self))
        await self.add_cog(NavigationBoundary(self))
        await self.add_cog(PriceBoundary(self))
        await self.add_cog(MonitorPriceBoundary(self))

# Run the bot
if __name__ == "__main__":
    bot = MyBot(command_prefix="!", intents=intents)
    print(f"Bot is starting...")
    bot.run(Config.DISCORD_TOKEN)
```

----- Contents of project_files_text.pdf -----

----- Contents of COMMIT_EDITMSG -----

going succesful with the login and getting prices and everything

----- Contents of config -----

```
[core]
```

```
	repositoryformatversion = 0
	filemode = false
	bare = false
	logallrefupdates = true
	symlinks = false
	ignorecase = true
[remote "origin"]
	url = https://github.com/oguzky7/DiscordBotProject_CISC699.git
	fetch = +refs/heads/*:refs/remotes/origin/*
[branch "develop"]
	remote = origin
	merge = refs/heads/develop
	vscode-merge-base = origin/develop
[branch "StartOverAgain"]
	remote = origin
	merge = refs/heads/StartOverAgain
	vscode-merge-base = origin/develop
[branch "StartOver"]
	remote = origin
	merge = refs/heads/StartOver
	vscode-merge-base = origin/develop
```

```
----- Contents of description -----
Unnamed repository; edit this file 'description' to name the repository.
```

----- Contents of FETCH_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69 branch 'develop' of https://github.com/oguzky7/DiscordBotProject_CISC699

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b not-for-merge branch 'StartOver' of https://github.com/oguzky7/DiscordBotProject_CISC699

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 not-for-merge branch 'StartOverAgain' of https://github.com/oguzky7/DiscordBotProject_CISC699

e6f9da804a74f224a4f50336480f1896b3142fde not-for-merge branch 'main' of https://github.com/oguzky7/DiscordBotProject_CISC699

----- Contents of HEAD -----

ref: refs/heads/StartOverAgain

----- Contents of index -----

----- Contents of ORIG_HEAD -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of packed-refs -----

# pack-refs with: peeled fully-peeled sorted

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b refs/remotes/origin/StartOver

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 refs/remotes/origin/StartOverAgain

dad457b1e774179c151ce94c80cca9d1605d2b69 refs/remotes/origin/develop

e6f9da804a74f224a4f50336480f1896b3142fde refs/remotes/origin/main

----- Contents of applypatch-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.  The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:
```

----- Contents of commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to check the commit log message.
# Called by "git commit" with one argument, the name of the file
# that has the commit message.  The hook should exit with non-zero
# status after issuing an appropriate message if it wants to stop the
# commit.  The hook is allowed to edit the commit message file.
```

```
#
# To enable this hook, rename this file to "commit-msg".

# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"

# This example catches duplicate Signed-off-by lines.

test "" = "$(grep '^Signed-off-by: ' "$1" |
  sort | uniq -c | sed -e '/^[  ]*1[  ]/d')" || {
 echo >&2 Duplicate Signed-off-by lines.
 exit 1
}
```

----- Contents of fsmonitor-watchman.sample -----

```
#!/usr/bin/perl

use strict;
use warnings;
use IPC::Open2;

# An example hook script to integrate Watchman
```

```
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 2) and last update token
# formatted as a string and outputs to stdout a new update token and
# all files that have been modified since the update token. Paths must
# be relative to the root of the working tree and separated by a single NUL.
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $last_update_token) = @ARGV;

# Uncomment for debugging
# print STDERR "$0 $version $last_update_token\n";

# Check the hook interface version
if ($version ne 2) {
 die "Unsupported query-fsmonitor hook version '$version'.\n" .
     "Falling back to scanning...\n";
}

my $git_work_tree = get_working_dir();

my $retry = 1;

my $json_pkg;
```

```
eval {
 require JSON::XS;
 $json_pkg = "JSON::XS";
 1;
} or do {
 require JSON::PP;
 $json_pkg = "JSON::PP";
};

launch_watchman();

sub launch_watchman {
 my $o = watchman_query();
 if (is_work_tree_watched($o)) {
  output_result($o->{clock}, @{$o->{files}});
 }
}

sub output_result {
 my ($clockid, @files) = @_;

 # Uncomment for debugging watchman output
 # open (my $fh, ">", ".git/watchman-output.out");
 # binmode $fh, ":utf8";
 # print $fh "$clockid\n@files\n";
 # close $fh;
```

```
 binmode STDOUT, ":utf8";
 print $clockid;
 print "\0";
 local $, = "\0";
 print @files;
}

sub watchman_clock {
 my $response = qx/watchman clock "$git_work_tree"/;
 die "Failed to get clock id on '$git_work_tree'.\n" .
  "Falling back to scanning...\n" if $? != 0;

 return $json_pkg->new->utf8->decode($response);
}

sub watchman_query {
 my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
 or die "open2() failed: $!\n" .
 "Falling back to scanning...\n";

 # In the query expression below we're asking for names of files that
 # changed since $last_update_token but not from the .git folder.
 #
 # To accomplish this, we're using the "since" generator to use the
 # recency index to select candidate nodes and "fields" to limit the
 # output to file names only. Then we're using the "expression" term to
 # further constrain the results.
```

```
my $last_update_line = "";
if (substr($last_update_token, 0, 1) eq "c") {
 $last_update_token = "\"$last_update_token\"";
 $last_update_line = qq[\n"since": $last_update_token,];
}
my $query = <<" END";
 ["query", "$git_work_tree", {$last_update_line
  "fields": ["name"],
  "expression": ["not", ["dirname", ".git"]]
 }]
END

# Uncomment for debugging the watchman query
# open (my $fh, ">", ".git/watchman-query.json");
# print $fh $query;
# close $fh;

print CHLD_IN $query;
close CHLD_IN;
my $response = do {local $/; <CHLD_OUT>};

# Uncomment for debugging the watch response
# open ($fh, ">", ".git/watchman-response.json");
# print $fh $response;
# close $fh;

die "Watchman: command returned no output.\n" .
```

```
 "Falling back to scanning...\n" if $response eq "";
 die "Watchman: command returned invalid output: $response\n" .
 "Falling back to scanning...\n" unless $response =~ /^\{/;

 return $json_pkg->new->utf8->decode($response);
}

sub is_work_tree_watched {
 my ($output) = @_;
 my $error = $output->{error};
 if ($retry > 0 and $error and $error =~ m/unable to resolve root .* directory (.*) is not watched/) {
  $retry--;
  my $response = qx/watchman watch "$git_work_tree"/;
  die "Failed to make watchman watch '$git_work_tree'.\n" .
      "Falling back to scanning...\n" if $? != 0;
  $output = $json_pkg->new->utf8->decode($response);
  $error = $output->{error};
  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  # Uncomment for debugging watchman output
  # open (my $fh, ">", ".git/watchman-output.out");
  # close $fh;

  # Watchman will always return all files on the first query so
  # return the fast "everything is dirty" flag to git and do the
  # Watchman query just to get it over with now so we won't pay
```

```
  # the cost in git to look up each individual file.
  my $o = watchman_clock();
  $error = $output->{error};

  die "Watchman: $error.\n" .
  "Falling back to scanning...\n" if $error;

  output_result($o->{clock}, ("/"));
  $last_update_token = $o->{clock};

  eval { launch_watchman() };
  return 0;
 }

 die "Watchman: $error.\n" .
 "Falling back to scanning...\n" if $error;

 return 1;
}

sub get_working_dir {
 my $working_dir;
 if ($^O =~ 'msys' || $^O =~ 'cygwin') {
  $working_dir = Win32::GetCwd();
  $working_dir =~ tr/\\/\//;
 } else {
  require Cwd;
```

```
  $working_dir = Cwd::cwd();
 }

 return $working_dir;
}
```

----- Contents of post-update.sample -----

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

----- Contents of pre-applypatch.sample -----

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.
#
```

```
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:



----- Contents of pre-commit.sample -----
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
 against=HEAD
else
 # Initial commit: diff against an empty tree object
 against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
```

```
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
if [ "$allownonascii" != "true" ] &&
 # Note that the use of brackets around a tr range is ok here, (it's
 # even required, for portability to Solaris 10's /usr/bin/tr), since
 # the square bracket bytes happen to fall in the designated range.
 test $(git diff --cached --name-only --diff-filter=A -z $against |
   LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
then
 cat <<\EOF
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.

To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:

  git config hooks.allownonascii true
EOF
 exit 1
```

```
fi

# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --



----- Contents of pre-merge-commit.sample -----
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
        exec "$GIT_DIR/hooks/pre-commit"
:



----- Contents of pre-push.sample -----
#!/bin/sh

# An example hook script to verify what is about to be pushed.  Called by "git
# push" after it has checked the remote status, but before anything has been
```

```
# pushed.  If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
#   <local ref> <local oid> <remote ref> <remote oid>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')

while read local_ref local_oid remote_ref remote_oid
do
 if test "$local_oid" = "$zero"
 then
  # Handle delete
```

```
  :
 else
  if test "$remote_oid" = "$zero"
  then
   # New branch, examine all commits
   range="$local_oid"
  else
   # Update to existing branch, examine new commits
   range="$remote_oid..$local_oid"
  fi

  # Check for WIP commit
  commit=$(git rev-list -n 1 --grep '^WIP' "$range")
  if test -n "$commit"
  then
   echo >&2 "Found WIP commit in $local_ref, not pushing"
   exit 1
  fi
 fi
done

exit 0
```

----- Contents of pre-rebase.sample -----

```
#!/bin/sh
#
```

```
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.

publish=next
basebranch="$1"
if test "$#" = 2
then
 topic="refs/heads/$2"
else
 topic=`git symbolic-ref HEAD` ||
 exit 0 ;# we do not interrupt rebasing detached HEAD
fi

case "$topic" in
refs/heads/??/*)
```

```
 ;;
*)
 exit 0 ;# we do not interrupt others.
 ;;
esac

# Now we are dealing with a topic branch being rebased
# on top of master.  Is it OK to rebase it?

# Does the topic really exist?
git show-ref -q "$topic" || {
 echo >&2 "No such branch $topic"
 exit 1
}

# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
 echo >&2 "$topic is fully merged to master; better remove it."
 exit 1 ;# we could allow it, but there is no point.
fi

# Is topic ever merged to next?  If so you should not be rebasing it.
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
only_next_2=`git rev-list ^master           ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
```

```
then
 not_in_topic=`git rev-list "^$topic" master`
 if test -z "$not_in_topic"
 then
  echo >&2 "$topic is already up to date with master"
  exit 1 ;# we could allow it, but there is no point.
 else
  exit 0
 fi
else
 not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
 /usr/bin/perl -e '
  my $topic = $ARGV[0];
  my $msg = "* $topic has commits already merged to public branch:\n";
  my (%not_in_next) = map {
   /^([0-9a-f]+) /;
   ($1 => 1);
  } split(/\n/, $ARGV[1]);
  for my $elem (map {
    /^([0-9a-f]+) (.*)$/;
    [$1 => $2];
   } split(/\n/, $ARGV[2])) {
   if (!exists $not_in_next{$elem->[0]}) {
    if ($msg) {
     print STDERR $msg;
     undef $msg;
    }
```

```
    print STDERR " $elem->[1]\n";
   }
  }
 ' "$topic" "$not_in_next" "$not_in_master"
 exit 1
fi

<<\DOC_END

This sample hook safeguards topic branches that have been
published from being rewound.

The workflow assumed here is:

 * Once a topic branch forks from "master", "master" is never
   merged into it again (either directly or indirectly).

 * Once a topic branch is fully cooked and merged into "master",
   it is deleted.  If you need to build on top of it to correct
   earlier mistakes, a new topic branch is created by forking at
   the tip of the "master".  This is not strictly necessary, but
   it makes it easier to keep your history simple.

 * Whenever you need to test or publish your changes to topic
   branches, merge them into "next" branch.

The script, being an example, hardcodes the publish branch name
```

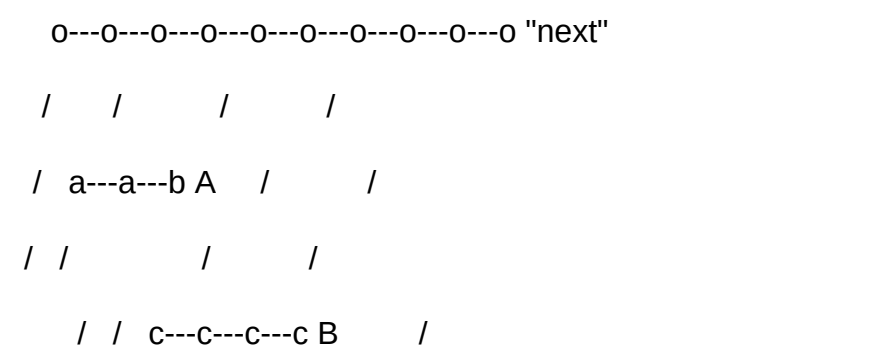
to be "next", but it is trivial to make it configurable via
$GIT_DIR/config mechanism.

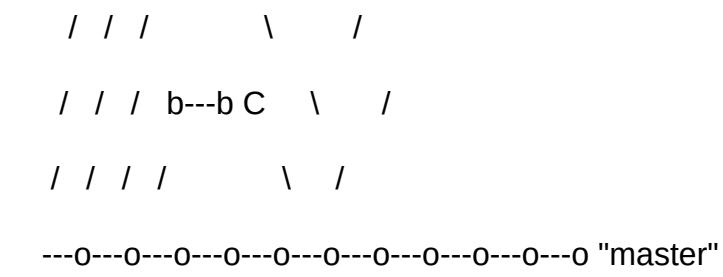
With this workflow, you would want to know:

(1) ... if a topic branch has ever been merged to "next". Young
topic branches can have stupid mistakes you would rather
clean up before publishing, and things that have not been
merged into other branches can be easily rebased without
affecting other people. But once it is published, you would
not want to rewind it.

(2) ... if a topic branch has been fully merged to "master".
Then you can delete it. More importantly, you should not
build on top of it -- other people may already want to
change things related to the topic as patches against your
"master", so if you need further changes, it is better to
fork the topic (perhaps with the same name) afresh from the
tip of "master".

Let's look at this example:






A, B and C are topic branches.

* A has one fix since it was merged up to "next".

* B has finished. It has been fully merged up to "master" and "next",
and is ready to be deleted.

* C has not merged to "next" at all.

We would want to allow C to be rebased, refuse A, and encourage
B to be deleted.

To compute (1):

```
git rev-list ^master ^topic next
git rev-list ^master        next
```

if these match, topic has not merged in next at all.

To compute (2):

```
 git rev-list master..topic

 if this is empty, it is fully merged to "master".

DOC_END



----- Contents of pre-receive.sample -----
#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"
then
 i=0
 while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
 do
  eval "value=\$GIT_PUSH_OPTION_$i"
  case "$value" in
  echoback=*)
   echo "echo from the pre-receive-hook: ${value#*=}" >&2
   ;;
  reject)
```

```
   exit 1
  esac
  i=$((i + 1))
 done
fi
```

----- Contents of prepare-commit-msg.sample -----

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source.  The hook's purpose is to edit the commit
# message file.  If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
# into the message, just before the "git status" output.  It is
# commented because it doesn't cope with --amend or with squashed
# commits.
#
```

```
# The third example adds a Signed-off-by line to the message, that can
# still be edited.  This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3

/usr/bin/perl -i.bak -ne 'print unless(m/^. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FILE"

# case "$COMMIT_SOURCE,$SHA1" in
#  ,|template,)
#    /usr/bin/perl -i.bak -pe '
#       print "\n" . `git diff --cached --name-status -r`
#   if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
#  *) ;;
# esac

# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
#   /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
# fi
```

----- Contents of push-to-checkout.sample -----

```
#!/bin/sh

# An example hook script to update a checked-out tree on a git push.
#
# This hook is invoked by git-receive-pack(1) when it reacts to git
# push and updates reference(s) in its repository, and when the push
# tries to update the branch that is currently checked out and the
# receive.denyCurrentBranch configuration variable is set to
# updateInstead.
#
# By default, such a push is refused if the working tree and the index
# of the remote repository has any difference from the currently
# checked out commit; when both the working tree and the index match
# the current commit, they are updated to match the newly pushed tip
# of the branch. This hook is to be used to override the default
# behaviour; however the code below reimplements the default behaviour
# as a starting point for convenient modification.
#
# The hook receives the commit with which the tip of the current
# branch is going to be updated:
commit=$1

# It can exit with a non-zero status to refuse the push (when it does
# so, it must not modify the index or the working tree).
die () {
 echo >&2 "$*"
 exit 1
```

```
}

# Or it can make any necessary changes to the working tree and to the
# index to bring them to the desired state when the tip of the current
# branch is updated to the new commit, and exit with a zero status.
#
# For example, the hook can simply run git read-tree -u -m HEAD "$1"
# in order to emulate git fetch that is run in the reverse direction
# with git push, as the two-tree form of git read-tree -u -m is
# essentially the same as git switch or git checkout that switches
# branches while keeping the local changes in the working tree that do
# not interfere with the difference between the branches.

