Team Anyalgorithms - Implementation of Feature / Fix

Deliverable 2

Ahmed Sirajuddin

Jon An

Kia Naderi

Ricky Chen

Robert Peng

Table of Contents

Potential Bug / Feature Choices

#4210: Telemetry is on by default

Link: https://github.com/mozilla-mobile/focus-android/issues/4210

When a user first runs Firefox-Focus, navigates to the settings menu, then to the “Privacy & Security” section, there is a “Sending Usage Data” switch which is set to on by default. For a browser that emphasizes its privacy features, this is a questionable choice and the users are not notified of this fact beforehand. A github user has issued a request to change it so that this switch is turned off by default. Our initial impression of this was simply that the solution would be to change the default value of this switch to false. However, this is assuming that the code for telemetry is controlled by an android switch without external factors. Our team expects this bug to be on the easier side of bugs and take around 72 hours; but there exists a possibility that solving this bug could potentially be more difficult than it seems as after looking through the codebase for this particular feature, there is more logic involved then just the xml itself.

#4397: "Safe browsing" section is missing from Settings"/"Privacy & Security"

Link: https://github.com/mozilla-mobile/focus-android/issues/4397

As of now, when a user opens Firefox Focus and tries to change their privacy settings in the settings menu, the section for “Safe browsing” is missing. A user has requested to make the section displayed in the “Privacy & Security” with the option to enable/disable. To implement this feature, we anticipate it would take around 8 hours of developer time and 72 hours worth of investigation. There are multiple questions to be asked as this issue does not come with a lot of information. For instance, is the safe browsing a tag for certain sections or is it its own setting? If it is its own setting, exactly what does it do? Currently, the settings for privacy are broken down into many different aspects of the web including but not exclusive to tracking protection, web content, cookies, etc. Our team anticipates that becoming familiar with these various components will be the most time consuming task of this implementation.

#4477: Make the button movable

Link: https://github.com/mozilla-mobile/focus-android/issues/4477

As of now, when a user browses a page with Firefox Focus, there is a button located at the bottom right of the screen where the user can switch tabs or erase the browsing history. The button disappears when the user scrolls down a page and re-appears when the user scrolls up a page. However, this button may obstruct web content and the user may accidentally press it which would pre-maturely end their browsing session. A user requested to make this button moveable by holding down on it and dragging it across the screen to the desired location. To implement this feature, we anticipate it would take at least two days' worth of work which would involve becoming familiar with gesture functions in Android Studio, how the button is implemented in the system, and then modifying as per feature request.

#4289: Open in Firefox Focus from notifications bar keeps crashing

Link: https://github.com/mozilla-mobile/focus-android/issues/4289

The following steps need to be taken to reproduce the app crashing is when clicking "Open" from the Settings menu. First, open Firefox Focus, then load a webpage. Next, enter the Settings menu, press the home button, then pull down the notification tray and click "Open" button. This issue currently exists when using Android devices (Android 9).

The cause of the issue, it appears, is that Firefox attempts to create a new display for the session when re-opening the browser, rather than using the same display. This causes Firefox Focus to crash. The solution is to locate the code that handles opening the browser and a new display (in the Rendering or Session Manager layer) and fix it so that a new display is not created when an existing display exists, as Firefox Focus is simply re-opening, rather than opening for the first time. This fix may take from a few hours to a couple (two or three) days. Taking on this task would require us to become more familiar with Kotlin as well as finding and fixing the code that is causing the issue. Verifying that is the correct and seemingly most efficient solution will take another day, on estimate.

#4431: Downloading a file with the same name as an existing one, overwrites the previous file

Link: https://github.com/mozilla-mobile/focus-android/issues/4431

When a new file is downloaded, it creates a name for that file. If the same file is downloaded, there is usually a number added to the file name to show that the file is a separate downloaded file. However, currently when focus downloads a file that has the same name as another file, the other file would be overwritten because only one file of such name can exist at a time. For instance, suppose we initially downloaded a file called important.png. If we then download another file called important.png, the expected behaviour of the browser would be to rename the new file to important(1).png. However, with the current behavior, it would overwrite the old file leaving the user with only one file called important.png. To fix this bug, we anticipate that it would take us one day to correct the code and a few days to create proper test cases to confirm that our change works.