# The below is a more-or-less exact translation to shell of the C code
# for the default behaviour for git's push-to-checkout hook defined in
# the push_to_deploy() function in builtin/receive-pack.c.
#
# Note that the hook will be executed from the repository directory,
# not from the working tree, so if you want to perform operations on
# the working tree, you will have to adapt your code accordingly, e.g.
# by adding "cd .." or using relative paths.

if ! git update-index -q --ignore-submodules --refresh
then
 die "Up-to-date check failed"
fi
```

```
if ! git diff-files --quiet --ignore-submodules --
then
 die "Working directory has unstaged changes"
fi

# This is a rough translation of:
#
#   head_has_history() ? "HEAD" : EMPTY_TREE_SHA1_HEX
if git cat-file -e HEAD 2>/dev/null
then
 head=HEAD
else
 head=$(git hash-object -t tree --stdin </dev/null)
fi

if ! git diff-index --quiet --cached --ignore-submodules $head --
then
 die "Working directory has staged changes"
fi

if ! git read-tree -u -m "$commit"
then
 die "Could not update working tree to new HEAD"
fi
```

----- Contents of update.sample -----

```
#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# ------
# hooks.allowunannotated
#   This boolean sets whether unannotated tags will be allowed into the
#   repository.  By default they won't be.
# hooks.allowdeletetag
#   This boolean sets whether deleting tags will be allowed in the
#   repository.  By default they won't be.
# hooks.allowmodifytag
#   This boolean sets whether a tag may be modified after creation. By default
#   it won't be.
# hooks.allowdeletebranch
#   This boolean sets whether deleting branches will be allowed in the
#   repository.  By default they won't be.
# hooks.denycreatebranch
#   This boolean sets whether remotely creating branches will be denied
#   in the repository.  By default this is allowed.
#


# --- Command line
```

```
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
 echo "Don't run this script from the command line." >&2
 echo " (if you want, you could supply GIT_DIR then run" >&2
 echo "  $0 <ref> <oldrev> <newrev>)" >&2
 exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
 echo "usage: $0 <ref> <oldrev> <newrev>" >&2
 exit 1
fi

# --- Config
allowunannotated=$(git config --type=bool hooks.allowunannotated)
allowdeletebranch=$(git config --type=bool hooks.allowdeletebranch)
denycreatebranch=$(git config --type=bool hooks.denycreatebranch)
allowdeletetag=$(git config --type=bool hooks.allowdeletetag)
allowmodifytag=$(git config --type=bool hooks.allowmodifytag)

# check for no description
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
case "$projectdesc" in
```

```
"Unnamed repository"* | "")
 echo "*** Project description file hasn't been set" >&2
 exit 1
 ;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero=$(git hash-object --stdin </dev/null | tr '[0-9a-f]' '0')
if [ "$newrev" = "$zero" ]; then
 newrev_type=delete
else
 newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
 refs/tags/*,commit)
  # un-annotated tag
  short_refname=${refname##refs/tags/}
  if [ "$allowunannotated" != "true" ]; then
   echo "*** The un-annotated tag, $short_refname, is not allowed in this repository" >&2
   echo "*** Use 'git tag [ -a | -s ]' for tags you want to propagate." >&2
   exit 1
  fi
  ;;
 refs/tags/*,delete)
  # delete tag
```

```
 if [ "$allowdeletetag" != "true" ]; then
  echo "*** Deleting a tag is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/tags/*,tag)
 # annotated tag
 if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
 then
  echo "*** Tag '$refname' already exists." >&2
  echo "*** Modifying a tag is not allowed in this repository." >&2
  exit 1
 fi
 ;;
refs/heads/*,commit)
 # branch
 if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
  echo "*** Creating a branch is not allowed in this repository" >&2
  exit 1
 fi
 ;;
refs/heads/*,delete)
 # delete branch
 if [ "$allowdeletebranch" != "true" ]; then
  echo "*** Deleting a branch is not allowed in this repository" >&2
  exit 1
 fi
```

```
  ;;
 refs/remotes/*,commit)
  # tracking branch
  ;;
 refs/remotes/*,delete)
  # delete tracking branch
  if [ "$allowdeletebranch" != "true" ]; then
   echo "*** Deleting a tracking branch is not allowed in this repository" >&2
   exit 1
  fi
  ;;
 *)
  # Anything else (is there anything else?)
  echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
  exit 1
  ;;
esac

# --- Finished
exit 0



----- Contents of exclude -----
# git ls-files --others --exclude-from=.git/info/exclude
# Lines that start with '#' are comments.
# For a project mostly in C, the following would be a good set of
# exclude patterns (uncomment them if you want to use them):
```

# *.[oa]

# *~

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

dad457b1e774179c151ce94c80cca9d1605d2b69 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 checkout: moving from develop to StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 checkout: moving from StartOverAgain to StartOver

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654644 -0500 checkout: moving from StartOver to StartOverAgain

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08 oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42 oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of develop -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOver -----

0000000000000000000000000000000000000000 6c001b5a1e047dbf0aba886623ea6f139d4f2f0b oguzky7 <oguzky7@gmail.com> 1725654594 -0500 branch: Created from refs/remotes/origin/StartOver

----- Contents of StartOverAgain -----

0000000000000000000000000000000000000000 47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 oguzky7 <oguzky7@gmail.com> 1725313551 -0500 branch: Created from refs/remotes/origin/StartOverAgain

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d oguzky7 <oguzky7@gmail.com> 1725649273 -0500 commit: new outline, starting over again

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681 oguzky7 <oguzky7@gmail.com> 1725652041 -0500 commit: started with sql connection. so far works, everything else is empty

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9 oguzky7 <oguzky7@gmail.com> 1725653627 -0500 commit: database connection for bce pattern

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 oguzky7 <oguzky7@gmail.com> 1725654565 -0500 commit: working on bot connection, need to swtich branches before

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483 oguzky7 <oguzky7@gmail.com> 1725658342 -0500 commit: bot connection and database works

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec oguzky7 <oguzky7@gmail.com> 1725664581 -0500 commit: wroking on database connection in discord

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed oguzky7 <oguzky7@gmail.com> 1725666445 -0500 commit: bot close problem

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723080 -0500 commit: add new column to database

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725776 -0500 commit: chatboundary deleted

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731940 -0500 commit: getting there

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732307 -0500 commit: login bestbuy works

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740310 -0500 commit: going succesful with the login and getting prices and everything

----- Contents of HEAD -----

0000000000000000000000000000000000000000

dad457b1e774179c151ce94c80cca9d1605d2b69 oguzky7 <oguzky7@gmail.com> 1725313447 -0500 clone: from https://github.com/oguzky7/DiscordBotProject_CISC699.git

----- Contents of StartOverAgain -----

47e31c9346cb96080b2fdf26f8781f75d8ce2cd3 b61019e219967d1a617e593c49eab9dbbc89302d

oguzky7 <oguzky7@gmail.com> 1725649275 -0500 update by push

b61019e219967d1a617e593c49eab9dbbc89302d 0c0d5a7309d480be697278ca6d58e46a30f02681

oguzky7 <oguzky7@gmail.com> 1725652043 -0500 update by push

0c0d5a7309d480be697278ca6d58e46a30f02681 217bbbe8178759121dbcda100e7c83a2c1d493e9

oguzky7 <oguzky7@gmail.com> 1725653629 -0500 update by push

217bbbe8178759121dbcda100e7c83a2c1d493e9 e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6

oguzky7 <oguzky7@gmail.com> 1725654567 -0500 update by push

e09ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 2b57837b989b94d629bfc5a49ba5fc0e8261e483

oguzky7 <oguzky7@gmail.com> 1725658344 -0500 update by push

2b57837b989b94d629bfc5a49ba5fc0e8261e483 f4db22b05f99c92e086de39ff9302bc5bd942eec

oguzky7 <oguzky7@gmail.com> 1725664583 -0500 update by push

f4db22b05f99c92e086de39ff9302bc5bd942eec 2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed

oguzky7 <oguzky7@gmail.com> 1725666447 -0500 update by push

2f7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed a0a99ac7ab128d7cb26d358957b6a913cb17cdce

oguzky7 <oguzky7@gmail.com> 1725723082 -0500 update by push

a0a99ac7ab128d7cb26d358957b6a913cb17cdce 97d073d6abfbf754fd97b9329f87cec83207ee08

oguzky7 <oguzky7@gmail.com> 1725725779 -0500 update by push

97d073d6abfbf754fd97b9329f87cec83207ee08 922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb

oguzky7 <oguzky7@gmail.com> 1725731942 -0500 update by push

922b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb c20f48a503d03a35ed438b638a1deb8fac419f42

oguzky7 <oguzky7@gmail.com> 1725732309 -0500 update by push

c20f48a503d03a35ed438b638a1deb8fac419f42 3fa951080c6be53c94cb18b588e50d8628833d3e

oguzky7 <oguzky7@gmail.com> 1725740312 -0500 update by push

----- Contents of 6d281c79b57abc36be3229d4bbab976d1b6bf3 -----

----- Contents of 634f7e652d3fb3b816adc257a0161cfaa59896 -----

----- Contents of d30f9eeb85ec53d1d3fe51515a9471ee79af75 -----

----- Contents of da87394b264ec22cc557849312040a03c1063e -----

----- Contents of acbb56d15479249f6c46de8e3acb0c8b20cb93 -----

----- Contents of cd8b5e55c2ae7f362dc0c8436c4b89a275da90 -----

----- Contents of 56de35ec5b020a6ed5805e2520c51bac37d017 -----

----- Contents of 80ceb22ccad5017ec7c7873c96e5203198cb7b -----

----- Contents of ae88962bd7ef5ee6892e9dbd1761ec6d56b7d1 -----

----- Contents of 0d5a7309d480be697278ca6d58e46a30f02681 -----

----- Contents of 2f7fda7f6f8ecf990caa0731e5be154b2adc39 -----

----- Contents of 670af298f99b15bd9dd09da31d72f0f1d38cdd -----

----- Contents of 707c217ce7f6abc9094c93c766f29b6eab99e4 -----

----- Contents of 73684a5a9f9b5ebf1227f6801d4e478fe16d4a -----

----- Contents of d21b42548625adec2ddceedb034d770341a8c8 -----

----- Contents of d19a6ce55b82d3b656da482d1c8e6a0b54b75b -----

----- Contents of 12668a4da67180c42fa2fab9cd1de9a032b227 -----

----- Contents of 8387cc096c39bd2853df1c025f6cda162264e7 -----

----- Contents of 7bedd012010871eb4fb46ee52ec2ec4bd1c14f -----

----- Contents of f7ffd8161a083f528f9374f3a9a394498db932 -----

----- Contents of 5f314f257340ec9bfbb711c9d13737f9423489 -----

----- Contents of 691ffa09c5106ba04323f77fd727a278646d28 -----

----- Contents of b53adee5cf0e0bccf2f8ad73d337a79dc020d4 -----

----- Contents of d862acaeafd794c8d81a637c85da6be6878e7d -----

----- Contents of 52beb42325a46c9fbd5a5e3e3fa9f54432bd3b -----

----- Contents of 5795b3882003e2fb6801781de44d8bf296680c -----

----- Contents of 740f61a43018fb8b31d987eb2c5e6c674ce9fc -----

----- Contents of 29623a9d92dc8f9e81e3e27a2deade8f63d70f -----

----- Contents of 7fa5fe2e9d0be5e56ceb6c07452184fccca8ad -----

----- Contents of e571a06a385f9280d5a100f60a2d1ffba1f221 -----

----- Contents of 08c8132fd411e30656c37d89b16b6dcd8b22f4 -----

----- Contents of 0c06257cbe7ea26c5c9ce349da25220ae8d9e2 -----

----- Contents of 0dfcd25d3efecd8aac12bcc1a41befb30deaf7 -----

----- Contents of 4dd067b87b28cc9a8cca80caac89fc53036596 -----

----- Contents of c2fabce9b84c9ba79bc2e79a1ac96cc34a1003 -----

----- Contents of 7bbbe8178759121dbcda100e7c83a2c1d493e9 -----

----- Contents of 6c22f5474f34c8623c296e4b6214d003eb3b08 -----

----- Contents of 56b260e1d2b7fd86f354e6fff1b011231cd844 -----

----- Contents of c1291d130ce613ec78e688bf51a576d5efc5c3 -----

----- Contents of e689cbc21247f718c17658183158af910d05d8 -----

----- Contents of 2e280b0131be8c9e98b7222577acd8341d8fa5 -----

----- Contents of e6f8d57f1f260304e67a6b0a0286689893b8e8 -----

----- Contents of c0afdc9c0e97448eb03cd4b9dd9ce145fd8d0b -----

----- Contents of 57837b989b94d629bfc5a49ba5fc0e8261e483 -----

----- Contents of c230c6518a1ad0586efaa11fd03274ae402a27 -----

----- Contents of 22650605791745310722ebcce1ece57872317e -----

----- Contents of 16855907d1ddccf87830cc3220e0815d2043d6 -----

----- Contents of 7e4b4a5c8584d3618c5d08efe60a9ae0dba9ed -----

----- Contents of 8be98f50f5ab158fff2c10db14f835a386fc99 -----

----- Contents of 5847fd8df25f2a9b603ed33703a21e91ba67c0 -----

----- Contents of 79b9a56dfb3a07e75198f466fceca1c3d8c1de -----

----- Contents of 7c13cbb1ada6eeb37aa304a9f8afb8744a47dc -----

----- Contents of 124da30b98b65e0df305d0900c2d1282b7d270 -----

----- Contents of 1cd1eadaa24a9a3f632873c244100ee7025794 -----

----- Contents of 36f14349b3a4410888ac1b43070bba46de9ede -----

----- Contents of 6ebe87b1695efcfc11786fb574e0f04bca1d18 -----

----- Contents of ddf2dd50c642602c7eba5268be402b32f66989 -----

----- Contents of 36742a002f99203e8212f1fba1a445a1bb03f6 -----

----- Contents of 623cd7312f1ff9c8abe8832fd36f20e6b84bcd -----

----- Contents of 7461d191d8a7a5de9217e88c6c140c95dd9bca -----

----- Contents of e326319a500dd5126f6173fdc28861abf9de10 -----

----- Contents of 167ddd3cb16efd9af1c2cd4539382ccd57e2fd -----

----- Contents of a951080c6be53c94cb18b588e50d8628833d3e -----

----- Contents of f3b3a6001ee55883e947e28ac1e2c7b08387bd -----

----- Contents of 2778616f503b20c4ebcb0ddd0acfaf1de01e39 -----

----- Contents of 2f14edff81d9dd2f05f2d2f8c7cd365d932bbc -----

----- Contents of 20fcb37bf59941b57c5a694a939d13d15b8900 -----

----- Contents of 98a6460ae86feb9f8049f407d2f6887424e6dc -----

----- Contents of 374847dfb20aff248fdcd59f87bdea40697b1d -----

----- Contents of f89b3afaa02365ae60cb072cf76abf1cf77ae9 -----

----- Contents of 81d6b895e5ee4ada44b133c81c73d79b001b04 -----

----- Contents of e8393441a7d2050cce36faa6564fc5cd9763ec -----

----- Contents of e434a795adc8737398a481f4c813fc380cebf3 -----

----- Contents of 7bdce5089b6b1b983d2a088b0b1a2bb2ebc56c -----

----- Contents of 790b96b9de32b415393a1c2aed46ffb49e6ae2 -----

----- Contents of e495a99648be872776623d90e76250b9cb227e -----

----- Contents of 92fdb8f360a7f817c203304d90f8e42bc44ba9 -----

----- Contents of 922e794ba24b02220cd9bb8547894d915a6e6c -----

----- Contents of bbfb724cd9bc1af776578e20073f1ec227dcfd -----

----- Contents of d010362b749bd8b0940c42084cc7192ed7d944 -----

----- Contents of 2e28edc4f64174222c015a31044054e9302f13 -----

----- Contents of 31e9454a465493fb8c05d28c52697757037d5e -----

----- Contents of c2bb829e59ca69e85259035c1e808b2c17b5b1 -----

----- Contents of 7a34d8216bffdd74b60e3e759e9ade11991cd4 -----

----- Contents of d705b2782872635fbf52906a037a130f467c0f -----

----- Contents of e8bfbb3ec7cce8b3945b1cd33efb7449fb8203 -----

----- Contents of 4c76280fb6b9eefe4f99a0dc7cc4719b85c93d -----

----- Contents of 7feec2f1765e89d736a9359a5c5e551b5f9c6d -----

----- Contents of d386fadadd939cc18d101127a21d961b148a2f -----

----- Contents of 23564225a6a0b77d1fd84d4c3b7e0800177c88 -----

----- Contents of a49e2dcd1c2d8824be05f5a7fa8d279c1c8be3 -----

----- Contents of 7422ef6e171a6dff0d231427b2d364d2dd5bb6 -----

----- Contents of b4cc5d5bf0c7aa29eb4ac829400843fcd51562 -----

----- Contents of 55b982b96b526b4e3db1ee1fe9375b73bba374 -----

----- Contents of ce2369a09426c22593285c67f91cfdcb5011ce -----

----- Contents of 4aa1ac0374fdefe7baf0c2ab6f1c65140b3190 -----

----- Contents of 5e2674b3cb3a1bed397efd15b859641dae7757 -----

----- Contents of 3184272755e64a6944c9869379c6f57ad9ba3b -----

----- Contents of f2c49d67157e02272e6521a0acb70994aff7a7 -----

----- Contents of d105aab298c832e87d7cd64886775dabf92683 -----

----- Contents of 36b1cc2ef835a0eb0f8dab3f76b700dee1ca9c -----

----- Contents of bb301025e8f69ddd490791a3c4b9fe54340715 -----

----- Contents of 3f46aa910385ec1c113d29d13a51b5933246f0 -----

----- Contents of 569d2ce7f8e2dbd43c62f90d8c1a4776fe3657 -----

----- Contents of 35ac6cdff50420307d1f6a29ae6fbce704431b -----

----- Contents of 49235dfb1543a8846b9a4317a4f3b2330af7f5 -----

----- Contents of e4df3e7aa9aa939067d082dc81129d8797e8b9 -----

----- Contents of 15d9428c470f960b9f68228b7e6e037d6fffdb -----

----- Contents of 55c1a5a11c29d9b7088fb5fae913582b80072e -----

----- Contents of 5963004ac6a7c723c76e303c6a933a7b60294e -----

----- Contents of 45a539b3b325f9d4078b26f7788fa7e4e20f4c -----

----- Contents of 6f2e8f648efc155b28e6a77f84c6d4a144f6f6 -----

----- Contents of aaf5c01d5dfff33e76d47f53e18763ae113efd -----

----- Contents of 870dde41518a8ea43fcc6a8b0c64fbf62f65f0 -----

----- Contents of 31e65729c1b72967bcf8d1bba9f4de309309ac -----

----- Contents of ffdef56d17933b2cfc82c5ad7b1fbe98641040 -----

----- Contents of 42bbb4484463cf439fba484bd82cda68230d00 -----

----- Contents of 93d2668edf7fa19ebd76cb7fd46619b412edb8 -----

----- Contents of 65f80d0fa9995fb5cd1505370703edba752087 -----

----- Contents of d12146fcaf19d62254656fde8a206024bd645e -----

----- Contents of fef29125046ebacc0b5bd63cdd89833f67d207 -----

----- Contents of 35fa7f7ad4a8914f3ff46466b98e724a942aad -----

----- Contents of 6cad680be43e263296434712ccfa7cb8683608 -----

----- Contents of 9c8e7e45537aec2443783b44ffe32a2a1d4038 -----

----- Contents of df0c2c3f663c5e0fdb705c6386fbac3c585e79 -----

----- Contents of df6da44831a6c19badb226138612e404e72fde -----

----- Contents of 95216415e1e4b423dbb14fc246d1c13f9af72c -----

----- Contents of 6048147042c058cc22c5089e6d17e8e82247e1 -----

----- Contents of cb233a35c89247d61009b08fdfbba29c952bb8 -----

----- Contents of a97486baf65dd5f97c2e2029485456bcd53875 -----

----- Contents of db4e4cb78b56559a72aca093e43364ab09e531 -----

----- Contents of dd929f339e858c99f485675a0cfd352b143313 -----

----- Contents of 134b6a845b8ff447308d5cfa26acf1abecff81 -----

----- Contents of a8744a97a8497b5a971774408b4f13efb2d36b -----

----- Contents of ce779286a458df17bc46ac6893fc66f279c068 -----

----- Contents of 1bfbc86a74447106a2a964f01541b0dc34c2b4 -----

----- Contents of 2b5ce3f1c5a21ad45b7767c1bd16d89cfad5fb -----

----- Contents of 2f8994666a261dade3f5a5c6edb13ce71f5f05 -----

----- Contents of 4fc156385de559702c75ad60d5a7a344792805 -----

----- Contents of a3700d5e8f8bb3918aa071f60ea5ded21f77d5 -----

----- Contents of bc23c3f2b221779c8bc2c90ffe3708d2171c48 -----

----- Contents of 04f9129475b746303fc616658fc3416237821d -----

----- Contents of d073d6abfbf754fd97b9329f87cec83207ee08 -----

----- Contents of e3c2abaabea3cafb640d47d6f9dc3030186fd2 -----

----- Contents of 547b1e262f315605e9d0b4d65983e72b19449c -----

----- Contents of f3d9725de4b2121766be9d9099dd28a04183e2 -----

----- Contents of 83d1857455ca55b2a4d164f109b372f65b04d4 -----

----- Contents of dcc0a73865688f0a6a2f34ce14f06bb20b5411 -----

----- Contents of 6062c4cd437cf0f7d9498f328b3cc7c39a7aab -----

----- Contents of 00b5e53f01be9fb9c3c8d49599d76948afc966 -----

----- Contents of 871be66b6a4a8f40fd9963b283d8369781dcc2 -----

----- Contents of a99ac7ab128d7cb26d358957b6a913cb17cdce -----

----- Contents of 37d15e04101d6d5747aa61b581a89ae2847b4d -----

----- Contents of 9c78053718a20d8cdf5995cb89f455dbd6becc -----

----- Contents of bd7d94612b16f8217360592a582f490e9d348c -----

----- Contents of 2287757c4e2c0b3c1e391536abb9d9a7b606a0 -----

----- Contents of 89cb5b54aa1d718008b7cd8d4ddc73fbff1c4d -----

----- Contents of 251901266fac8881c65424cf4c60ec63098f04 -----

----- Contents of 6c9ae2c2205d4c1d6823bb667316f36832129d -----

----- Contents of 93a5648f3b5744e544ade4c54a9b2558de33df -----

----- Contents of cf8ddc0f32beaf6deb845b818a237d4e2b12bc -----

----- Contents of 5bcb586c22d55e4d9b45010159cfbe1eb97b3e -----

----- Contents of 11500072f70a22804f86198228a9bf6f4b1723 -----

----- Contents of 212707bdf961cf6127136e77ce1663ca7fa0ac -----

----- Contents of 5a2099c5e29e57c313a0d69bf170ddc1dc7773 -----

----- Contents of 5ee0932ba1d58cdcced9dbc58b5061e0cdc525 -----

----- Contents of 5fc4dc3b398fe65a45236bb26508673b60ab0a -----

----- Contents of 3b2af53524bb1eadba3659d4a3e8e3e8cef960 -----

----- Contents of 5374ec1094c5935714866aa2a07d319cabb50d -----

----- Contents of c45b719c2f6ef47b5a79a6ddeb192ec3c43c9c -----

----- Contents of 3e220c1a0abd5c31bf03f3488ee49da7ba5791 -----

----- Contents of cd6e47d89f170e14c1b089b027c2bfa32a0514 -----

----- Contents of 1019e219967d1a617e593c49eab9dbbc89302d -----

----- Contents of 1b4632f741d513f68f5c9f2147e564f8e6f44e -----

----- Contents of 1b54931841d32971dbfd232f771191f49cea62 -----

----- Contents of 4917f7466bac6768c4b2c970163f10a3da883c -----

----- Contents of d08c9e2e5e8725a91bab9594774b6acda8c235 -----

----- Contents of 83cb74283680914278ac74988973d92220e580 -----

----- Contents of 2db1c7d8b292deaabb4f080f8e287e18c3fc38 -----

----- Contents of 3a8d669dcddfa7a3373422dc3c136dd1f70025 -----

----- Contents of 07750681d74929e692538a74117db1b1779b97 -----

----- Contents of 238ce60d725df8031818872526f17bdc42af50 -----

----- Contents of 7c546782379dc48fa0d181dcffc313b3822924 -----

----- Contents of 0f48a503d03a35ed438b638a1deb8fac419f42 -----

----- Contents of 2104def8712b4ee3d0cdd2a157cf0221c9b77b -----

----- Contents of 04dbf40d154e56070b1344ba8c3cfc5f564746 -----

----- Contents of 6b2e6e73058c710b05b2893bcf4a7e98b7bbad -----

----- Contents of b55569ac58142006b2bd8a09e4e761d57805c1 -----

----- Contents of 25470382adde44f12e3109bd5934c2eecdc918 -----

----- Contents of 4685b2534290d4b5f0cf6ca0d12fabf5d79e8b -----

----- Contents of 751a6ec93ec6beea13aff04bba59d57bf0ea88 -----

----- Contents of 580bd42a3222857aea5885bed9215a1d05b7c6 -----

----- Contents of 964410b7eebddb7e50578e6ccb13c7e8ab7f27 -----

----- Contents of 4c944d46b01fc2734fe68f52ebfde0cb7998e5 -----

----- Contents of 12fb46a5acc7b60a63fc627bdf6f2791543d58 -----

----- Contents of 3237d5a3531bff37efac2e869560a125ece585 -----

----- Contents of 5eea14070cdff5e8b36d391984776aa89aab44 -----

----- Contents of 7ae1b68f35a89244d46925d6cfebd4b74e6c32 -----

----- Contents of f6136b4665613b61ce58bf20a3ceb85d188776 -----

----- Contents of 83d94a59451c4c9a239ac1ae95b06ecb5587d1 -----

----- Contents of 73fd1a7d3b7afbd81ec494aac0cf63725c97f4 -----

----- Contents of f2528590e95cab41dc7026b769871a74e85090 -----

----- Contents of 10cd67f2809bd3968b90699d6e8e7bc7ce438e -----

----- Contents of b96482d8c2ef82f77977c56260947c27516f67 -----

----- Contents of 4b722e77016ace745c5592948cbe8e0a3cb033 -----

----- Contents of ad546333c05b7217d396c913e59ca48241fcc9 -----

----- Contents of 1fd6e881aef18349e7d110677b3472dcacf2a4 -----

----- Contents of 9ca3ab261c3ddc98e8dbfdc56ec36976cc73c6 -----

----- Contents of 41098bb377a0dfca7f862a6009ece6f6d30e4a -----

----- Contents of 78506479aeaf02315ca23e1ce3992b0af824be -----

----- Contents of bbcb0d6a2e1593e7f104095475e4d7d151d091 -----

----- Contents of 9de29bb2d1d6434b8b29ae775ad8c2e48c5391 -----

----- Contents of 09778dc77ac090a0d932499a2232daf1c94abf -----

----- Contents of 2b50c836c96adfa7f3259d99b4b4eb9f2d00e3 -----

----- Contents of 3920a9f5f3d14ccd1ce1c75d288a4e47498d64 -----

----- Contents of b48fdfb32f07250c2d917eef0d362fb6acb61a -----

----- Contents of c168dfbd74cce6f87298b1bcf0a4760533422a -----

----- Contents of 6c3d41c4698dc82ac757363e6998cba6f17c55 -----

----- Contents of b96d1f73ddca7d36a459c4fa161229ab6c68bb -----

----- Contents of db22b05f99c92e086de39ff9302bc5bd942eec -----

----- Contents of 68ca09597ba99ba5a13c4fe0f4e20ad40ea4e1 -----

----- Contents of 17e0509fe70c56442d04a203f7ccc565de0921 -----

----- Contents of 1a629ef9b315d397c4f1bf56b2bbbfc143ff49 -----

----- Contents of 872a50ce0cb03d65925b2a183db893a95ac571 -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.idx -----

----- Contents of pack-ff4a1df23482847a0b54105edb4fa880ee76c3d4.pack -----

----- Contents of develop -----

dad457b1e774179c151ce94c80cca9d1605d2b69

----- Contents of StartOver -----

6c001b5a1e047dbf0aba886623ea6f139d4f2f0b

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of HEAD -----

ref: refs/remotes/origin/develop

----- Contents of StartOverAgain -----

3fa951080c6be53c94cb18b588e50d8628833d3e

----- Contents of AccountBoundary.py -----

from discord.ext import commands

from control.AccountControl import AccountControl

class AccountBoundary(commands.Cog):

```
def __init__(self, bot):
    self.bot = bot
    self.account_control = AccountControl()



@commands.command(name='fetch_accounts')
async def fetch_accounts(self, ctx):
    """Fetch and display all accounts."""
    accounts = self.account_control.fetch_accounts()

    # Send each account or the no accounts message to Discord
    for account in accounts:
        await ctx.send(account)




@commands.command(name="add_account")
async def add_account(self, ctx, username: str, password: str):
    """Add a new user account to the database."""
    result = self.account_control.add_account(username, password)
    if result:
        await ctx.send(f"Account for {username} added successfully.")
    else:
        await ctx.send(f"Failed to add account for {username}.")
```

```
    @commands.command(name="delete_account")
    async def delete_account(self, ctx, user_id: int):
        """Delete a user account from the database."""
        result = self.account_control.delete_account(user_id)
        if result:
            await ctx.send(f"Account with ID {user_id} deleted successfully.")
        else:
            await ctx.send(f"Failed to delete account with ID {user_id}.")
```

----- Contents of AvailabilityBoundary.py -----

----- Contents of BotBoundary.py -----

```
from discord.ext import commands
from control.ChatControl import ChatControl
from Config import Config

class BotBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.chat_control = ChatControl()

    @commands.Cog.listener()
    async def on_ready(self):
        """Bot startup message when ready."""
        print(f'Logged in as {self.bot.user.name}')
```

```
        channel = self.bot.get_channel(Config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online!")

    @commands.Cog.listener()
    async def on_message(self, message):
        """Handle non-prefixed messages and command-prefixed messages."""
        if message.author == self.bot.user:
            return

        # Handle non-prefixed messages (like greetings)
        if not message.content.startswith('!'):
            response = self.chat_control.process_non_prefixed_message(message.content)
            await message.channel.send(response)

    @commands.Cog.listener()
    async def on_command_error(self, ctx, error):
        """Handle unrecognized commands."""
        if isinstance(error, commands.CommandNotFound):
            # Handle unknown command
            response = self.chat_control.handle_unrecognized_command()
            await ctx.send(response)
```

----- Contents of BrowserBoundary.py -----

```
from discord.ext import commands
from control.BrowserControl import BrowserControl
```

```
class BrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.browser_control = BrowserControl()

    @commands.command(name='launch_browser')
    async def launch_browser(self, ctx, *args):
        """Command to launch the browser."""
        incognito = "incognito" in args
        response = self.browser_control.launch_browser(ctx.author, incognito)
        await ctx.send(response)
```

----- Contents of CloseBrowserBoundary.py -----

```
from discord.ext import commands
from control.CloseBrowserControl import CloseBrowserControl

class CloseBrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.close_browser_control = CloseBrowserControl()

    @commands.command(name='close_browser')
    async def close_browser(self, ctx):
        """Command to close the browser."""
        response = self.close_browser_control.close_browser()
```

```
        await ctx.send(response)
```

----- Contents of DataExtractionBoundary.py -----

----- Contents of HelpBoundary.py -----

```
from discord.ext import commands
from control.HelpControl import HelpControl

class HelpBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.help_control = HelpControl()

    @commands.command(name='project_help')
    async def project_help(self, ctx):
        """Handles the project_help command."""
        help_message = self.help_control.get_help_message()
        await ctx.send(help_message)
```

----- Contents of LoginBoundary.py -----

```
from discord.ext import commands
from control.LoginControl import LoginControl

class LoginBoundary(commands.Cog):
```

```
    def __init__(self, bot):
        self.bot = bot
        self.login_control = LoginControl()

    @commands.command(name='login')
    async def login(self, ctx, site: str, *args):
        """Command to log into a website using stored credentials."""
        incognito = "incognito" in args
        retries = next((int(arg) for arg in args if arg.isdigit()), 1)
        response = await self.login_control.login(site, incognito, retries)
        await ctx.send(response)
```

----- Contents of MonitorPriceBoundary.py -----

```
from discord.ext import commands
from control.MonitorPriceControl import MonitorPriceControl

class MonitorPriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.monitor_price_control = MonitorPriceControl()

    @commands.command(name='monitor_price')
    async def monitor_price(self, ctx, url: str, frequency: int = 1):
        """Command to monitor the price at regular intervals."""
        await self.monitor_price_control.monitor_price(ctx, url, frequency)
```

----- Contents of NavigationBoundary.py -----

```
from discord.ext import commands
from control.NavigationControl import NavigationControl

class NavigationBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.navigation_control = NavigationControl()

    @commands.command(name='navigate_to_website')
    async def navigate_to_website(self, ctx, url: str):
        """Command to navigate to a specified URL."""
        response = self.navigation_control.navigate_to_url(url)
        await ctx.send(response)
```

----- Contents of NotificationBoundary.py -----

----- Contents of PriceBoundary.py -----

```
from discord.ext import commands
from control.PriceControl import PriceControl

class PriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
```

```
        self.price_control = PriceControl()

    @commands.command(name='get_price')
    async def get_price(self, ctx, url: str):
        """Command to get the price from the given URL."""
        response = await self.price_control.get_price(ctx, url)
        await ctx.send(response)
```

----- Contents of StopBoundary.py -----

```
from discord.ext import commands
from control.BotControl import BotControl

class StopBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.bot_control = BotControl(bot)

    @commands.command(name="stop_bot")
    async def stop_bot(self, ctx):
        """Handles the stop command and gracefully shuts down the bot."""
        await ctx.send("Stopping the bot...")
        await self.bot_control.stop_bot()
```

----- Contents of StopMonitoringBoundary.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of AccountBoundary.cpython-312.pyc -----

----- Contents of BotBoundary.cpython-312.pyc -----

----- Contents of BrowserBoundary.cpython-312.pyc -----

----- Contents of ChatBoundary.cpython-312.pyc -----

----- Contents of CloseBrowserBoundary.cpython-312.pyc -----

----- Contents of HelpBoundary.cpython-312.pyc -----

----- Contents of LoginBoundary.cpython-312.pyc -----

----- Contents of MonitorPriceBoundary.cpython-312.pyc -----

----- Contents of NavigationBoundary.cpython-312.pyc -----

----- Contents of PriceBoundary.cpython-312.pyc -----

----- Contents of StopBoundary.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountControl.py -----

```
from entity.AccountEntity import AccountEntity

class AccountControl:
    def __init__(self):
        self.account_entity = AccountEntity()

    def add_account(self, username, password, webSite):
        self.account_entity.connect()
        self.account_entity.add_account(username, password, webSite)
        self.account_entity.close()
```

```
    def fetch_accounts(self):
        """Fetch all accounts and return them."""
        self.account_entity.connect()
        accounts = self.account_entity.fetch_accounts()

        if accounts:
            account_messages = []
            for account in accounts:
                message = f"ID: {account[0]}, Username: {account[1]}, Password: {account[2]}, Website:
{account[3]}"
                print(message)  # For terminal output
                account_messages.append(message)
            self.account_entity.close()
            return account_messages
        else:
            print("No accounts found.")  # For terminal output
            self.account_entity.close()
            return ["No accounts found."]


    def fetch_account_by_website(self, website):
            """Fetch the username and password where the website matches."""
            self.account_entity.connect()
            account = self.account_entity.fetch_account_by_website(website)  # Call the entity method
            self.account_entity.close()
            return account
```

```
    def delete_account(self, account_id):
        self.account_entity.connect()
        self.account_entity.delete_account(account_id)
        self.account_entity.reset_id_sequence()
        self.account_entity.close()
```

----- Contents of AvailabilityControl.py -----

----- Contents of BotControl.py -----

```
import asyncio

class BotControl:
    def __init__(self, bot):
        self.bot = bot

    async def send_greeting(self):
        """Sends a greeting when the bot comes online."""
        channel = self.bot.get_channel(self.bot.config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online! type '!project_help' to see what I can do")

    async def stop_bot(self):
        """Stops the bot gracefully, ensuring all connections are closed."""
```

```
        print("Bot is stopping...")

        await self.bot.close()
```

----- Contents of BrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl  # Use AccountControl for consistency

class BrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()  # Use AccountControl to fetch credentials

    def launch_browser(self, user, incognito=False):
        return self.browser_entity.launch_browser(incognito=incognito, user=user)
```

----- Contents of ChatControl.py -----

```
# ChatControl in control/ChatControl.py
class ChatControl:
    def process_non_prefixed_message(self, message):
        """Process non-prefixed messages like 'hi', 'hello'."""
        if message.lower() in ["hi", "hello"]:
            return "Hello! How can I assist you today? Type !project_help for assistance."
        else:
            return "I didn't recognize that. Type !project_help to see available commands."
```

```
    def handle_unrecognized_command(self):
        """Handle unrecognized command from on_command_error."""
        return "I didn't recognize that command. Type !project_help for assistance."
```

----- Contents of CloseBrowserControl.py -----

```
from entity.BrowserEntity import BrowserEntity

class CloseBrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def close_browser(self):
        return self.browser_entity.close_browser()
```

----- Contents of DataExtractionControl.py -----

----- Contents of HelpControl.py -----

```
class HelpControl:
    def get_help_message(self):
        """Returns a list of available bot commands."""
        return (
            "Here are the available commands:\n"
            "!project_help - Get help on available commands.\n"
            "!chat_with_bot - Say hi to the bot.\n"
```

```
            "!login_to_website - Log in to a website.\n"
            "!launch_browser - Launch the browser.\n"
            "!close_browser - Close the browser.\n"
            "!navigate_to_website - Navigate to a website.\n"
            "!track_price - Track a product price.\n"
            "!check_price - Check the price of a product.\n"
            "!check_availability - Check the availability of a product.\n"
            "!stop_tracking - Stop tracking a product.\n"
            "!receive_notifications - Receive notifications for price changes.\n"
            "!extract_data - Export data to Excel or HTML.\n"
            "!stop - Stop the bot.\n"
        )



----- Contents of LoginControl.py -----
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl

class LoginControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()

    async def login(self, site, incognito=False, retries=1):
        # Fetch credentials using AccountControl
        account = self.account_control.fetch_account_by_website(site)
        if account:
```

```
            username, password = account
            return await self.browser_entity.login(site, username, password, incognito, retries)
        else:
            return f"No account found for website {site}"



----- Contents of MonitorPriceControl.py -----
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class MonitorPriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("MonitorPriceControl")

    async def monitor_price(self, ctx, url, frequency=1):
        """Monitor the price at a given interval."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                await ctx.send(f"Monitoring price every {frequency} minute(s).")
                previous_price = None

                while True:
                    current_price = self.price_entity.get_price(url)
                    if current_price:
```

```
                if previous_price is None:
                    await ctx.send(f"Starting price monitoring. Current price is: {current_price}")
                else:
                    if current_price > previous_price:
                        await ctx.send(f"Price went up! Current price: {current_price} (Previous: {previous_price})")
                    elif current_price < previous_price:
                        await ctx.send(f"Price went down! Current price: {current_price} (Previous: {previous_price})")
                    else:
                        await ctx.send(f"Price remains the same: {current_price}")
                    previous_price = current_price
                else:
                    await ctx.send("Failed to retrieve the price.")
                await asyncio.sleep(frequency * 60)  # Wait for the next check
            except Exception as e:
                self.logger.error(f"Failed to monitor price for {url}: {e}")
                await ctx.send(f"Failed to monitor price: {e}")
        else:
            await ctx.send("This command can only be used in the designated channel.")
```

----- Contents of NavigationControl.py -----

```
from entity.BrowserEntity import BrowserEntity

class NavigationControl:
    def __init__(self):
```

```
        self.browser_entity = BrowserEntity()

    def navigate_to_url(self, url):
        """Navigate to a specific URL."""
        return self.browser_entity.navigate_to_url(url)
```