#3652: Replace checkbox with switch for “Request desktop site”

Link: https://github.com/mozilla-mobile/focus-android/issues/3652

Currently, when a user opens the drop-down menu while browsing, the “Request desktop site” option is currently operated with the use of a checkbox. A user has requested to change the checkbox to an on/off switch; this would make all boolean functions in the menu (the other being “Trackers blocked”) be represented in a consistent manner. We anticipate it would take a day to implement this as it would involve becoming familiar with the implementation of UI elements on Android Studio and modifying the relevant code to be compatible with the change.

Two Bugs / Features Chosen

#4289: Open in Firefox Focus from notifications bar keeps crashing

Link: https://github.com/mozilla-mobile/focus-android/issues/4289

Reason Chosen

When considering the available options above, our team wanted our first fix to be something with meaningful impact but also focus on minimizing risk and effort on our side. This bug affected the users who used the Android device's notification tray to open apps and it was definitely an intrusive bug as it causes the entire app to crash. Another factor in our choice was that the description of the bug on GitHub was clear and concise; greatly minimizing efforts needed to get other relevant information compared to other issues. We already had the information needed to become familiar with the nature of the issue and be able to reproduce the bug easily; it was not necessary to ask any new questions. There is little to no risk with the changes involved in implementing this bug fix as it does not necessarily change the behaviour of the app -- it simply causes the app to not crash and resume its intended purpose. This gave our team a significant boost in confidence as failure to fix the bug would not drastically change the dynamic of the app and it would mean less time analyzing the structure of the app to get to the file we needed to change. So we anticipate that the extent to which we must become familiar with the code to resolve this issue and the task of properly implementing the bug fix is reasonable in scope considering the time limitation of this deliverable.

Details on Implementation

The problem exists for both when the user clicks "open" and "erase and open" from the Android device's notification tray. The code related to these problems can be found in the `TelemetryWrapper.kt` file here (and on line 545). In particular, the `eraseAndOpenNotificationActionEvent` and `openNotificationActionEvent` functions. These functions call `TelemetryEvent.create`. The code causes the app to crash because it attempts to create a new display for the session when one already exists, as the app is simply just re-opening. The fix was to change the code so that a new display is not created when re-opening the app. The exact code which causes the problem is within the `openNotificationActionEvent` function:

```
                TelemetryEvent.create(Category.ACTION, Method.CLICK,
Object.NOTIFICATION_ACTION, Value.OPEN).queue()
```

We noticed that similar code exists, that does not create a new display for the session, and instead behaves like re-opening the app. We decided that instead of retyping the same working code in a different place, we reused the code so that `openNotificationActionEvent` uses it. This makes the code more efficient, as the same code isn't typed in different places. The fix is as follows:

```
            TelemetryEvent.create(Category.ACTION,    Method.CLICK,
Object.HOMESCREEN_SHORTCUT, Value.OPEN).queue()
```

This fix uses `Object.HOMESCREEN_SHORTCUT`, and `openNotificationActionEvent` now behaves as intended. With this fix, the app is successfully reopens to the settings page after the user loads the settings page and attempts to reopen it from the Android device's notification tray. A comment was left above this line of code to inform users that the intended action for this function is to reopen the app without crashing, and that this change properly accomplishes the action, unlike the previous code that causes the app to crash when opened while on the Settings page.

Technical Commentary on Effects of Changes

The changes for this fix do not affect the structure of the code. It changes `openNotificationActionEvent` so that it uses `Object.HOMESCREEN_SHORTCUT` instead of `Object.NOTIFICATION_ACTION` as the former accomplishes the intended action correctly, while the latter crashes the app. This fix simply makes Firefox Focus not attempt to create a new display for an existing session, which is what was happening when the latter was being used. This was the significant difference between the two. The fix simply takes an instance of when the app crashes and prevents the crash and allows the intended action to take place. The change was in `TelemetryWrapper.kt` found here. The arguments to `TelemetryEvent.create` within the `openNotificationActionEvent()` function were fixed.