----- Contents of NotificationControl.py -----

----- Contents of PriceControl.py -----

```
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class PriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("PriceControl")

    async def get_price(self, ctx, url):
        """Fetch the current price from the given URL."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                price = self.price_entity.get_price(url)
                if price:
```

```
                return f"The current price is: {price}"
            else:
                return "Failed to retrieve the price."
        except Exception as e:
            self.logger.error(f"Failed to get price for {url}: {e}")
            return f"Error getting price: {e}"
    else:
        return "This command can only be used in the designated channel."
```

----- Contents of StopMonitoringControl.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of AccountControl.cpython-312.pyc -----

----- Contents of BotControl.cpython-312.pyc -----

----- Contents of BrowserControl.cpython-312.pyc -----

----- Contents of ChatControl.cpython-312.pyc -----

----- Contents of CloseBrowserControl.cpython-312.pyc -----

----- Contents of HelpControl.cpython-312.pyc -----

----- Contents of LoginControl.cpython-312.pyc -----

----- Contents of MonitorPriceControl.cpython-312.pyc -----

----- Contents of NavigationControl.cpython-312.pyc -----

----- Contents of PriceControl.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of AccountEntity.py -----

```
import psycopg2
from Config import Config


class AccountEntity:
    def __init__(self):
        self.dbname = "postgres"
        self.user = "postgres"
        self.host = "localhost"
        self.port = "5432"
        self.password = Config.DATABASE_PASSWORD

    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
                password=self.password,
                host=self.host,
                port=self.port
            )
            self.cursor = self.connection.cursor()
            print("Database Connection Established.")
        except Exception as error:
            print(f"Error connecting to the database: {error}")
            self.connection = None
            self.cursor = None
```

```
    def add_account(self, username, password, webSite):
        """Insert a new account into the accounts table."""
        try:
            if self.cursor:
                    self.cursor.execute("INSERT INTO accounts (username, password, website) VALUES
(%s, %s, %s)", (username, password, webSite))
                self.connection.commit()
                print(f"Account {username} added successfully.")
        except Exception as error:
            print(f"Error inserting account: {error}")

    def fetch_accounts(self):
        """Fetch all accounts from the accounts table."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts;")
                accounts = self.cursor.fetchall()
                return accounts
        except Exception as error:
            print(f"Error fetching accounts: {error}")
            return None

    def delete_account(self, account_id):
        """Delete an account by ID."""
        try:
            if self.cursor:
```

```
            self.cursor.execute("SELECT * FROM accounts WHERE id = %s", (account_id,))
            account = self.cursor.fetchone()
            if account:
                self.cursor.execute("DELETE FROM accounts WHERE id = %s", (account_id,))
                self.connection.commit()
                print(f"Account with ID {account_id} deleted successfully.")
            else:
                print(f"Account with ID {account_id} not found. No deletion performed.")
        except Exception as error:
            print(f"Error deleting account: {error}")



    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        try:
                        self.cursor.execute("SELECT username, password FROM accounts WHERE
LOWER(website) = LOWER(%s)", (website,))
            return self.cursor.fetchone()  # Returns one matching account
        except Exception as error:
            print(f"Error fetching account for website {website}: {error}")
            return None



    def reset_id_sequence(self):
        """Reset the account ID sequence to the next available value."""
        try:
```

```
            if self.cursor:
                self.cursor.execute("SELECT COALESCE(MAX(id), 0) + 1 FROM accounts")
                next_id = self.cursor.fetchone()[0]
                        self.cursor.execute("ALTER SEQUENCE accounts_id_seq RESTART WITH %s",
(next_id,))
                self.connection.commit()
                print(f"ID sequence reset to {next_id}.")
        except Exception as error:
            print(f"Error resetting ID sequence: {error}")


    def close(self):
        """Close the database connection."""
        if self.cursor:
            self.cursor.close()
        if self.connection:
            self.connection.close()
            print("Database Connection closed.")



----- Contents of BrowserEntity.py -----
import asyncio
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from utils.css_selectors import Selectors  # Import CSS selectors for the website
```

```
class BrowserEntity:
    _instance = None  # Singleton instance

    def __new__(cls, *args, **kwargs):
        if cls._instance is None:
            cls._instance = super(BrowserEntity, cls).__new__(cls)
            cls._instance.driver = None  # Initialize driver to None
        return cls._instance

    def launch_browser(self, incognito=False, user=None):
        if self.driver:
            print("Browser is already running. No need to launch a new one.")
            return "Browser is already running."

        try:
            # Special launch options as per your original implementation
            options = webdriver.ChromeOptions()

            # Add options to avoid crashing and improve performance
            options.add_argument("--remote-debugging-port=9222")
            options.add_experimental_option("excludeSwitches", ["enable-automation"])
            options.add_experimental_option('useAutomationExtension', False)
            options.add_argument("--start-maximized")
            options.add_argument("--disable-notifications")
            options.add_argument("--disable-popup-blocking")
            options.add_argument("--disable-infobars")
```

```
            options.add_argument("--disable-extensions")
            options.add_argument("--disable-webgl")
            options.add_argument("--disable-webrtc")
            options.add_argument("--disable-rtc-smoothing")

            if incognito:
                options.add_argument("--incognito")

            self.driver = webdriver.Chrome(service=Service(), options=options)
            success_message = "Chrome browser launched successfully in incognito mode." if incognito
else "Chrome browser launched successfully."
            print(f"Driver initialized: {self.driver}")  # Debug: Print the driver
            return success_message
        except Exception as e:
            error_message = f"Failed to launch browser: {e}"
            print(error_message)
            raise

    def navigate_to_url(self, url):
        if not self.driver:
            print("Driver is not initialized, launching browser first.")  # Debug
            self.launch_browser()
        try:
            self.driver.get(url)
            return f"Navigated to URL: {url}"
        except Exception as e:
            raise
```

```
    def close_browser(self):
        print(f"Closing browser. Current driver: {self.driver}")  # Debug: Check the driver status
        if self.driver:
            self.driver.quit()  # Close the browser session
            self.driver = None  # Set to None after closing
            print("Browser closed successfully.")
            return "Browser closed successfully."
        else:
            print("No browser is currently open.")
            return "No browser is currently open."



    async def login(self, site, username, password, incognito=False, retries=1):
        # Get the URL and selectors from css_selectors
        url = Selectors.get_selectors_for_url(site)['url']
        for attempt in range(retries):
            try:
                self.navigate_to_url(url)
                await asyncio.sleep(3)


                # Enter the email address
                                                email_field    =    self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['email_field'])
                email_field.click()
                email_field.send_keys(username)
                await asyncio.sleep(3)
```

```
            # Enter the password
                                        password_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['password_field'])
            password_field.click()
            password_field.send_keys(password)
            await asyncio.sleep(3)

            # Click the login button
                                        sign_in_button = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['SignIn_button'])
            sign_in_button.click()
            await asyncio.sleep(5)

            # Wait for the homepage to load after login
            WebDriverWait(self.driver, 30).until(
                                                EC.presence_of_element_located((By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['homePage'])))

            return f"Logged in to {url} successfully with username: {username}"
        except Exception as e:
            if attempt < retries - 1:
                await asyncio.sleep(3)
            else:
                raise e
```

----- Contents of DateEntity.py -----

----- Contents of NotificationEntity.py -----

----- Contents of PriceEntity.py -----

```
import time
from selenium.webdriver.common.by import By
from utils.css_selectors import Selectors
from entity.BrowserEntity import BrowserEntity  # Import the browser interaction logic

class PriceEntity:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def get_price(self, url):
        """Fetch the price from the provided URL using CSS selectors."""
        selectors = Selectors.get_selectors_for_url(url)
        if not selectors:
            raise ValueError(f"No selectors found for URL: {url}")

        # Navigate to the URL using the browser entity
        self.browser_entity.navigate_to_url(url)
        time.sleep(2)  # Wait for the page to load

        try:
```

```
            # Use the CSS selector to find the price on the page
                        price_element = self.browser_entity.driver.find_element(By.CSS_SELECTOR,
selectors['price'])
            price = price_element.text
            print(f"Price found: {price}")
            return price
        except Exception as e:
            print(f"Error finding price: {e}")
            return None
```

----- Contents of PriceHistoryEntity.py -----

----- Contents of __init__.py -----

#empty init file

----- Contents of AccountEntity.cpython-312.pyc -----

----- Contents of BrowserEntity.cpython-312.pyc -----

----- Contents of PriceEntity.cpython-312.pyc -----

----- Contents of __init__.cpython-312.pyc -----

----- Contents of project.txt -----

DiscordBotProject_CISC699 - Project Overview

Introduction

This project is a Discord bot designed to perform various tasks, including tracking product prices, checking availability, logging into websites, and exporting data.

The bot interacts with users via commands sent through Discord and responds based on the requested use case.

The project follows a clear structure, adhering to software engineering best practices, and separates the logic into Boundary, Control, and Entity objects to manage the flow of data and logic.

Scroll all the way down for project outline

Objects and Their Roles

Entity Objects

Entity objects represent the core business data and operations related to those entities. They store data and perform business logic related to that data. They do not interact directly with the user.

ProductEntity: Represents product information such as price and features. It handles product-related data (e.g., retrieving the current price).

DateEntity: Handles date and availability logic for booking or checking availability of services.

AccountEntity: Manages user login credentials for websites like BestBuy or eBay.

TrackingHistoryEntity: Stores and tracks historical data on product prices. Helps to compare past prices with current ones.

BrowserEntity: Manages the state of the browser (e.g., if the browser is running, whether it's in incognito mode, etc.).

NotificationEntity: Handles user preferences for receiving notifications, such as when prices change or product availability is updated.

Control Objects

Control objects are responsible for handling the logic of each use case. They interact with entity objects to manage data and handle business rules. Control objects execute the steps required to fulfill a use case.

HelpControl: Provides a list of commands available to the user.

ChatControl: Handles basic user interaction, such as greetings and responses to basic phrases like "hi" or "hello."

LoginControl: Manages the process of logging into a website, including retrieving login credentials from the database and passing them to the browser.

BrowserControl: Manages the launch and setup of the browser, including handling incognito mode and configuring the browser.

CloseBrowserControl: Handles the logic for closing the browser when requested by the user.

NavigationControl: Manages the process of navigating to a specific URL in the browser.

ProductTrackingControl: Manages the tracking of a product's price over time, scheduling regular price checks.

ProductControl: Checks the current price of a product and retrieves relevant product data.

AvailabilityControl: Handles checking the availability of a product or service based on user-provided dates.

StopTrackingControl: Stops the tracking process for a product or service.

NotificationControl: Monitors for changes in tracked products and sends notifications when a price or availability change occurs.

DataExtractionControl: Manages the extraction of tracking data, exporting it to Excel or HTML files.

BotControl: Manages the overall lifecycle of the Discord bot, including starting, stopping, and managing the registration of commands.

Boundary Objects

Boundary objects serve as the bridge between the user (or external actor) and the system. They collect data from the user and forward it to the appropriate control object. Boundary objects are responsible for interacting with the actor but not for executing business logic.

HelpBoundary: Collects the user?s help request and forwards it to HelpControl.

ChatBoundary: Receives chat commands from the user and forwards them to ChatControl.

LoginBoundary: Collects login credentials from the user and forwards them to LoginControl.

BrowserBoundary: Receives commands to launch the browser and forwards them to BrowserControl.

CloseBrowserBoundary: Receives the user?s request to close the browser and forwards it to CloseBrowserControl.

NavigationBoundary: Receives URL input from the user and forwards it to NavigationControl.

ProductTrackingBoundary: Collects the user?s request to track a product and forwards it to ProductTrackingControl.

ProductBoundary: Receives the user?s request to check a product price and forwards it to ProductControl.

AvailabilityBoundary: Collects the user?s availability check request and forwards it to AvailabilityControl.

StopTrackingBoundary: Receives the user?s request to stop tracking a product and forwards it to StopTrackingControl.

NotificationBoundary: Collects user preferences for receiving notifications and forwards them to NotificationControl.

DataExtractionBoundary: Collects the user?s request to export data and forwards it to DataExtractionControl.

StopBoundary: Receives the request to stop the bot and forwards it to BotControl.

Capabilities

Here?s what the bot can do:

1. !project_help

Description: Provides a list of available commands the user can issue.

Objects Involved:

Boundary: HelpBoundary

Control: HelpControl

Interaction: HelpBoundary collects the user?s help request and forwards it to HelpControl, which responds with the list of commands.

2. !chat_with_bot

Description: Responds to simple greetings (e.g., "hi", "hello") and provides a welcome message.

Objects Involved:

Boundary: ChatBoundary

Control: ChatControl

Interaction: ChatBoundary collects chat input and forwards it to ChatControl, which sends back a predefined response.

3. !login_to_website

Description: Logs into a website using stored credentials (e.g., BestBuy).

Objects Involved:

Boundary: LoginBoundary

Control: LoginControl, BrowserControl, NavigationControl

Entity: AccountEntity

Interaction:

LoginBoundary collects login credentials and forwards them to LoginControl.

LoginControl works with BrowserControl to launch the browser.

NavigationControl navigates to the website's login page.

AccountEntity retrieves the stored credentials from the database and logs the user in.

4. !launch_browser

Description: Launches the browser, optionally in incognito mode.

Objects Involved:

Boundary: BrowserBoundary

Control: BrowserControl

Entity: BrowserEntity

Interaction: BrowserBoundary collects the user's request to launch the browser and sends it to BrowserControl. BrowserControl uses BrowserEntity to configure and launch the browser.

5. !close_browser

Description: Closes the currently open browser session.

Objects Involved:

Boundary: CloseBrowserBoundary

Control: CloseBrowserControl

Entity: BrowserEntity

Interaction: CloseBrowserBoundary forwards the user?s request to CloseBrowserControl, which then tells BrowserEntity to close the browser session.

6. !navigate_to_website

Description: Navigates to a specific website URL in the browser.

Objects Involved:

Boundary: NavigationBoundary

Control: NavigationControl

Entity: BrowserEntity

Interaction: NavigationBoundary collects the URL input from the user and forwards it to NavigationControl. NavigationControl instructs BrowserEntity to navigate to the specified URL.

7. !track_price

Description: Tracks the price of a product over time and sends notifications if the price changes.

Objects Involved:

Boundary: ProductTrackingBoundary

Control: ProductTrackingControl, ProductControl, NotificationControl

Entity: ProductEntity, TrackingHistoryEntity, NotificationEntity

Interaction:

ProductTrackingBoundary collects the product URL from the user.

ProductTrackingControl initiates price tracking and uses ProductControl to fetch the current price.

The current price is stored in TrackingHistoryEntity.

If there?s a price change, NotificationControl sends an alert via NotificationEntity.

8. !check_price

Description: Manually checks the current price of a product.

Objects Involved:

Boundary: ProductBoundary

Control: ProductControl

Entity: ProductEntity

Interaction: ProductBoundary collects the product information from the user, and ProductControl retrieves the current price using ProductEntity.

9. !check_availability

Description: Checks the availability of a product or service on a specific date.

Objects Involved:

Boundary: AvailabilityBoundary

Control: AvailabilityControl

Entity: DateEntity

Interaction: AvailabilityBoundary collects the date and product/service details. AvailabilityControl checks the availability via DateEntity.

10. !stop_tracking

Description: Stops tracking the price or availability of a product.

Objects Involved:

Boundary: StopTrackingBoundary

Control: StopTrackingControl

Entity: TrackingHistoryEntity

Interaction: StopTrackingBoundary collects the stop request from the user. StopTrackingControl stops the tracking and updates TrackingHistoryEntity.

11. !receive_notifications

Description: Sends notifications when there?s a change in price or availability for tracked products/services.

Objects Involved:

Boundary: NotificationBoundary

Control: NotificationControl

Entity: NotificationEntity, TrackingHistoryEntity

Interaction: NotificationBoundary collects the user?s preferences for receiving notifications. NotificationControl monitors for changes and uses NotificationEntity to send alerts when changes occur.

12. !extract_data

Description: Extracts the tracked product data and exports it to Excel or HTML format.

Objects Involved:

Boundary: DataExtractionBoundary

Control: DataExtractionControl

Entity: TrackingHistoryEntity

Utilities: ExcelUtils, HTMLUtils

Interaction: DataExtractionBoundary collects the user?s request for data extraction. DataExtractionControl retrieves data from TrackingHistoryEntity and uses ExcelUtils or HTMLUtils to export the data to the desired format.

13. !stop

Description: Stops the Discord bot from running.

Objects Involved:

Boundary: StopBoundary

Control: BotControl

Interaction: StopBoundary collects the stop command from the user and forwards it to BotControl, which gracefully stops the bot.

DiscordBotProject_CISC699/

?

??? boundary/

?   ??? AccountBoundary.py

?   ??? HelpBoundary.py

?   ??? ChatBoundary.py

?   ??? LoginBoundary.py

?   ??? BrowserBoundary.py

?   ??? CloseBrowserBoundary.py

?   ??? NavigationBoundary.py

?   ??? ProductTrackingBoundary.py

?   ??? ProductBoundary.py

?   ??? AvailabilityBoundary.py

?   ??? StopTrackingBoundary.py

?   ??? NotificationBoundary.py

?   ??? DataExtractionBoundary.py

?   ??? StopBoundary.py

?

??? control/

?   ??? AccountControl.py

?   ??? HelpControl.py

?   ??? ChatControl.py

?   ??? LoginControl.py

?   ??? BrowserControl.py

?   ??? CloseBrowserControl.py

?   ??? NavigationControl.py

?   ??? ProductTrackingControl.py

?   ??? ProductControl.py

?   ??? AvailabilityControl.py

?   ??? StopTrackingControl.py

?   ??? NotificationControl.py

?   ??? DataExtractionControl.py

?   ??? BotControl.py

?

??? entity/

?   ??? ProductEntity.py

?   ??? DateEntity.py

?   ??? AccountEntity.py

?   ??? TrackingHistoryEntity.py

?   ??? BrowserEntity.py

?   ??? NotificationEntity.py

?

??? utils/

?   ??? ExcelUtils.py

?   ??? HTMLUtils.py

?   ??? DiscordUtils.py

?

??? test/

?   ??? test_addAccount.py

?   ??? test_deleteAccount.py

?   ??? test_fetchAccounts.py

?   ??? test_excel_creation.py

?   ??? test_html_creation.py

?

??? Config.py

??? main.py

??? project.txt

----- Contents of project_structure.py -----

```
import os

def list_files_and_folders(directory, output_file):
    with open(output_file, 'w') as f:
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            f.write(f"Directory: {root}\n")
            for dir_name in dirs:
                f.write(f"  Folder: {dir_name}\n")
            for file_name in files:
                f.write(f"  File: {file_name}\n")

# Update the directory path to your project folder
```

```
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699"
output_file = os.path.join(project_directory, "project_structure.txt")

# Call the function to list files and save output to .txt
list_files_and_folders(project_directory, output_file)

print(f"File structure saved to {output_file}")
```

----- Contents of project_structure.txt -----

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699

- Folder: boundary
- Folder: control
- Folder: entity
- Folder: test
- Folder: utils
- File: Config.py
- File: main.py
- File: project.txt
- File: project_structure.txt
- File: temporary.py
- File: Tests_URLs.txt

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\boundary

- File: AccountBoundary.py

File: AvailabilityBoundary.py

File: BotBoundary.py

File: BrowserBoundary.py

File: CloseBrowserBoundary.py

File: DataExtractionBoundary.py

File: HelpBoundary.py

File: LoginBoundary.py

File: NavigationBoundary.py

File: NotificationBoundary.py

File: ProductBoundary.py

File: ProductTrackingBoundary.py

File: StopBoundary.py

File: StopTrackingBoundary.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\control

File: AccountControl.py

File: AvailabilityControl.py

File: BotControl.py

File: BrowserControl.py

File: ChatControl.py

File: CloseBrowserControl.py

File: DataExtractionControl.py

File: HelpControl.py

File: LoginControl.py

File: NavigationControl.py

File: NotificationControl.py

File: ProductControl.py

File: ProductTrackingControl.py

File: StopTrackingControl.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\entity

File: AccountEntity.py

File: BrowserEntity.py

File: DateEntity.py

File: NotificationEntity.py

File: ProductEntity.py

File: TrackingHistoryEntity.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\test

File: test_addAccount.py

File: test_deleteAccount.py

File: test_excel_creation.py

File: test_fetchAccounts.py

File: test_html_creation.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\utils

File: css_selectors.py

File: DiscordUtils.py

File: ExcelUtils.py

File: HTMLUtils.py

----- Contents of temporary.py -----

```
import os


def get_all_files_text(directory, output_file):
    with open(output_file, 'w', encoding='utf-8') as outfile:
        # Traverse the directory
        for foldername, subfolders, filenames in os.walk(directory):
            for filename in filenames:
                file_path = os.path.join(foldername, filename)
                try:
                    # Open and read each file's content
                    with open(file_path, 'r', encoding='utf-8') as infile:
                        outfile.write(f"----- Contents of {filename} -----\n")
                        outfile.write(infile.read())
                        outfile.write("\n\n")
                except Exception as e:
                    print(f"Error reading {file_path}: {e}")


if __name__ == "__main__":
        directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699"
        output_file = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\all_files_content.txt"
    get_all_files_text(directory, output_file)
```

----- Contents of Tests_URLs.txt -----

database password: postgres

Working Commands: Test commands

!project_help

!login bestbuy

!launch_browser

!close_browser

!navigate_to_website https://www.google.com/

!monitor_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price
https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

!check_availability https://www.opentable.com/r/bar-spero-washington/

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22"

!stop_monitoring

!stop

**********

Working on it:

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22" "8:00 PM"

**********

URLs to Test:

https://www.opentable.com/r/bar-spero-washington/

https://www.ebay.com/itm/314411766963?_trkparms=amclksrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3A

vlp_homepage

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

----- Contents of test_addAccount.py -----

```
import sys, os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_add_account():
    account_control = AccountControl()

    # Adding a new account
    account_control.add_account("newUser", "newPassword123", "newWebsite")

if __name__ == "__main__":
    test_add_account()
```

----- Contents of test_deleteAccount.py -----

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl
```

```
def test_delete_account():
    account_control = AccountControl()

    account_control.delete_account(4)

if __name__ == "__main__":
    test_delete_account()
```

----- Contents of test_excel_creation.py -----

----- Contents of test_fetchAccounts.py -----

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_fetch_accounts():
    account_control = AccountControl()

    # Fetching all accounts
    account_control.fetch_accounts()

def test_fetch_account_by_website(website):
```

```
    account_control = AccountControl()

    # Fetch the account by website directly
    account = account_control.fetch_account_by_website(website)

    if account:
        username, password = account  # Unpack the returned tuple
        print(f"Website: {website}, Username: {username}, Password: {password}")
    else:
        print(f"No account found for website: {website}")

if __name__ == "__main__":
    test_fetch_accounts()
    test_fetch_account_by_website("ebay")
```

----- Contents of test_html_creation.py -----

----- Contents of __init__.py -----

```
#empty init file
```

----- Contents of css_selectors.py -----

```
class Selectors:
    SELECTORS = {
        "trendyol": {
            "price": ".featured-prices .prc-dsc"  # Selector for Trendyol price
```

```
    },
    "ebay": {
        "url": "https://signin.ebay.com/signin/",
        "email_field": "#userid",
        "continue_button": "[data-testid*='signin-continue-btn']",
        "password_field": "#pass",
        "login_button": "#sgnBt",
        "price": ".x-price-primary span"  # CSS selector for Ebay price
    },
    "bestbuy": {
        "url": "https://www.bestbuy.com/signin/",
        "email_field": "#fld-e",
        #"continue_button": ".cia-form__controls  button",
        "password_field": "#fld-p1",
        "SignIn_button": ".cia-form__controls  button",
        "price": "[data-testid='customer-price'] span",  # CSS selector for BestBuy price
        "homePage": ".v-p-right-xxs.line-clamp"
    },
    "opentable": {
        "url": "https://www.opentable.com/",
        "date_field": "#restProfileSideBarDtpDayPicker-label",
        "time_field": "#restProfileSideBartimePickerDtpPicker",
        "find_table_button": ".find-table-button",  # Example selector for the Find Table button
        "availability_result": ".availability-result",  # Example selector for availability results
            "show_next_available_button": "button[data-test='multi-day-availability-button']",  # Show
next available button
        "available_dates": "ul[data-test='time-slots'] > li",  # Available dates and times
```

```
            "no_availability": "div._8ye6OVzeOuU- span"
        }
    }

    @staticmethod
    def get_selectors_for_url(url):
        for keyword, selectors in Selectors.SELECTORS.items():
            if keyword in url.lower():
                return selectors
        return None  # Return None if no matching selectors are found
```

----- Contents of DiscordUtils.py -----

----- Contents of ExcelUtils.py -----

----- Contents of HTMLUtils.py -----

----- Contents of css_selectors.cpython-312.pyc -----

----- Contents of HelpText.cpython-312.pyc -----

----- Contents of Config.cpython-312.pyc -----

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\AccountBoundary.py ---

```
from discord.ext import commands
```

```
from control.AccountControl import AccountControl

class AccountBoundary(commands.Cog):

    def __init__(self, bot):
        self.bot = bot
        self.account_control = AccountControl()


    @commands.command(name='fetch_accounts')
    async def fetch_accounts(self, ctx):
        """Fetch and display all accounts."""
        accounts = self.account_control.fetch_accounts()

        # Send each account or the no accounts message to Discord
        for account in accounts:
            await ctx.send(account)



    @commands.command(name="add_account")
    async def add_account(self, ctx, username: str, password: str):
        """Add a new user account to the database."""
        result = self.account_control.add_account(username, password)
        if result:
            await ctx.send(f"Account for {username} added successfully.")
        else:
```

```
            await ctx.send(f"Failed to add account for {username}.")



    @commands.command(name="delete_account")
    async def delete_account(self, ctx, user_id: int):
        """Delete a user account from the database."""
        result = self.account_control.delete_account(user_id)
        if result:
            await ctx.send(f"Account with ID {user_id} deleted successfully.")
        else:
            await ctx.send(f"Failed to delete account with ID {user_id}.")
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\AvailabilityBoundary.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\BotBoundary.py ---

```
from discord.ext import commands
from control.ChatControl import ChatControl
from Config import Config

class BotBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.chat_control = ChatControl()
```

```
@commands.Cog.listener()
async def on_ready(self):
    """Bot startup message when ready."""
    print(f'Logged in as {self.bot.user.name}')
    channel = self.bot.get_channel(Config.CHANNEL_ID)
    if channel:
        await channel.send("Hi, I'm online!")

@commands.Cog.listener()
async def on_message(self, message):
    """Handle non-prefixed messages and command-prefixed messages."""
    if message.author == self.bot.user:
        return

    # Handle non-prefixed messages (like greetings)
    if not message.content.startswith('!'):
        response = self.chat_control.process_non_prefixed_message(message.content)
        await message.channel.send(response)

@commands.Cog.listener()
async def on_command_error(self, ctx, error):
    """Handle unrecognized commands."""
    if isinstance(error, commands.CommandNotFound):
        # Handle unknown command
        response = self.chat_control.handle_unrecognized_command()
        await ctx.send(response)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\BrowserBoundary.py ---

```
from discord.ext import commands
from control.BrowserControl import BrowserControl

class BrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.browser_control = BrowserControl()

    @commands.command(name='launch_browser')
    async def launch_browser(self, ctx, *args):
        """Command to launch the browser."""
        incognito = "incognito" in args
        response = self.browser_control.launch_browser(ctx.author, incognito)
        await ctx.send(response)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\CloseBrowserBoundary.py ---

```
from discord.ext import commands
from control.CloseBrowserControl import CloseBrowserControl

class CloseBrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.close_browser_control = CloseBrowserControl()
```

```
    @commands.command(name='close_browser')
    async def close_browser(self, ctx):
        """Command to close the browser."""
        response = self.close_browser_control.close_browser()
        await ctx.send(response)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\DataExtractionBoundary.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\HelpBoundary.py ---

```
from discord.ext import commands
from control.HelpControl import HelpControl

class HelpBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.help_control = HelpControl()

    @commands.command(name='project_help')
    async def project_help(self, ctx):
        """Handles the project_help command."""
        help_message = self.help_control.get_help_message()
        await ctx.send(help_message)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\LoginBoundary.py ---

```
from discord.ext import commands
from control.LoginControl import LoginControl

class LoginBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.login_control = LoginControl()

    @commands.command(name='login')
    async def login(self, ctx, site: str, *args):
        """Command to log into a website using stored credentials."""
        incognito = "incognito" in args
        retries = next((int(arg) for arg in args if arg.isdigit()), 1)
        response = await self.login_control.login(site, incognito, retries)
        await ctx.send(response)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\MonitorPriceBoundary.py ---

```
from discord.ext import commands
from control.MonitorPriceControl import MonitorPriceControl

class MonitorPriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.monitor_price_control = MonitorPriceControl()

    @commands.command(name='monitor_price')
```

```
    async def monitor_price(self, ctx, url: str, frequency: int = 1):
        """Command to monitor the price at regular intervals."""
        await self.monitor_price_control.monitor_price(ctx, url, frequency)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\NavigationBoundary.py ---

```
from discord.ext import commands
from control.NavigationControl import NavigationControl

class NavigationBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.navigation_control = NavigationControl()

    @commands.command(name='navigate_to_website')
    async def navigate_to_website(self, ctx, url: str):
        """Command to navigate to a specified URL."""
        response = self.navigation_control.navigate_to_url(url)
        await ctx.send(response)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\NotificationBoundary.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\PriceBoundary.py ---

```
from discord.ext import commands
from control.PriceControl import PriceControl
```

```
class PriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.price_control = PriceControl()

    @commands.command(name='get_price')
    async def get_price(self, ctx, url: str):
        """Command to get the price from the given URL."""
        response = await self.price_control.get_price(ctx, url)
        await ctx.send(response)

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\StopBoundary.py ---
from discord.ext import commands
from control.BotControl import BotControl

class StopBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.bot_control = BotControl(bot)

    @commands.command(name="stop_bot")
    async def stop_bot(self, ctx):
        """Handles the stop command and gracefully shuts down the bot."""
        await ctx.send("Stopping the bot...")
        await self.bot_control.stop_bot()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary\StopMonitoringBoundary.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\boundary__init__.py ---

#empty init file

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\AccountControl.py ---

```
from entity.AccountEntity import AccountEntity

class AccountControl:
    def __init__(self):
        self.account_entity = AccountEntity()

    def add_account(self, username, password, webSite):
        self.account_entity.connect()
        self.account_entity.add_account(username, password, webSite)
        self.account_entity.close()

    def fetch_accounts(self):
        """Fetch all accounts and return them."""
        self.account_entity.connect()
        accounts = self.account_entity.fetch_accounts()

        if accounts:
            account_messages = []
```

```
            for account in accounts:
                message = f"ID: {account[0]}, Username: {account[1]}, Password: {account[2]}, Website: {account[3]}"
                print(message)  # For terminal output
                account_messages.append(message)
            self.account_entity.close()
            return account_messages
        else:
            print("No accounts found.")  # For terminal output
            self.account_entity.close()
            return ["No accounts found."]


    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        self.account_entity.connect()
        account = self.account_entity.fetch_account_by_website(website)  # Call the entity method
        self.account_entity.close()
        return account


    def delete_account(self, account_id):
        self.account_entity.connect()
        self.account_entity.delete_account(account_id)
        self.account_entity.reset_id_sequence()
        self.account_entity.close()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\AvailabilityControl.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\BotControl.py ---

```
import asyncio

class BotControl:
    def __init__(self, bot):
        self.bot = bot

    async def send_greeting(self):
        """Sends a greeting when the bot comes online."""
        channel = self.bot.get_channel(self.bot.config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online! type '!project_help' to see what I can do")

    async def stop_bot(self):
        """Stops the bot gracefully, ensuring all connections are closed."""
        print("Bot is stopping...")

        await self.bot.close()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\BrowserControl.py ---

```
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl  # Use AccountControl for consistency
```

```
class BrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()  # Use AccountControl to fetch credentials

    def launch_browser(self, user, incognito=False):
        return self.browser_entity.launch_browser(incognito=incognito, user=user)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\ChatControl.py ---

```
# ChatControl in control/ChatControl.py
class ChatControl:
    def process_non_prefixed_message(self, message):
        """Process non-prefixed messages like 'hi', 'hello'."""
        if message.lower() in ["hi", "hello"]:
            return "Hello! How can I assist you today? Type !project_help for assistance."
        else:
            return "I didn't recognize that. Type !project_help to see available commands."

    def handle_unrecognized_command(self):
        """Handle unrecognized command from on_command_error."""
        return "I didn't recognize that command. Type !project_help for assistance."
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\CloseBrowserControl.py ---

```
from entity.BrowserEntity import BrowserEntity
```

```
class CloseBrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def close_browser(self):
        return self.browser_entity.close_browser()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\DataExtractionControl.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\HelpControl.py ---

```
class HelpControl:
    def get_help_message(self):
        """Returns a list of available bot commands."""
        return (
            "Here are the available commands:\n"
            "!project_help - Get help on available commands.\n"
            "!chat_with_bot - Say hi to the bot.\n"
            "!login_to_website - Log in to a website.\n"
            "!launch_browser - Launch the browser.\n"
            "!close_browser - Close the browser.\n"
            "!navigate_to_website - Navigate to a website.\n"
            "!track_price - Track a product price.\n"
            "!check_price - Check the price of a product.\n"
            "!check_availability - Check the availability of a product.\n"
            "!stop_tracking - Stop tracking a product.\n"
```

```
            "!receive_notifications - Receive notifications for price changes.\n"
            "!extract_data - Export data to Excel or HTML.\n"
            "!stop - Stop the bot.\n"
        )

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\LoginControl.py ---
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl

class LoginControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()

    async def login(self, site, incognito=False, retries=1):
        # Fetch credentials using AccountControl
        account = self.account_control.fetch_account_by_website(site)
        if account:
            username, password = account
            return await self.browser_entity.login(site, username, password, incognito, retries)
        else:
            return f"No account found for website {site}"

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\MonitorPriceControl.py ---
import asyncio
```

```
from entity.PriceEntity import PriceEntity
from Config import Config
import logging


class MonitorPriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("MonitorPriceControl")

    async def monitor_price(self, ctx, url, frequency=1):
        """Monitor the price at a given interval."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                await ctx.send(f"Monitoring price every {frequency} minute(s).")
                previous_price = None

                while True:
                    current_price = self.price_entity.get_price(url)
                    if current_price:
                        if previous_price is None:
                            await ctx.send(f"Starting price monitoring. Current price is: {current_price}")
                        else:
                            if current_price > previous_price:
                                await ctx.send(f"Price went up! Current price: {current_price} (Previous:
{previous_price})")
                            elif current_price < previous_price:
                                await ctx.send(f"Price went down! Current price: {current_price} (Previous:
```

```
{previous_price})")
                        else:
                            await ctx.send(f"Price remains the same: {current_price}")
                    previous_price = current_price
                else:
                    await ctx.send("Failed to retrieve the price.")
                await asyncio.sleep(frequency * 60)  # Wait for the next check
            except Exception as e:
                self.logger.error(f"Failed to monitor price for {url}: {e}")
                await ctx.send(f"Failed to monitor price: {e}")
        else:
            await ctx.send("This command can only be used in the designated channel.")
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\NavigationControl.py ---

```
from entity.BrowserEntity import BrowserEntity

class NavigationControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def navigate_to_url(self, url):
        """Navigate to a specific URL."""
        return self.browser_entity.navigate_to_url(url)
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\NotificationControl.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\PriceControl.py ---

```
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging


class PriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("PriceControl")

    async def get_price(self, ctx, url):
        """Fetch the current price from the given URL."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                price = self.price_entity.get_price(url)
                if price:
                    return f"The current price is: {price}"
                else:
                    return "Failed to retrieve the price."
            except Exception as e:
                self.logger.error(f"Failed to get price for {url}: {e}")
                return f"Error getting price: {e}"
        else:
            return "This command can only be used in the designated channel."
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control\StopMonitoringControl.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\control__init__.py ---

```
#empty init file
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity\AccountEntity.py ---

```
import psycopg2
from Config import Config

class AccountEntity:
    def __init__(self):
        self.dbname = "postgres"
        self.user = "postgres"
        self.host = "localhost"
        self.port = "5432"
        self.password = Config.DATABASE_PASSWORD

    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
                password=self.password,
                host=self.host,
```

```
            port=self.port
        )
        self.cursor = self.connection.cursor()
        print("Database Connection Established.")
    except Exception as error:
        print(f"Error connecting to the database: {error}")
        self.connection = None
        self.cursor = None

def add_account(self, username, password, webSite):
    """Insert a new account into the accounts table."""
    try:
        if self.cursor:
            self.cursor.execute("INSERT INTO accounts (username, password, website) VALUES
(%s, %s, %s)", (username, password, webSite))
            self.connection.commit()
            print(f"Account {username} added successfully.")
    except Exception as error:
        print(f"Error inserting account: {error}")

def fetch_accounts(self):
    """Fetch all accounts from the accounts table."""
    try:
        if self.cursor:
            self.cursor.execute("SELECT * FROM accounts;")
            accounts = self.cursor.fetchall()
            return accounts
```

```
        except Exception as error:
            print(f"Error fetching accounts: {error}")
            return None

    def delete_account(self, account_id):
        """Delete an account by ID."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts WHERE id = %s", (account_id,))
                account = self.cursor.fetchone()
                if account:
                    self.cursor.execute("DELETE FROM accounts WHERE id = %s", (account_id,))
                    self.connection.commit()
                    print(f"Account with ID {account_id} deleted successfully.")
                else:
                    print(f"Account with ID {account_id} not found. No deletion performed.")
        except Exception as error:
            print(f"Error deleting account: {error}")


    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        try:
                        self.cursor.execute("SELECT username, password FROM accounts WHERE
LOWER(website) = LOWER(%s)", (website,))
            return self.cursor.fetchone()  # Returns one matching account
        except Exception as error:
```

```
            print(f"Error fetching account for website {website}: {error}")
            return None



    def reset_id_sequence(self):
        """Reset the account ID sequence to the next available value."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT COALESCE(MAX(id), 0) + 1 FROM accounts")
                next_id = self.cursor.fetchone()[0]
                        self.cursor.execute("ALTER SEQUENCE accounts_id_seq RESTART WITH %s",
(next_id,))
                self.connection.commit()
                print(f"ID sequence reset to {next_id}.")
        except Exception as error:
            print(f"Error resetting ID sequence: {error}")


    def close(self):
        """Close the database connection."""
        if self.cursor:
            self.cursor.close()
        if self.connection:
            self.connection.close()
            print("Database Connection closed.")

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC
```