#3652: Replace checkbox with switch for “Request desktop site”

Link: https://github.com/mozilla-mobile/focus-android/issues/3652

This was chosen as the second change to implement for this deliverable. Similar to our other choice of bug / feature, a large consideration for this feature was based off of the team’s general unfamiliarity with Firefox Focus, Android Studio and Kotlin; this struck us as a relatively easy change to make to the system and an opportunity to learn and contribute without risking failing to implement a more ambitious change that is beyond our current ability.

Details on Implementation

As the feature in question is in regards to a change in UI for “Request desktop site”, the relevant files to be changed are `RequestDesktopsCheckItemViewHolder.kt` and `request_desktop_check_menu_item.xml`.

For the change in UI, the `request_desktop_check_menu_item.xml` file was modified so that the Checkbox became a Switch. The corresponding file -- `RequestDesktopsCheckItemViewHolder.kt` -- had to be changed to be compatible with the Switch object. The line import `android.widget.Switch` was changed to `import android.widget.Switch`. The line that formerly initialized the checkbox -- `private val checkbox: CheckBox = itemView.findViewById(R.id.check_menu_item_checkbox)` was modified to `private val desktop_switch: Switch = itemView.findViewById(R.id.check_menu_item_checkbox)` and all calls formerly made to checkbox had their name changed accordingly. As both objects of types `Checkbox` and `Switch` inherit methods from `CompoundButton`, the only necessary change that was needed for the code was simply a matter of changing the object type imported and initialized rather than re-coding the functionality.

Technical Commentary on Effect of Changes

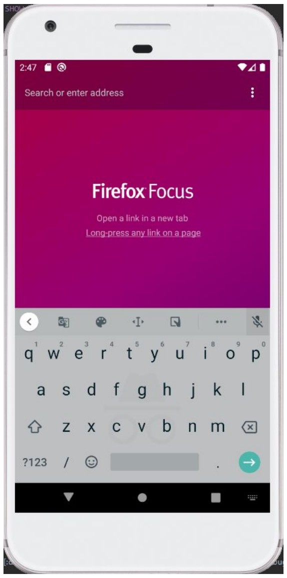
The extent to which this implementation changed the system is minimal in scope as it only required the modification of two files: the file containing the graphical properties -- `request_desktop_check_menu_item.xml` -- and the file containing the functional properties of “Request desktop site” -- `RequestDesktopsCheckItemViewHolder.kt`. With this implementation, the “Request desktop site” option now makes use of a switch instead of a checkbox.

Test Cases

#4289: Open in Firefox Focus from notifications bar keeps crashing

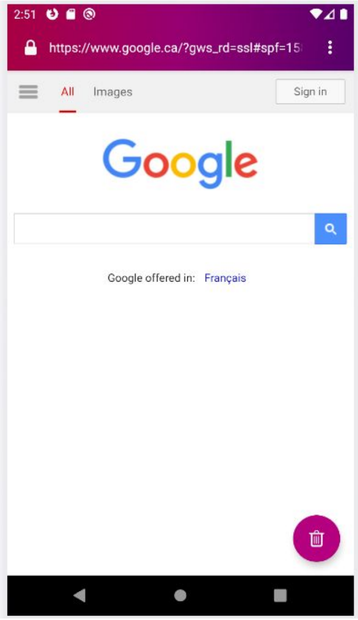
Acceptance test:

1. Open firefox focus



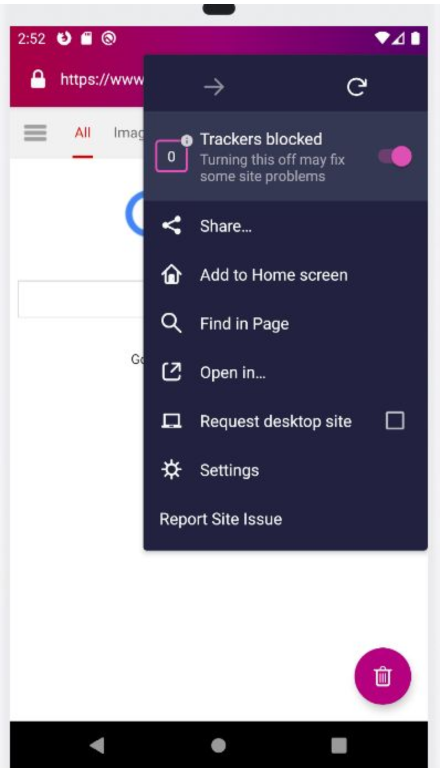
Open a link in a new tab
Long-press any link on a page