699\DiscordBotProject_CISC699\entity\BrowserEntity.py ---

```
import asyncio
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from utils.css_selectors import Selectors  # Import CSS selectors for the website


class BrowserEntity:
    _instance = None  # Singleton instance

    def __new__(cls, *args, **kwargs):
        if cls._instance is None:
            cls._instance = super(BrowserEntity, cls).__new__(cls)
            cls._instance.driver = None  # Initialize driver to None
        return cls._instance

    def launch_browser(self, incognito=False, user=None):
        if self.driver:
            print("Browser is already running. No need to launch a new one.")
            return "Browser is already running."

        try:
            # Special launch options as per your original implementation
            options = webdriver.ChromeOptions()
```

```
            # Add options to avoid crashing and improve performance
            options.add_argument("--remote-debugging-port=9222")
            options.add_experimental_option("excludeSwitches", ["enable-automation"])
            options.add_experimental_option('useAutomationExtension', False)
            options.add_argument("--start-maximized")
            options.add_argument("--disable-notifications")
            options.add_argument("--disable-popup-blocking")
            options.add_argument("--disable-infobars")
            options.add_argument("--disable-extensions")
            options.add_argument("--disable-webgl")
            options.add_argument("--disable-webrtc")
            options.add_argument("--disable-rtc-smoothing")

            if incognito:
                options.add_argument("--incognito")

            self.driver = webdriver.Chrome(service=Service(), options=options)
            success_message = "Chrome browser launched successfully in incognito mode." if incognito
else "Chrome browser launched successfully."
            print(f"Driver initialized: {self.driver}")  # Debug: Print the driver
            return success_message
        except Exception as e:
            error_message = f"Failed to launch browser: {e}"
            print(error_message)
            raise

    def navigate_to_url(self, url):
```

```
    if not self.driver:
        print("Driver is not initialized, launching browser first.")  # Debug
        self.launch_browser()
    try:
        self.driver.get(url)
        return f"Navigated to URL: {url}"
    except Exception as e:
        raise

def close_browser(self):
    print(f"Closing browser. Current driver: {self.driver}")  # Debug: Check the driver status
    if self.driver:
        self.driver.quit()  # Close the browser session
        self.driver = None  # Set to None after closing
        print("Browser closed successfully.")
        return "Browser closed successfully."
    else:
        print("No browser is currently open.")
        return "No browser is currently open."

async def login(self, site, username, password, incognito=False, retries=1):
    # Get the URL and selectors from css_selectors
    url = Selectors.get_selectors_for_url(site)['url']
    for attempt in range(retries):
        try:
            self.navigate_to_url(url)
```

```
        await asyncio.sleep(3)

        # Enter the email address
                                    email_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['email_field'])
        email_field.click()
        email_field.send_keys(username)
        await asyncio.sleep(3)

        # Enter the password
                                password_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['password_field'])
        password_field.click()
        password_field.send_keys(password)
        await asyncio.sleep(3)

        # Click the login button
                                sign_in_button = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['SignIn_button'])
        sign_in_button.click()
        await asyncio.sleep(5)

        # Wait for the homepage to load after login
        WebDriverWait(self.driver, 30).until(
                                        EC.presence_of_element_located((By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['homePage'])))
```

```
            return f"Logged in to {url} successfully with username: {username}"
        except Exception as e:
            if attempt < retries - 1:
                await asyncio.sleep(3)
            else:
                raise e
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity\DateEntity.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity\NotificationEntity.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity\PriceEntity.py ---

```
import time
from selenium.webdriver.common.by import By
from utils.css_selectors import Selectors
from entity.BrowserEntity import BrowserEntity  # Import the browser interaction logic

class PriceEntity:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def get_price(self, url):
        """Fetch the price from the provided URL using CSS selectors."""
        selectors = Selectors.get_selectors_for_url(url)
```

```
        if not selectors:
            raise ValueError(f"No selectors found for URL: {url}")

        # Navigate to the URL using the browser entity
        self.browser_entity.navigate_to_url(url)
        time.sleep(2)  # Wait for the page to load

        try:
            # Use the CSS selector to find the price on the page
                        price_element = self.browser_entity.driver.find_element(By.CSS_SELECTOR,
selectors['price'])
            price = price_element.text
            print(f"Price found: {price}")
            return price
        except Exception as e:
            print(f"Error finding price: {e}")
            return None
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity\PriceHistoryEntity.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\entity__init__.py ---

```
#empty init file
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\other\project.txt ---

DiscordBotProject_CISC699 - Project Overview

## Introduction

This project is a Discord bot designed to perform various tasks, including tracking product prices, checking availability, logging into websites, and exporting data.

The bot interacts with users via commands sent through Discord and responds based on the requested use case.

The project follows a clear structure, adhering to software engineering best practices, and separates the logic into Boundary, Control, and Entity objects to manage the flow of data and logic.

Scroll all the way down for project outline

## Objects and Their Roles

### Entity Objects

Entity objects represent the core business data and operations related to those entities. They store data and perform business logic related to that data. They do not interact directly with the user.

ProductEntity: Represents product information such as price and features. It handles product-related data (e.g., retrieving the current price).

DateEntity: Handles date and availability logic for booking or checking availability of services.

AccountEntity: Manages user login credentials for websites like BestBuy or eBay.

TrackingHistoryEntity: Stores and tracks historical data on product prices. Helps to compare past prices with current ones.

BrowserEntity: Manages the state of the browser (e.g., if the browser is running, whether it's in

incognito mode, etc.).

NotificationEntity: Handles user preferences for receiving notifications, such as when prices change or product availability is updated.

Control Objects

Control objects are responsible for handling the logic of each use case. They interact with entity objects to manage data and handle business rules. Control objects execute the steps required to fulfill a use case.

HelpControl: Provides a list of commands available to the user.

ChatControl: Handles basic user interaction, such as greetings and responses to basic phrases like "hi" or "hello."

LoginControl: Manages the process of logging into a website, including retrieving login credentials from the database and passing them to the browser.

BrowserControl: Manages the launch and setup of the browser, including handling incognito mode and configuring the browser.

CloseBrowserControl: Handles the logic for closing the browser when requested by the user.

NavigationControl: Manages the process of navigating to a specific URL in the browser.

ProductTrackingControl: Manages the tracking of a product's price over time, scheduling regular price checks.

ProductControl: Checks the current price of a product and retrieves relevant product data.

AvailabilityControl: Handles checking the availability of a product or service based on user-provided dates.

StopTrackingControl: Stops the tracking process for a product or service.

NotificationControl: Monitors for changes in tracked products and sends notifications when a price or availability change occurs.

DataExtractionControl: Manages the extraction of tracking data, exporting it to Excel or HTML files.

BotControl: Manages the overall lifecycle of the Discord bot, including starting, stopping, and managing the registration of commands.

Boundary Objects

Boundary objects serve as the bridge between the user (or external actor) and the system. They collect data from the user and forward it to the appropriate control object. Boundary objects are responsible for interacting with the actor but not for executing business logic.

HelpBoundary: Collects the user?s help request and forwards it to HelpControl.

ChatBoundary: Receives chat commands from the user and forwards them to ChatControl.

LoginBoundary: Collects login credentials from the user and forwards them to LoginControl.

BrowserBoundary: Receives commands to launch the browser and forwards them to BrowserControl.

CloseBrowserBoundary: Receives the user?s request to close the browser and forwards it to CloseBrowserControl.

NavigationBoundary: Receives URL input from the user and forwards it to NavigationControl.

ProductTrackingBoundary: Collects the user?s request to track a product and forwards it to ProductTrackingControl.

ProductBoundary: Receives the user?s request to check a product price and forwards it to ProductControl.

AvailabilityBoundary: Collects the user?s availability check request and forwards it to AvailabilityControl.

StopTrackingBoundary: Receives the user?s request to stop tracking a product and forwards it to StopTrackingControl.

NotificationBoundary: Collects user preferences for receiving notifications and forwards them to NotificationControl.

DataExtractionBoundary: Collects the user?s request to export data and forwards it to DataExtractionControl.

StopBoundary: Receives the request to stop the bot and forwards it to BotControl.

Capabilities

Here?s what the bot can do:

1. !project_help

Description: Provides a list of available commands the user can issue.

Objects Involved:

Boundary: HelpBoundary

Control: HelpControl

Interaction: HelpBoundary collects the user?s help request and forwards it to HelpControl, which responds with the list of commands.

2. !chat_with_bot

Description: Responds to simple greetings (e.g., "hi", "hello") and provides a welcome message.

Objects Involved:

Boundary: ChatBoundary

Control: ChatControl

Interaction: ChatBoundary collects chat input and forwards it to ChatControl, which sends back a predefined response.

3. !login_to_website

Description: Logs into a website using stored credentials (e.g., BestBuy).

Objects Involved:

Boundary: LoginBoundary

Control: LoginControl, BrowserControl, NavigationControl

Entity: AccountEntity

Interaction:

LoginBoundary collects login credentials and forwards them to LoginControl.

LoginControl works with BrowserControl to launch the browser.

NavigationControl navigates to the website's login page.

AccountEntity retrieves the stored credentials from the database and logs the user in.

4. !launch_browser

Description: Launches the browser, optionally in incognito mode.

Objects Involved:

Boundary: BrowserBoundary

Control: BrowserControl

Entity: BrowserEntity

Interaction: BrowserBoundary collects the user's request to launch the browser and sends it to BrowserControl. BrowserControl uses BrowserEntity to configure and launch the browser.

5. !close_browser

Description: Closes the currently open browser session.

Objects Involved:

Boundary: CloseBrowserBoundary

Control: CloseBrowserControl

Entity: BrowserEntity

Interaction: CloseBrowserBoundary forwards the user?s request to CloseBrowserControl, which then tells BrowserEntity to close the browser session.

6. !navigate_to_website

Description: Navigates to a specific website URL in the browser.

Objects Involved:

Boundary: NavigationBoundary

Control: NavigationControl

Entity: BrowserEntity

Interaction: NavigationBoundary collects the URL input from the user and forwards it to NavigationControl. NavigationControl instructs BrowserEntity to navigate to the specified URL.

7. !track_price

Description: Tracks the price of a product over time and sends notifications if the price changes.

Objects Involved:

Boundary: ProductTrackingBoundary

Control: ProductTrackingControl, ProductControl, NotificationControl

Entity: ProductEntity, TrackingHistoryEntity, NotificationEntity

Interaction:

ProductTrackingBoundary collects the product URL from the user.

ProductTrackingControl initiates price tracking and uses ProductControl to fetch the current price.

The current price is stored in TrackingHistoryEntity.

If there?s a price change, NotificationControl sends an alert via NotificationEntity.

8. !check_price

Description: Manually checks the current price of a product.

Objects Involved:

Boundary: ProductBoundary

Control: ProductControl

Entity: ProductEntity

Interaction: ProductBoundary collects the product information from the user, and ProductControl retrieves the current price using ProductEntity.

9. !check_availability

Description: Checks the availability of a product or service on a specific date.

Objects Involved:

Boundary: AvailabilityBoundary

Control: AvailabilityControl

Entity: DateEntity

Interaction: AvailabilityBoundary collects the date and product/service details. AvailabilityControl checks the availability via DateEntity.

10. !stop_tracking

Description: Stops tracking the price or availability of a product.

Objects Involved:

Boundary: StopTrackingBoundary

Control: StopTrackingControl

Entity: TrackingHistoryEntity

Interaction: StopTrackingBoundary collects the stop request from the user. StopTrackingControl stops the tracking and updates TrackingHistoryEntity.

11. !receive_notifications

Description: Sends notifications when there?s a change in price or availability for tracked products/services.

Objects Involved:

Boundary: NotificationBoundary

Control: NotificationControl

Entity: NotificationEntity, TrackingHistoryEntity

Interaction: NotificationBoundary collects the user?s preferences for receiving notifications. NotificationControl monitors for changes and uses NotificationEntity to send alerts when changes occur.

12. !extract_data

Description: Extracts the tracked product data and exports it to Excel or HTML format.

Objects Involved:

Boundary: DataExtractionBoundary

Control: DataExtractionControl

Entity: TrackingHistoryEntity

Utilities: ExcelUtils, HTMLUtils

Interaction: DataExtractionBoundary collects the user?s request for data extraction. DataExtractionControl retrieves data from TrackingHistoryEntity and uses ExcelUtils or HTMLUtils to export the data to the desired format.

13. !stop

Description: Stops the Discord bot from running.

Objects Involved:

Boundary: StopBoundary

Control: BotControl

Interaction: StopBoundary collects the stop command from the user and forwards it to BotControl, which gracefully stops the bot.

DiscordBotProject_CISC699/

?

??? boundary/

?   ??? AccountBoundary.py

?   ??? HelpBoundary.py

?   ??? ChatBoundary.py

?   ??? LoginBoundary.py

?   ??? BrowserBoundary.py

?   ??? CloseBrowserBoundary.py

?   ??? NavigationBoundary.py

?   ??? ProductTrackingBoundary.py

?   ??? ProductBoundary.py

?   ??? AvailabilityBoundary.py

?   ??? StopTrackingBoundary.py

?   ??? NotificationBoundary.py

?   ??? DataExtractionBoundary.py

?   ??? StopBoundary.py

?

??? control/

?   ??? AccountControl.py

?   ??? HelpControl.py

?   ??? ChatControl.py

?   ??? LoginControl.py

?   ??? BrowserControl.py

?   ??? CloseBrowserControl.py

?   ??? NavigationControl.py

?   ??? ProductTrackingControl.py

?   ??? ProductControl.py

?   ??? AvailabilityControl.py

?   ??? StopTrackingControl.py

?   ??? NotificationControl.py

?   ??? DataExtractionControl.py

?   ??? BotControl.py

?

??? entity/

?   ??? ProductEntity.py

?   ??? DateEntity.py

?   ??? AccountEntity.py

?   ??? TrackingHistoryEntity.py

?   ??? BrowserEntity.py

?   ??? NotificationEntity.py

?

??? utils/

?   ??? ExcelUtils.py

?   ??? HTMLUtils.py

?   ??? DiscordUtils.py

?

??? test/

?   ??? test_addAccount.py

?   ??? test_deleteAccount.py

?   ??? test_fetchAccounts.py

?   ??? test_excel_creation.py

?   ??? test_html_creation.py

?

??? Config.py

??? main.py

??? project.txt

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\other\project_structure.py ---

```
import os

def list_files_and_folders(directory, output_file):
    with open(output_file, 'w') as f:
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            f.write(f"Directory: {root}\n")
            for dir_name in dirs:
                f.write(f"  Folder: {dir_name}\n")
```

```
            for file_name in files:
                f.write(f"  File: {file_name}\n")

# Update the directory path to your project folder
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699"
output_file = os.path.join(project_directory, "project_structure.txt")

# Call the function to list files and save output to .txt
list_files_and_folders(project_directory, output_file)

print(f"File structure saved to {output_file}")
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\other\project_structure.txt ---
Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699
  Folder: boundary
  Folder: control
  Folder: entity
  Folder: test
  Folder: utils
  File: Config.py
  File: main.py
  File: project.txt
  File: project_structure.txt
  File: temporary.py

File: Tests_URLs.txt

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\boundary

File: AccountBoundary.py

File: AvailabilityBoundary.py

File: BotBoundary.py

File: BrowserBoundary.py

File: CloseBrowserBoundary.py

File: DataExtractionBoundary.py

File: HelpBoundary.py

File: LoginBoundary.py

File: NavigationBoundary.py

File: NotificationBoundary.py

File: ProductBoundary.py

File: ProductTrackingBoundary.py

File: StopBoundary.py

File: StopTrackingBoundary.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\control

File: AccountControl.py

File: AvailabilityControl.py

File: BotControl.py

File: BrowserControl.py

File: ChatControl.py

File: CloseBrowserControl.py

File: DataExtractionControl.py

File: HelpControl.py

File: LoginControl.py

File: NavigationControl.py

File: NotificationControl.py

File: ProductControl.py

File: ProductTrackingControl.py

File: StopTrackingControl.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\entity

File: AccountEntity.py

File: BrowserEntity.py

File: DateEntity.py

File: NotificationEntity.py

File: ProductEntity.py

File: TrackingHistoryEntity.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\test

File: test_addAccount.py

File: test_deleteAccount.py

File: test_excel_creation.py

File: test_fetchAccounts.py

File: test_html_creation.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\utils

File: css_selectors.py

File: DiscordUtils.py

File: ExcelUtils.py

File: HTMLUtils.py

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\other\temporary.py ---

```
import os

def write_project_files_to_txt(directory, output_file):
    with open(output_file, 'w', encoding='utf-8') as out_file:
        # Walk through the directory
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            for file in files:
                file_path = os.path.join(root, file)
                if file.endswith('.py') or file.endswith('.txt') or file.endswith('.json'):
                    out_file.write(f"\n--- {file_path} ---\n")
                    try:
                        with open(file_path, 'r', encoding='utf-8') as f:
                            out_file.write(f.read())
                    except Exception as e:
                        out_file.write(f"\nError reading {file}: {e}\n")

# Directory and output file path
```

directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699"
output_file = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\project_summary.txt"

write_project_files_to_txt(directory, output_file)

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\other\Tests_URLs.txt ---
database password: postgres

Working Commands: Test commands

!project_help

!login bestbuy

!launch_browser

!close_browser

!navigate_to_website https://www.google.com/

!monitor_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price

https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

!check_availability https://www.opentable.com/r/bar-spero-washington/

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22"

!stop_monitoring

!stop

**********

Working on it:

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22" "8:00 PM"

**********

URLs to Test:

https://www.opentable.com/r/bar-spero-washington/

https://www.ebay.com/itm/314411766963?_trkparms=amclksrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3Avlp_homepage

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test\test_addAccount.py ---

```
import sys, os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_add_account():
    account_control = AccountControl()

    # Adding a new account
    account_control.add_account("newUser", "newPassword123", "newWebsite")

if __name__ == "__main__":
    test_add_account()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test\test_deleteAccount.py ---

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_delete_account():
    account_control = AccountControl()

    account_control.delete_account(4)

if __name__ == "__main__":
    test_delete_account()
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test\test_excel_creation.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test\test_fetchAccounts.py ---

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl
```

```
def test_fetch_accounts():
    account_control = AccountControl()

    # Fetching all accounts
    account_control.fetch_accounts()

def test_fetch_account_by_website(website):
    account_control = AccountControl()

    # Fetch the account by website directly
    account = account_control.fetch_account_by_website(website)

    if account:
        username, password = account  # Unpack the returned tuple
        print(f"Website: {website}, Username: {username}, Password: {password}")
    else:
        print(f"No account found for website: {website}")

if __name__ == "__main__":
    test_fetch_accounts()
    test_fetch_account_by_website("ebay")
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test\test_html_creation.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\test__init__.py ---

```
#empty init file

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\utils\css_selectors.py ---

class Selectors:
    SELECTORS = {
        "trendyol": {
            "price": ".featured-prices .prc-dsc"  # Selector for Trendyol price
        },
        "ebay": {
            "url": "https://signin.ebay.com/signin/",
            "email_field": "#userid",
            "continue_button": "[data-testid*='signin-continue-btn']",
            "password_field": "#pass",
            "login_button": "#sgnBt",
            "price": ".x-price-primary span"  # CSS selector for Ebay price
        },
        "bestbuy": {
            "url": "https://www.bestbuy.com/signin/",
            "email_field": "#fld-e",
            #"continue_button": ".cia-form__controls  button",
            "password_field": "#fld-p1",
            "SignIn_button": ".cia-form__controls  button",
            "price": "[data-testid='customer-price'] span",  # CSS selector for BestBuy price
            "homePage": ".v-p-right-xxs.line-clamp"
        },
        "opentable": {
            "url": "https://www.opentable.com/",
```

```
            "date_field": "#restProfileSideBarDtpDayPicker-label",
            "time_field": "#restProfileSideBartimePickerDtpPicker",
            "find_table_button": ".find-table-button",  # Example selector for the Find Table button
            "availability_result": ".availability-result",  # Example selector for availability results
                "show_next_available_button": "button[data-test='multi-day-availability-button']",  # Show next available button
            "available_dates": "ul[data-test='time-slots'] > li",  # Available dates and times
            "no_availability": "div._8ye6OVzeOuU- span"
        }
    }

    @staticmethod
    def get_selectors_for_url(url):
        for keyword, selectors in Selectors.SELECTORS.items():
            if keyword in url.lower():
                return selectors
        return None  # Return None if no matching selectors are found
```

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\utils\DiscordUtils.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\utils\ExcelUtils.py ---

--- D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699\utils\HTMLUtils.py ---

```
--- AccountBoundary.py ---
from discord.ext import commands
from control.AccountControl import AccountControl

class AccountBoundary(commands.Cog):

    def __init__(self, bot):
        self.bot = bot
        self.account_control = AccountControl()


    @commands.command(name='fetch_accounts')
    async def fetch_accounts(self, ctx):
        """Fetch and display all accounts."""
        accounts = self.account_control.fetch_accounts()

        # Send each account or the no accounts message to Discord
        for account in accounts:
            await ctx.send(account)



    @commands.command(name="add_account")
    async def add_account(self, ctx, username: str, password: str):
        """Add a new user account to the database."""
        result = self.account_control.add_account(username, password)
        if result:
```

```
            await ctx.send(f"Account for {username} added successfully.")
        else:
            await ctx.send(f"Failed to add account for {username}.")



    @commands.command(name="delete_account")
    async def delete_account(self, ctx, user_id: int):
        """Delete a user account from the database."""
        result = self.account_control.delete_account(user_id)
        if result:
            await ctx.send(f"Account with ID {user_id} deleted successfully.")
        else:
            await ctx.send(f"Failed to delete account with ID {user_id}.")
```

--- AvailabilityBoundary.py ---

--- BotBoundary.py ---

```
from discord.ext import commands
from control.ChatControl import ChatControl
from Config import Config

class BotBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
```

```
    self.chat_control = ChatControl()

@commands.Cog.listener()
async def on_ready(self):
    """Bot startup message when ready."""
    print(f'Logged in as {self.bot.user.name}')
    channel = self.bot.get_channel(Config.CHANNEL_ID)
    if channel:
        await channel.send("Hi, I'm online!")

@commands.Cog.listener()
async def on_message(self, message):
    """Handle non-prefixed messages and command-prefixed messages."""
    if message.author == self.bot.user:
        return

    # Handle non-prefixed messages (like greetings)
    if not message.content.startswith('!'):
        response = self.chat_control.process_non_prefixed_message(message.content)
        await message.channel.send(response)

@commands.Cog.listener()
async def on_command_error(self, ctx, error):
    """Handle unrecognized commands."""
    if isinstance(error, commands.CommandNotFound):
        # Handle unknown command
        response = self.chat_control.handle_unrecognized_command()
```

```
            await ctx.send(response)



--- BrowserBoundary.py ---
from discord.ext import commands
from control.BrowserControl import BrowserControl

class BrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.browser_control = BrowserControl()

    @commands.command(name='launch_browser')
    async def launch_browser(self, ctx, *args):
        """Command to launch the browser."""
        incognito = "incognito" in args
        response = self.browser_control.launch_browser(ctx.author, incognito)
        await ctx.send(response)



--- CloseBrowserBoundary.py ---
from discord.ext import commands
from control.CloseBrowserControl import CloseBrowserControl

class CloseBrowserBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
```

```
        self.close_browser_control = CloseBrowserControl()

    @commands.command(name='close_browser')
    async def close_browser(self, ctx):
        """Command to close the browser."""
        response = self.close_browser_control.close_browser()
        await ctx.send(response)
```

--- DataExtractionBoundary.py ---

--- HelpBoundary.py ---

```
from discord.ext import commands
from control.HelpControl import HelpControl

class HelpBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.help_control = HelpControl()

    @commands.command(name='project_help')
    async def project_help(self, ctx):
        """Handles the project_help command."""
        help_message = self.help_control.get_help_message()
        await ctx.send(help_message)
```

--- LoginBoundary.py ---

```
from discord.ext import commands
from control.LoginControl import LoginControl


class LoginBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.login_control = LoginControl()

    @commands.command(name='login')
    async def login(self, ctx, site: str, *args):
        """Command to log into a website using stored credentials."""
        incognito = "incognito" in args
        retries = next((int(arg) for arg in args if arg.isdigit()), 1)
        response = await self.login_control.login(site, incognito, retries)
        await ctx.send(response)
```

--- MonitorPriceBoundary.py ---

```
from discord.ext import commands
from control.MonitorPriceControl import MonitorPriceControl


class MonitorPriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.monitor_price_control = MonitorPriceControl()
```

```
    @commands.command(name='monitor_price')
    async def monitor_price(self, ctx, url: str, frequency: int = 1):
        """Command to monitor the price at regular intervals."""
        await self.monitor_price_control.monitor_price(ctx, url, frequency)
```

--- NavigationBoundary.py ---

```
from discord.ext import commands
from control.NavigationControl import NavigationControl

class NavigationBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.navigation_control = NavigationControl()

    @commands.command(name='navigate_to_website')
    async def navigate_to_website(self, ctx, url: str):
        """Command to navigate to a specified URL."""
        response = self.navigation_control.navigate_to_url(url)
        await ctx.send(response)
```

--- NotificationBoundary.py ---

--- PriceBoundary.py ---

```
from discord.ext import commands
from control.PriceControl import PriceControl

class PriceBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.price_control = PriceControl()

    @commands.command(name='get_price')
    async def get_price(self, ctx, url: str):
        """Command to get the price from the given URL."""
        response = await self.price_control.get_price(ctx, url)
        await ctx.send(response)
```

--- StopBoundary.py ---

```
from discord.ext import commands
from control.BotControl import BotControl

class StopBoundary(commands.Cog):
    def __init__(self, bot):
        self.bot = bot
        self.bot_control = BotControl(bot)

    @commands.command(name="stop_bot")
    async def stop_bot(self, ctx):
        """Handles the stop command and gracefully shuts down the bot."""
```

```
        await ctx.send("Stopping the bot...")
        await self.bot_control.stop_bot()
```

--- StopMonitoringBoundary.py ---

--- __init__.py ---

```
#empty init file
```

--- AccountControl.py ---

```
from entity.AccountEntity import AccountEntity

class AccountControl:
    def __init__(self):
        self.account_entity = AccountEntity()

    def add_account(self, username, password, webSite):
        self.account_entity.connect()
        self.account_entity.add_account(username, password, webSite)
        self.account_entity.close()

    def fetch_accounts(self):
        """Fetch all accounts and return them."""
        self.account_entity.connect()
        accounts = self.account_entity.fetch_accounts()
```

```
        if accounts:
            account_messages = []
            for account in accounts:
                message = f"ID: {account[0]}, Username: {account[1]}, Password: {account[2]}, Website:
{account[3]}"
                print(message)  # For terminal output
                account_messages.append(message)
            self.account_entity.close()
            return account_messages
        else:
            print("No accounts found.")  # For terminal output
            self.account_entity.close()
            return ["No accounts found."]



    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        self.account_entity.connect()
        account = self.account_entity.fetch_account_by_website(website)  # Call the entity method
        self.account_entity.close()
        return account



    def delete_account(self, account_id):
        self.account_entity.connect()
        self.account_entity.delete_account(account_id)
        self.account_entity.reset_id_sequence()
```

```
        self.account_entity.close()



--- AvailabilityControl.py ---



--- BotControl.py ---
import asyncio

class BotControl:
    def __init__(self, bot):
        self.bot = bot

    async def send_greeting(self):
        """Sends a greeting when the bot comes online."""
        channel = self.bot.get_channel(self.bot.config.CHANNEL_ID)
        if channel:
            await channel.send("Hi, I'm online! type '!project_help' to see what I can do")

    async def stop_bot(self):
        """Stops the bot gracefully, ensuring all connections are closed."""
        print("Bot is stopping...")

        await self.bot.close()

--- BrowserControl.py ---
from entity.BrowserEntity import BrowserEntity
```

```
from control.AccountControl import AccountControl  # Use AccountControl for consistency

class BrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()  # Use AccountControl to fetch credentials

    def launch_browser(self, user, incognito=False):
        return self.browser_entity.launch_browser(incognito=incognito, user=user)
```

--- ChatControl.py ---

```
# ChatControl in control/ChatControl.py
class ChatControl:
    def process_non_prefixed_message(self, message):
        """Process non-prefixed messages like 'hi', 'hello'."""
        if message.lower() in ["hi", "hello"]:
            return "Hello! How can I assist you today? Type !project_help for assistance."
        else:
            return "I didn't recognize that. Type !project_help to see available commands."

    def handle_unrecognized_command(self):
        """Handle unrecognized command from on_command_error."""
        return "I didn't recognize that command. Type !project_help for assistance."
```

--- CloseBrowserControl.py ---

```
from entity.BrowserEntity import BrowserEntity


class CloseBrowserControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def close_browser(self):
        return self.browser_entity.close_browser()
```

--- DataExtractionControl.py ---

--- HelpControl.py ---

```
class HelpControl:
    def get_help_message(self):
        """Returns a list of available bot commands."""
        return (
            "Here are the available commands:\n"
            "!project_help - Get help on available commands.\n"
            "!chat_with_bot - Say hi to the bot.\n"
            "!login_to_website - Log in to a website.\n"
            "!launch_browser - Launch the browser.\n"
            "!close_browser - Close the browser.\n"
            "!navigate_to_website - Navigate to a website.\n"
            "!track_price - Track a product price.\n"
            "!check_price - Check the price of a product.\n"
```

```
            "!check_availability - Check the availability of a product.\n"
            "!stop_tracking - Stop tracking a product.\n"
            "!receive_notifications - Receive notifications for price changes.\n"
            "!extract_data - Export data to Excel or HTML.\n"
            "!stop - Stop the bot.\n"
        )



--- LoginControl.py ---
from entity.BrowserEntity import BrowserEntity
from control.AccountControl import AccountControl

class LoginControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()
        self.account_control = AccountControl()

    async def login(self, site, incognito=False, retries=1):
        # Fetch credentials using AccountControl
        account = self.account_control.fetch_account_by_website(site)
        if account:
            username, password = account
            return await self.browser_entity.login(site, username, password, incognito, retries)
        else:
            return f"No account found for website {site}"
```

```
--- MonitorPriceControl.py ---
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging


class MonitorPriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("MonitorPriceControl")


    async def monitor_price(self, ctx, url, frequency=1):
        """Monitor the price at a given interval."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                await ctx.send(f"Monitoring price every {frequency} minute(s).")
                previous_price = None


                while True:
                    current_price = self.price_entity.get_price(url)
                    if current_price:
                        if previous_price is None:
                            await ctx.send(f"Starting price monitoring. Current price is: {current_price}")
                        else:
                            if current_price > previous_price:
                                    await ctx.send(f"Price went up! Current price: {current_price} (Previous:
{previous_price})")
```

```
                    elif current_price < previous_price:
                        await ctx.send(f"Price went down! Current price: {current_price} (Previous: {previous_price})")
                    else:
                        await ctx.send(f"Price remains the same: {current_price}")
                    previous_price = current_price
                else:
                    await ctx.send("Failed to retrieve the price.")
                await asyncio.sleep(frequency * 60)  # Wait for the next check
            except Exception as e:
                self.logger.error(f"Failed to monitor price for {url}: {e}")
                await ctx.send(f"Failed to monitor price: {e}")
        else:
            await ctx.send("This command can only be used in the designated channel.")
```

--- NavigationControl.py ---

```
from entity.BrowserEntity import BrowserEntity

class NavigationControl:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def navigate_to_url(self, url):
        """Navigate to a specific URL."""
        return self.browser_entity.navigate_to_url(url)
```

--- NotificationControl.py ---

--- PriceControl.py ---

```
import asyncio
from entity.PriceEntity import PriceEntity
from Config import Config
import logging

class PriceControl:
    def __init__(self):
        self.price_entity = PriceEntity()
        self.logger = logging.getLogger("PriceControl")

    async def get_price(self, ctx, url):
        """Fetch the current price from the given URL."""
        if ctx.channel.id == Config.CHANNEL_ID:
            try:
                price = self.price_entity.get_price(url)
                if price:
                    return f"The current price is: {price}"
                else:
                    return "Failed to retrieve the price."
            except Exception as e:
                self.logger.error(f"Failed to get price for {url}: {e}")
                return f"Error getting price: {e}"
```

```
        else:
            return "This command can only be used in the designated channel."
```

--- StopMonitoringControl.py ---

--- __init__.py ---

```
#empty init file
```

--- AccountEntity.py ---

```
import psycopg2
from Config import Config

class AccountEntity:
    def __init__(self):
        self.dbname = "postgres"
        self.user = "postgres"
        self.host = "localhost"
        self.port = "5432"
        self.password = Config.DATABASE_PASSWORD

    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
```

```
            password=self.password,
            host=self.host,
            port=self.port
        )
        self.cursor = self.connection.cursor()
        print("Database Connection Established.")
    except Exception as error:
        print(f"Error connecting to the database: {error}")
        self.connection = None
        self.cursor = None

def add_account(self, username, password, webSite):
    """Insert a new account into the accounts table."""
    try:
        if self.cursor:
            self.cursor.execute("INSERT INTO accounts (username, password, website) VALUES
(%s, %s, %s)", (username, password, webSite))
            self.connection.commit()
            print(f"Account {username} added successfully.")
    except Exception as error:
        print(f"Error inserting account: {error}")

def fetch_accounts(self):
    """Fetch all accounts from the accounts table."""
    try:
        if self.cursor:
            self.cursor.execute("SELECT * FROM accounts;")
```

```
            accounts = self.cursor.fetchall()
            return accounts
        except Exception as error:
            print(f"Error fetching accounts: {error}")
            return None

    def delete_account(self, account_id):
        """Delete an account by ID."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT * FROM accounts WHERE id = %s", (account_id,))
                account = self.cursor.fetchone()
                if account:
                    self.cursor.execute("DELETE FROM accounts WHERE id = %s", (account_id,))
                    self.connection.commit()
                    print(f"Account with ID {account_id} deleted successfully.")
                else:
                    print(f"Account with ID {account_id} not found. No deletion performed.")
        except Exception as error:
            print(f"Error deleting account: {error}")


    def fetch_account_by_website(self, website):
        """Fetch the username and password where the website matches."""
        try:
                        self.cursor.execute("SELECT username, password FROM accounts WHERE
LOWER(website) = LOWER(%s)", (website,))
```

```
            return self.cursor.fetchone()  # Returns one matching account
        except Exception as error:
            print(f"Error fetching account for website {website}: {error}")
            return None




    def reset_id_sequence(self):
        """Reset the account ID sequence to the next available value."""
        try:
            if self.cursor:
                self.cursor.execute("SELECT COALESCE(MAX(id), 0) + 1 FROM accounts")
                next_id = self.cursor.fetchone()[0]
                        self.cursor.execute("ALTER SEQUENCE accounts_id_seq RESTART WITH %s",
(next_id,))
                self.connection.commit()
                print(f"ID sequence reset to {next_id}.")
        except Exception as error:
            print(f"Error resetting ID sequence: {error}")


    def close(self):
        """Close the database connection."""
        if self.cursor:
            self.cursor.close()
        if self.connection:
            self.connection.close()
            print("Database Connection closed.")
```

```
--- BrowserEntity.py ---
import asyncio
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from utils.css_selectors import Selectors  # Import CSS selectors for the website

class BrowserEntity:
    _instance = None  # Singleton instance

    def __new__(cls, *args, **kwargs):
        if cls._instance is None:
            cls._instance = super(BrowserEntity, cls).__new__(cls)
            cls._instance.driver = None  # Initialize driver to None
        return cls._instance

    def launch_browser(self, incognito=False, user=None):
        if self.driver:
            print("Browser is already running. No need to launch a new one.")
            return "Browser is already running."

        try:
            # Special launch options as per your original implementation
```

```
            options = webdriver.ChromeOptions()

            # Add options to avoid crashing and improve performance
            options.add_argument("--remote-debugging-port=9222")
            options.add_experimental_option("excludeSwitches", ["enable-automation"])
            options.add_experimental_option('useAutomationExtension', False)
            options.add_argument("--start-maximized")
            options.add_argument("--disable-notifications")
            options.add_argument("--disable-popup-blocking")
            options.add_argument("--disable-infobars")
            options.add_argument("--disable-extensions")
            options.add_argument("--disable-webgl")
            options.add_argument("--disable-webrtc")
            options.add_argument("--disable-rtc-smoothing")

            if incognito:
                options.add_argument("--incognito")

            self.driver = webdriver.Chrome(service=Service(), options=options)
            success_message = "Chrome browser launched successfully in incognito mode." if incognito
else "Chrome browser launched successfully."
            print(f"Driver initialized: {self.driver}")  # Debug: Print the driver
            return success_message
        except Exception as e:
            error_message = f"Failed to launch browser: {e}"
            print(error_message)
            raise
```

```
def navigate_to_url(self, url):
    if not self.driver:
        print("Driver is not initialized, launching browser first.")  # Debug
        self.launch_browser()
    try:
        self.driver.get(url)
        return f"Navigated to URL: {url}"
    except Exception as e:
        raise


def close_browser(self):
    print(f"Closing browser. Current driver: {self.driver}")  # Debug: Check the driver status
    if self.driver:
        self.driver.quit()  # Close the browser session
        self.driver = None  # Set to None after closing
        print("Browser closed successfully.")
        return "Browser closed successfully."
    else:
        print("No browser is currently open.")
        return "No browser is currently open."



async def login(self, site, username, password, incognito=False, retries=1):
    # Get the URL and selectors from css_selectors
    url = Selectors.get_selectors_for_url(site)['url']
    for attempt in range(retries):
```

```
        try:
            self.navigate_to_url(url)
            await asyncio.sleep(3)

            # Enter the email address
                                        email_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['email_field'])
            email_field.click()
            email_field.send_keys(username)
            await asyncio.sleep(3)

            # Enter the password
                                    password_field = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['password_field'])
            password_field.click()
            password_field.send_keys(password)
            await asyncio.sleep(3)