2. Go to any webpage (e.g google.ca)

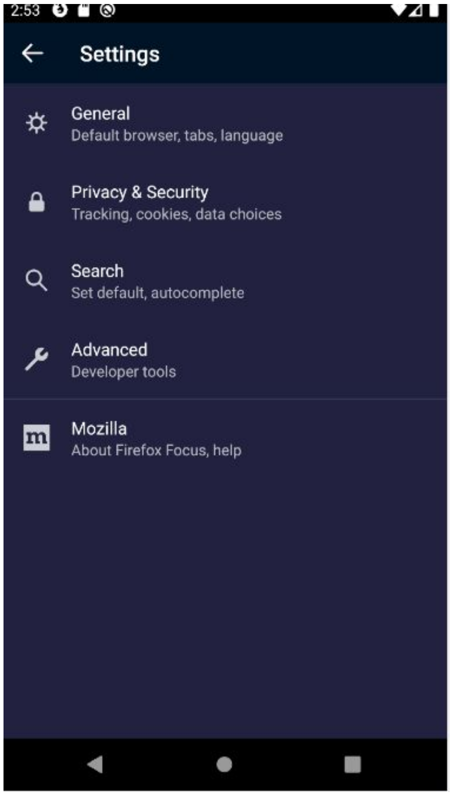


Google offered in: Français

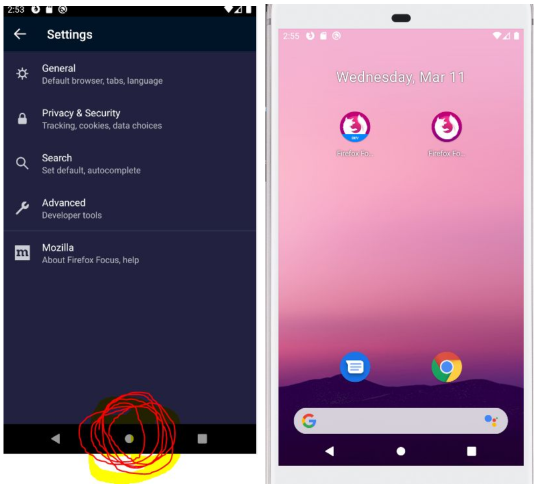
3. Top right corner click the 3 dots



4. Click Settings


2:53
Settings
General
Default browser, tabs, language
Privacy & Security
Tracking, cookies, data choices
Search
Set default, autocomplete
Advanced
Developer tools
Mozilla
About Firefox Focus, help