            # Click the login button
                                    sign_in_button = self.driver.find_element(By.CSS_SELECTOR,
Selectors.get_selectors_for_url(site)['SignIn_button'])
            sign_in_button.click()
            await asyncio.sleep(5)

            # Wait for the homepage to load after login
            WebDriverWait(self.driver, 30).until(
                                            EC.presence_of_element_located((By.CSS_SELECTOR,
```

```
Selectors.get_selectors_for_url(site)['homePage'])))

                return f"Logged in to {url} successfully with username: {username}"
            except Exception as e:
                if attempt < retries - 1:
                    await asyncio.sleep(3)
                else:
                    raise e



--- DateEntity.py ---



--- NotificationEntity.py ---



--- PriceEntity.py ---
import time
from selenium.webdriver.common.by import By
from utils.css_selectors import Selectors
from entity.BrowserEntity import BrowserEntity  # Import the browser interaction logic

class PriceEntity:
    def __init__(self):
        self.browser_entity = BrowserEntity()

    def get_price(self, url):
```

```
        """Fetch the price from the provided URL using CSS selectors."""
        selectors = Selectors.get_selectors_for_url(url)
        if not selectors:
            raise ValueError(f"No selectors found for URL: {url}")

        # Navigate to the URL using the browser entity
        self.browser_entity.navigate_to_url(url)
        time.sleep(2)  # Wait for the page to load

        try:
            # Use the CSS selector to find the price on the page
                        price_element = self.browser_entity.driver.find_element(By.CSS_SELECTOR,
selectors['price'])
            price = price_element.text
            print(f"Price found: {price}")
            return price
        except Exception as e:
            print(f"Error finding price: {e}")
            return None
```

--- PriceHistoryEntity.py ---

--- __init__.py ---

```
#empty init file
```

--- project.txt ---

DiscordBotProject_CISC699 - Project Overview

Introduction

This project is a Discord bot designed to perform various tasks, including tracking product prices, checking availability, logging into websites, and exporting data.

The bot interacts with users via commands sent through Discord and responds based on the requested use case.

The project follows a clear structure, adhering to software engineering best practices, and separates the logic into Boundary, Control, and Entity objects to manage the flow of data and logic.

Scroll all the way down for project outline

Objects and Their Roles

Entity Objects

Entity objects represent the core business data and operations related to those entities. They store data and perform business logic related to that data. They do not interact directly with the user.

ProductEntity: Represents product information such as price and features. It handles product-related data (e.g., retrieving the current price).

DateEntity: Handles date and availability logic for booking or checking availability of services.

AccountEntity: Manages user login credentials for websites like BestBuy or eBay.

TrackingHistoryEntity: Stores and tracks historical data on product prices. Helps to compare past prices with current ones.

BrowserEntity: Manages the state of the browser (e.g., if the browser is running, whether it's in incognito mode, etc.).

NotificationEntity: Handles user preferences for receiving notifications, such as when prices change or product availability is updated.

Control Objects

Control objects are responsible for handling the logic of each use case. They interact with entity objects to manage data and handle business rules. Control objects execute the steps required to fulfill a use case.

HelpControl: Provides a list of commands available to the user.

ChatControl: Handles basic user interaction, such as greetings and responses to basic phrases like "hi" or "hello."

LoginControl: Manages the process of logging into a website, including retrieving login credentials from the database and passing them to the browser.

BrowserControl: Manages the launch and setup of the browser, including handling incognito mode and configuring the browser.

CloseBrowserControl: Handles the logic for closing the browser when requested by the user.

NavigationControl: Manages the process of navigating to a specific URL in the browser.

ProductTrackingControl: Manages the tracking of a product's price over time, scheduling regular

price checks.

ProductControl: Checks the current price of a product and retrieves relevant product data.

AvailabilityControl: Handles checking the availability of a product or service based on user-provided dates.

StopTrackingControl: Stops the tracking process for a product or service.

NotificationControl: Monitors for changes in tracked products and sends notifications when a price or availability change occurs.

DataExtractionControl: Manages the extraction of tracking data, exporting it to Excel or HTML files.

BotControl: Manages the overall lifecycle of the Discord bot, including starting, stopping, and managing the registration of commands.

Boundary Objects

Boundary objects serve as the bridge between the user (or external actor) and the system. They collect data from the user and forward it to the appropriate control object. Boundary objects are responsible for interacting with the actor but not for executing business logic.

HelpBoundary: Collects the user?s help request and forwards it to HelpControl.

ChatBoundary: Receives chat commands from the user and forwards them to ChatControl.

LoginBoundary: Collects login credentials from the user and forwards them to LoginControl.

BrowserBoundary: Receives commands to launch the browser and forwards them to BrowserControl.

CloseBrowserBoundary: Receives the user?s request to close the browser and forwards it to CloseBrowserControl.

NavigationBoundary: Receives URL input from the user and forwards it to NavigationControl.

ProductTrackingBoundary: Collects the user?s request to track a product and forwards it to ProductTrackingControl.

ProductBoundary: Receives the user?s request to check a product price and forwards it to ProductControl.

AvailabilityBoundary: Collects the user?s availability check request and forwards it to AvailabilityControl.

StopTrackingBoundary: Receives the user?s request to stop tracking a product and forwards it to StopTrackingControl.

NotificationBoundary: Collects user preferences for receiving notifications and forwards them to NotificationControl.

DataExtractionBoundary: Collects the user?s request to export data and forwards it to DataExtractionControl.

StopBoundary: Receives the request to stop the bot and forwards it to BotControl.

Capabilities

Here?s what the bot can do:

1. !project_help

Description: Provides a list of available commands the user can issue.

Objects Involved:

Boundary: HelpBoundary

Control: HelpControl

Interaction: HelpBoundary collects the user?s help request and forwards it to HelpControl, which responds with the list of commands.

2. !chat_with_bot

Description: Responds to simple greetings (e.g., "hi", "hello") and provides a welcome message.

Objects Involved:

Boundary: ChatBoundary

Control: ChatControl

Interaction: ChatBoundary collects chat input and forwards it to ChatControl, which sends back a predefined response.

3. !login_to_website

Description: Logs into a website using stored credentials (e.g., BestBuy).

Objects Involved:

Boundary: LoginBoundary

Control: LoginControl, BrowserControl, NavigationControl

Entity: AccountEntity

Interaction:

LoginBoundary collects login credentials and forwards them to LoginControl.

LoginControl works with BrowserControl to launch the browser.

NavigationControl navigates to the website's login page.

AccountEntity retrieves the stored credentials from the database and logs the user in.

4. !launch_browser

Description: Launches the browser, optionally in incognito mode.

Objects Involved:

Boundary: BrowserBoundary

Control: BrowserControl

Entity: BrowserEntity

Interaction: BrowserBoundary collects the user's request to launch the browser and sends it to BrowserControl. BrowserControl uses BrowserEntity to configure and launch the browser.

5. !close_browser

Description: Closes the currently open browser session.

Objects Involved:

Boundary: CloseBrowserBoundary

Control: CloseBrowserControl

Entity: BrowserEntity

Interaction: CloseBrowserBoundary forwards the user?s request to CloseBrowserControl, which then tells BrowserEntity to close the browser session.

6. !navigate_to_website

Description: Navigates to a specific website URL in the browser.

Objects Involved:

Boundary: NavigationBoundary

Control: NavigationControl

Entity: BrowserEntity

Interaction: NavigationBoundary collects the URL input from the user and forwards it to

NavigationControl. NavigationControl instructs BrowserEntity to navigate to the specified URL.

7. !track_price

Description: Tracks the price of a product over time and sends notifications if the price changes.

Objects Involved:

Boundary: ProductTrackingBoundary

Control: ProductTrackingControl, ProductControl, NotificationControl

Entity: ProductEntity, TrackingHistoryEntity, NotificationEntity

Interaction:

ProductTrackingBoundary collects the product URL from the user.

ProductTrackingControl initiates price tracking and uses ProductControl to fetch the current price.

The current price is stored in TrackingHistoryEntity.

If there?s a price change, NotificationControl sends an alert via NotificationEntity.

8. !check_price

Description: Manually checks the current price of a product.

Objects Involved:

Boundary: ProductBoundary

Control: ProductControl

Entity: ProductEntity

Interaction: ProductBoundary collects the product information from the user, and ProductControl retrieves the current price using ProductEntity.

9. !check_availability

Description: Checks the availability of a product or service on a specific date.

Objects Involved:

Boundary: AvailabilityBoundary

Control: AvailabilityControl

Entity: DateEntity

Interaction: AvailabilityBoundary collects the date and product/service details. AvailabilityControl

checks the availability via DateEntity.

10. !stop_tracking

Description: Stops tracking the price or availability of a product.

Objects Involved:

Boundary: StopTrackingBoundary

Control: StopTrackingControl

Entity: TrackingHistoryEntity

Interaction: StopTrackingBoundary collects the stop request from the user. StopTrackingControl stops the tracking and updates TrackingHistoryEntity.

11. !receive_notifications

Description: Sends notifications when there?s a change in price or availability for tracked products/services.

Objects Involved:

Boundary: NotificationBoundary

Control: NotificationControl

Entity: NotificationEntity, TrackingHistoryEntity

Interaction: NotificationBoundary collects the user?s preferences for receiving notifications. NotificationControl monitors for changes and uses NotificationEntity to send alerts when changes occur.

12. !extract_data

Description: Extracts the tracked product data and exports it to Excel or HTML format.

Objects Involved:

Boundary: DataExtractionBoundary

Control: DataExtractionControl

Entity: TrackingHistoryEntity

Utilities: ExcelUtils, HTMLUtils

Interaction: DataExtractionBoundary collects the user?s request for data extraction.

DataExtractionControl retrieves data from TrackingHistoryEntity and uses ExcelUtils or HTMLUtils to export the data to the desired format.

13. !stop

Description: Stops the Discord bot from running.

Objects Involved:

Boundary: StopBoundary

Control: BotControl

Interaction: StopBoundary collects the stop command from the user and forwards it to BotControl, which gracefully stops the bot.

DiscordBotProject_CISC699/

?

??? boundary/

?  ??? AccountBoundary.py

?  ??? HelpBoundary.py

?  ??? ChatBoundary.py

?  ??? LoginBoundary.py

?  ??? BrowserBoundary.py

?  ??? CloseBrowserBoundary.py

?  ??? NavigationBoundary.py

?  ??? ProductTrackingBoundary.py

?  ??? ProductBoundary.py

?  ??? AvailabilityBoundary.py

?  ??? StopTrackingBoundary.py

?  ??? NotificationBoundary.py

?   ??? DataExtractionBoundary.py

?   ??? StopBoundary.py

?

??? control/

?   ??? AccountControl.py

?   ??? HelpControl.py

?   ??? ChatControl.py

?   ??? LoginControl.py

?   ??? BrowserControl.py

?   ??? CloseBrowserControl.py

?   ??? NavigationControl.py

?   ??? ProductTrackingControl.py

?   ??? ProductControl.py

?   ??? AvailabilityControl.py

?   ??? StopTrackingControl.py

?   ??? NotificationControl.py

?   ??? DataExtractionControl.py

?   ??? BotControl.py

?

??? entity/

?   ??? ProductEntity.py

?   ??? DateEntity.py

?   ??? AccountEntity.py

?   ??? TrackingHistoryEntity.py

?   ??? BrowserEntity.py

?   ??? NotificationEntity.py

?

```
??? utils/
?   ??? ExcelUtils.py
?   ??? HTMLUtils.py
?   ??? DiscordUtils.py
?
??? test/
?   ??? test_addAccount.py
?   ??? test_deleteAccount.py
?   ??? test_fetchAccounts.py
?   ??? test_excel_creation.py
?   ??? test_html_creation.py
?
??? Config.py
??? main.py
??? project.txt



--- project_structure.py ---
import os

def list_files_and_folders(directory, output_file):
    with open(output_file, 'w') as f:
        for root, dirs, files in os.walk(directory):
            # Ignore .git and __pycache__ folders
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

            f.write(f"Directory: {root}\n")
```

```
        for dir_name in dirs:
            f.write(f"  Folder: {dir_name}\n")
        for file_name in files:
            f.write(f"  File: {file_name}\n")

# Update the directory path to your project folder
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699"
output_file = os.path.join(project_directory, "project_structure.txt")

# Call the function to list files and save output to .txt
list_files_and_folders(project_directory, output_file)

print(f"File structure saved to {output_file}")
```

--- project_structure.txt ---

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699

  Folder: boundary

  Folder: control

  Folder: entity

  Folder: test

  Folder: utils

  File: Config.py

  File: main.py

  File: project.txt

File: project_structure.txt

File: temporary.py

File: Tests_URLs.txt

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\boundary

File: AccountBoundary.py

File: AvailabilityBoundary.py

File: BotBoundary.py

File: BrowserBoundary.py

File: CloseBrowserBoundary.py

File: DataExtractionBoundary.py

File: HelpBoundary.py

File: LoginBoundary.py

File: NavigationBoundary.py

File: NotificationBoundary.py

File: ProductBoundary.py

File: ProductTrackingBoundary.py

File: StopBoundary.py

File: StopTrackingBoundary.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\control

File: AccountControl.py

File: AvailabilityControl.py

File: BotControl.py

File: BrowserControl.py

File: ChatControl.py

File: CloseBrowserControl.py

File: DataExtractionControl.py

File: HelpControl.py

File: LoginControl.py

File: NavigationControl.py

File: NotificationControl.py

File: ProductControl.py

File: ProductTrackingControl.py

File: StopTrackingControl.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\entity

File: AccountEntity.py

File: BrowserEntity.py

File: DateEntity.py

File: NotificationEntity.py

File: ProductEntity.py

File: TrackingHistoryEntity.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\test

File: test_addAccount.py

File: test_deleteAccount.py

File: test_excel_creation.py

File: test_fetchAccounts.py

File: test_html_creation.py

File: __init__.py

Directory: D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC 699/DiscordBotProject_CISC699\utils

File: css_selectors.py

File: DiscordUtils.py

File: ExcelUtils.py

File: HTMLUtils.py

--- temporary.py ---

```
import os
from fpdf import FPDF

# Directory where the project files are located
directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC 699\DiscordBotProject_CISC699"
output_pdf_path = os.path.join(directory, "project_text.pdf")

# Function to retrieve all text from files, ignoring .git and __pycache__ directories
def extract_project_text(directory):
    project_text = ""
    for root, dirs, files in os.walk(directory):
        # Ignore .git and __pycache__ directories
        dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]

        for file in files:
            if file.endswith('.py') or file.endswith('.txt') or file.endswith('.md'):  # Only considering relevant file types
```

```
            file_path = os.path.join(root, file)
            try:
                with open(file_path, 'r', encoding='utf-8') as f:
                    project_text += f"--- {file} ---\n"
                    project_text += f.read() + "\n\n"
            except Exception as e:
                print(f"Could not read file {file_path}: {e}")

    return project_text

# Function to generate a PDF with the extracted text
def create_pdf(text, output_path):
    pdf = FPDF()
    pdf.set_auto_page_break(auto=True, margin=15)
    pdf.add_page()
    pdf.set_font("Arial", size=12)

    # Ensure proper encoding handling
    for line in text.split("\n"):
        # Convert the text to UTF-8 and handle unsupported characters
        try:
            pdf.multi_cell(0, 10, line.encode('latin1', 'replace').decode('latin1'))
        except UnicodeEncodeError:
            # Handle any other encoding issues
            pdf.multi_cell(0, 10, line.encode('ascii', 'replace').decode('ascii'))

    pdf.output(output_path)
```

```
# Extract project text and create the PDF
project_text = extract_project_text(directory)
if project_text:
    create_pdf(project_text, output_pdf_path)
    output_pdf_path
else:
    "No project text found."
```

--- Tests_URLs.txt ---

database password: postgres

Working Commands: Test commands

!project_help

!login bestbuy

!launch_browser

!close_browser

!navigate_to_website https://www.google.com/

!monitor_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price
https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!get_price
https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

!check_availability https://www.opentable.com/r/bar-spero-washington/

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22"

!stop_monitoring

!stop

**********
Working on it:

!check_availability https://www.opentable.com/r/bar-spero-washington/ "August 22" "8:00 PM"

**********

URLs to Test:

https://www.opentable.com/r/bar-spero-washington/

https://www.ebay.com/itm/314411766963?_trkparms=amclksrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3Avlp_homepage

https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true

--- test_addAccount.py ---

```
import sys, os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_add_account():
    account_control = AccountControl()
```

```
    # Adding a new account
    account_control.add_account("newUser", "newPassword123", "newWebsite")

if __name__ == "__main__":
    test_add_account()
```

--- test_deleteAccount.py ---

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_delete_account():
    account_control = AccountControl()

    account_control.delete_account(4)

if __name__ == "__main__":
    test_delete_account()
```

--- test_excel_creation.py ---

--- test_fetchAccounts.py ---

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from control.AccountControl import AccountControl

def test_fetch_accounts():
    account_control = AccountControl()

    # Fetching all accounts
    account_control.fetch_accounts()

def test_fetch_account_by_website(website):
    account_control = AccountControl()

    # Fetch the account by website directly
    account = account_control.fetch_account_by_website(website)

    if account:
        username, password = account  # Unpack the returned tuple
        print(f"Website: {website}, Username: {username}, Password: {password}")
    else:
        print(f"No account found for website: {website}")

if __name__ == "__main__":
    test_fetch_accounts()
    test_fetch_account_by_website("ebay")
```

--- test_html_creation.py ---

--- __init__.py ---

```
#empty init file
```

--- css_selectors.py ---

```
class Selectors:
    SELECTORS = {
        "trendyol": {
            "price": ".featured-prices .prc-dsc"  # Selector for Trendyol price
        },
        "ebay": {
            "url": "https://signin.ebay.com/signin/",
            "email_field": "#userid",
            "continue_button": "[data-testid*='signin-continue-btn']",
            "password_field": "#pass",
            "login_button": "#sgnBt",
            "price": ".x-price-primary span"  # CSS selector for Ebay price
        },
        "bestbuy": {
            "url": "https://www.bestbuy.com/signin/",
            "email_field": "#fld-e",
            #"continue_button": ".cia-form__controls  button",
            "password_field": "#fld-p1",
```

```
            "SignIn_button": ".cia-form__controls  button",
            "price": "[data-testid='customer-price'] span",  # CSS selector for BestBuy price
            "homePage": ".v-p-right-xxs.line-clamp"
        },
        "opentable": {
            "url": "https://www.opentable.com/",
            "date_field": "#restProfileSideBarDtpDayPicker-label",
            "time_field": "#restProfileSideBartimePickerDtpPicker",
            "find_table_button": ".find-table-button",  # Example selector for the Find Table button
            "availability_result": ".availability-result",  # Example selector for availability results
                "show_next_available_button": "button[data-test='multi-day-availability-button']",  # Show
next available button
            "available_dates": "ul[data-test='time-slots'] > li",  # Available dates and times
            "no_availability": "div._8ye6OVzeOuU- span"
        }
    }

    @staticmethod
    def get_selectors_for_url(url):
        for keyword, selectors in Selectors.SELECTORS.items():
            if keyword in url.lower():
                return selectors
        return None  # Return None if no matching selectors are found
```

--- DiscordUtils.py ---

--- ExcelUtils.py ---

--- HTMLUtils.py ---