5. Click home screen button on the phone



2:53

Settings

General
Default browser, tabs, language

Privacy & Security
Tracking, cookies, data choices

Search
Set default, autocomplete

Advanced
Developer tools

Mozilla
About Firefox Focus, help

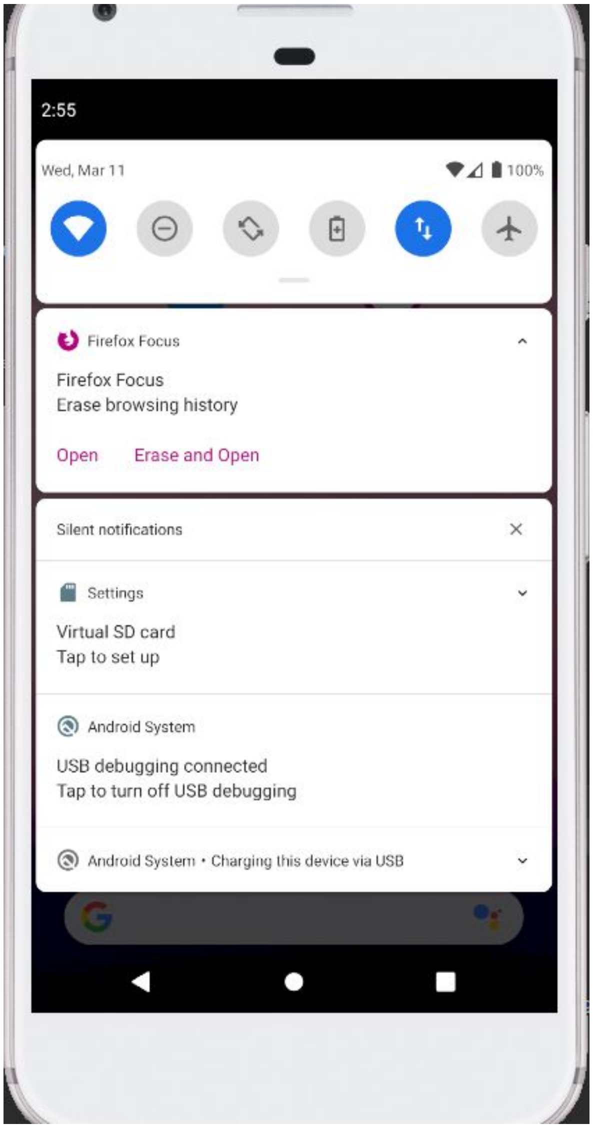
2:55

Wednesday, Mar 11

Firefox Fo...

Firefox Fo...

6. Drag down the tray



2:55

Wed, Mar 11 100%

Firefox Focus

Firefox Focus
Erase browsing history

Open Erase and Open

Silent notifications

Settings

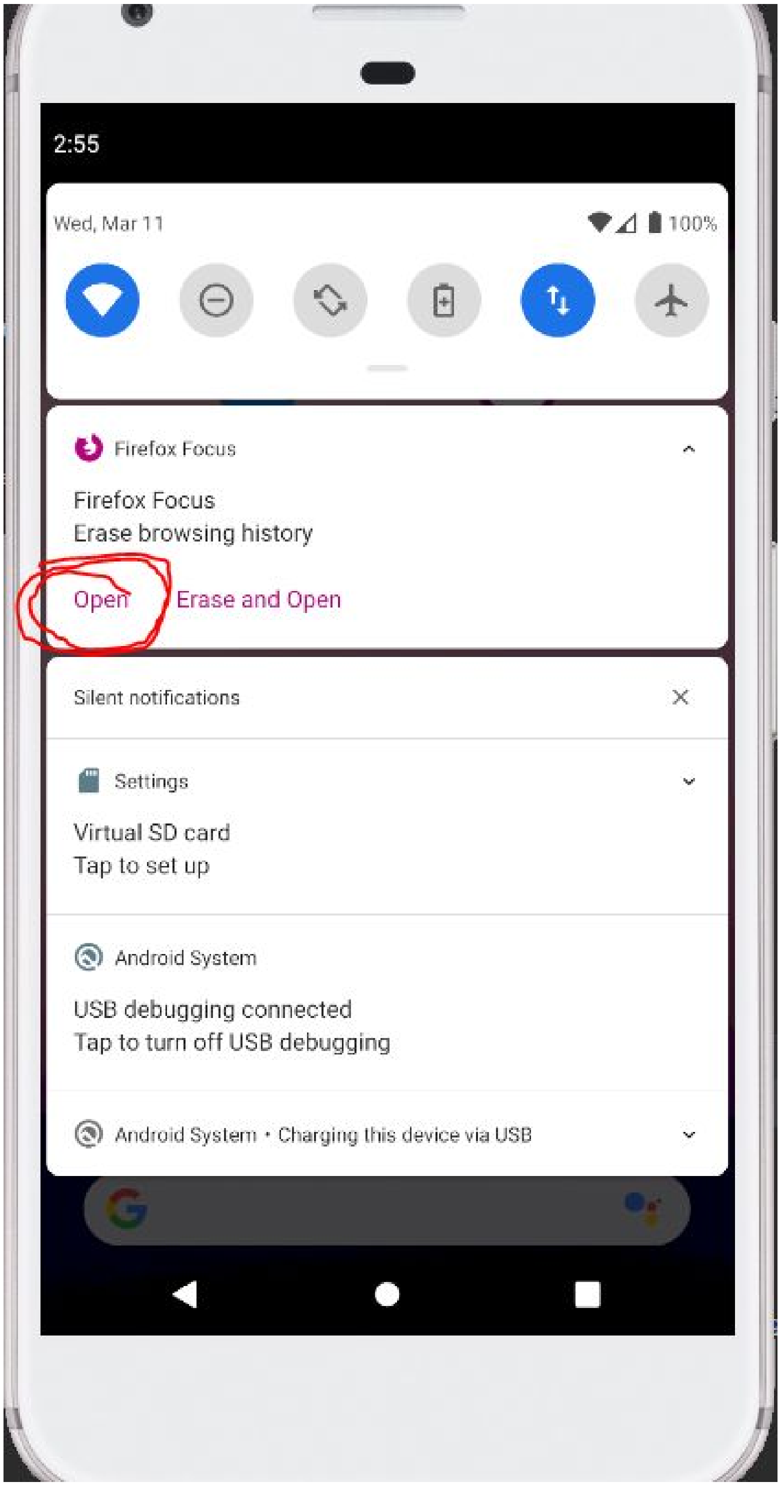
Virtual SD card
Tap to set up

Android System

USB debugging connected
Tap to turn off USB debugging

Android System • Charging this device via USB

7. Click open



Firefox Focus
Erase browsing history

Virtual SD card
Tap to set up

USB debugging connected
Tap to turn off USB debugging

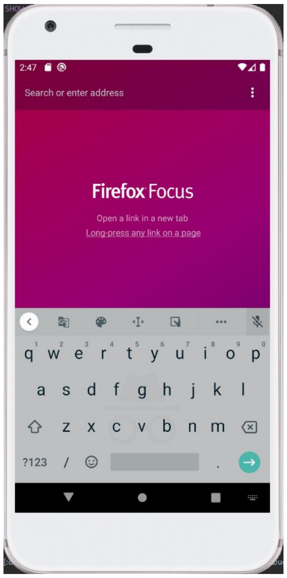
Android System • Charging this device via USB

User should ensure that the app has not crashed after clicking open.

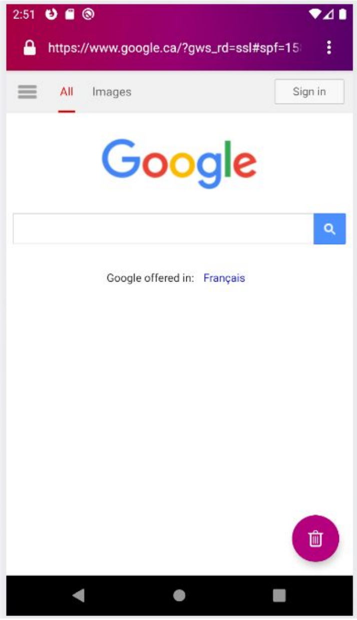
#3652: Replace checkbox with switch for “Request desktop site”

Acceptance test:

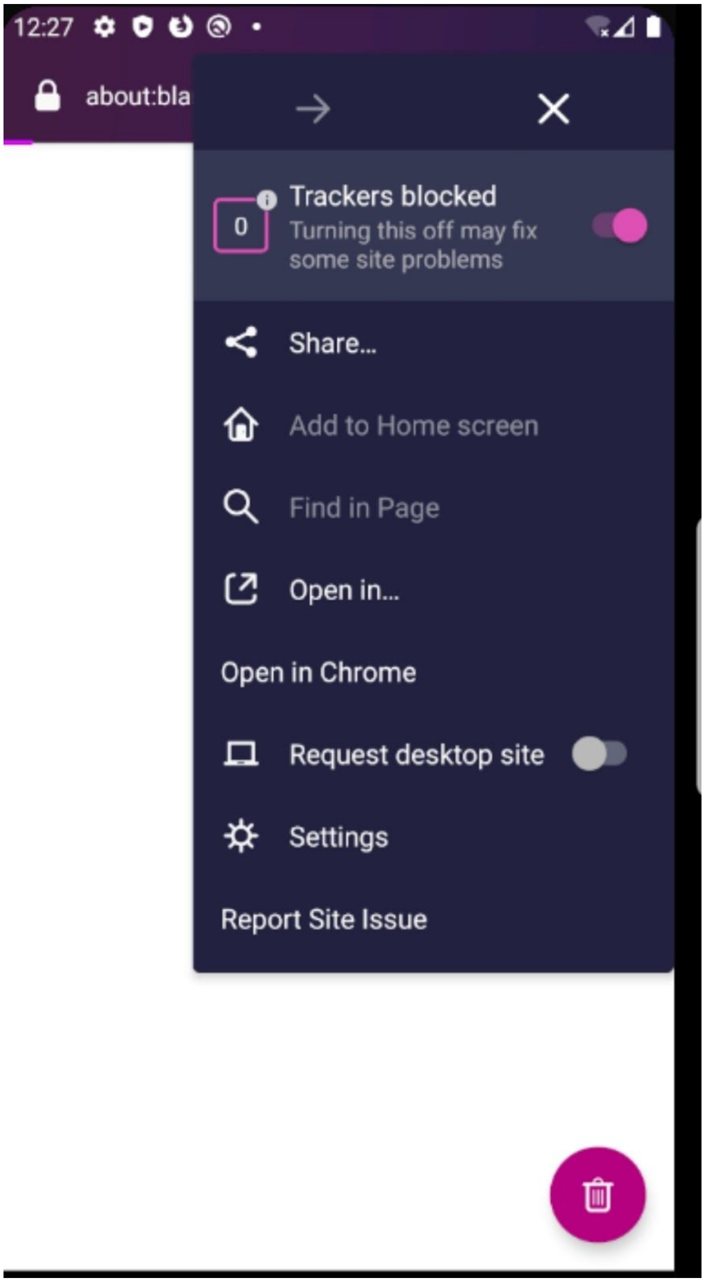
1. Open Firefox Focus.



2. Go to any website (e.g. google.ca).


2:51
https://www.google.ca/?gws_rd=ssl#spf=15
All
Images
Sign in
Google
Google offered in: Français

3. Tap on the menu button at the top-right corner of the screen. Users should see that the “Request desktop site” option is a toggle instead of check box.


12:27
about:bla
0
Trackers blocked
Turning this off may fix some site problems
Share…
Add to Home screen
Find in Page
Open in…
Open in Chrome
Request desktop site
Settings
Report Site Issue

The Software Process

We have been using Trello as a tool to follow the Kanban software process. For our board, we have five columns with the following categories: Todo, WIP, Code Review, QA, and Complete. The way in which we made use of the board is as follows.

We broke down all the tasks that are required to be completed for Deliverable 2 and made each its own card; ten cards were made for this deliverable which consisted of: five cards with each being a summary for a bug fix/feature request, two cards for implementing a bug fix or feature we have chosen, two cards for the test cases for the aforementioned implementations, and one card for the final write-up of our deliverable. All cards were initially in the Todo column. An individual was assigned to a card and the card would move to the WIP column. When a non-coding card was completed in WIP, it would move to the Complete column. When a coding card was completed in WIP, it would move to the Code Review column, then the QA column, then, finally, the Complete column when each respective tasks were finished.

The board gave us a clear indication of our progress in this deliverable as we could see what each member was doing and how many tasks were completed. However, one piece of information that each of the cards did not show is whether they relied on another card being completed first. Though this information could have been conveyed on the card, the team decided that it would be easier to make a separate “to-do” list that categorized each card in a phase of this deliverable. In summary, the phases for Deliverable 2 were as follows: find and summarize at least five bug fix/feature requests, then implement two requests of our choice as well as make test cases for them, then the final write-up for this deliverable.

Although it was not always possible to have the whole team attend a meeting, those who were able to attend would share a brief report on their progress on their assigned task, the difficulties they may have been facing, and what they intended to get done. For those who could not attend a meeting, they could nevertheless get a good understanding of the progress of the project simply with a quick glance at our board.

Overall, the Kanban process has been useful as a tool in coordinating the team in completing this deliverable as we were all able to clearly see the tasks that were not yet assigned, the tasks that were in progress, and the tasks that were completed. This has aided us in the selection of tasks as well as planning for future tasks.