First Steps

Setup	git clone https://github.com/GoogleChrome/push-notifications.git
Webserver (Chrome required)	https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhemlocgigb
Some basic links for the exercise!	

Links

Online Tutorial	https://developers.google.com/web/fundamentals/codelabs/push-notifications/
Google Drive	https://drive.google.com/drive/u/2/folders/1S2sFykIJ1q-PRhsTB1QWmG59ms_WBtKFK
Generate Keys	https://web-push-codelab.glitch.me/
These links were used to perform the exercise!	

Supplementary Features

Notification Click	sw.js	event.notifica- tion.close();
Notification Click	sw.js	clients.openWi- ndow('htt- ps://develop- ers.google.c- om/web/')
Unsubscribe	main.js	subscription.u- nsubscribe(); & updateSub- scrip- tionOnServer- (null);
sw.js and main.js are included in the setup folder!		

Activate Push Notification

Public Private Key	https://web-push-codelab.glitch.me/
Copy paste the subscription	located on bottom of the page

Basic changes from the original

Service Worker	sw.js	navigator.serv- iceWorker.regis- ter('sw.js')
Attach key	main.js	const applicati- onServerPubl- icKey = '<Your Public Key>';
update-Btn()	main.js	enables / disables button depending on subscr- iption of user | call in initializeUI()
Click Listener	initia-lizeUI()	pushButton.add- EventListener('- click', function() {...}
subscr-ibe-User()	main.js	swRegistrat- ion.pushMana- ger.subscribe({ userVisibleOnly: true, applicati- onServerKey: applicationServ- erKey }) .then(- function(- subscr- iption) {...})
update-OnS-erver()	main.js	Sends subscription to the backend
Handle Push Event	sw.js	listens for the event and shows a notifi- cation (console.log)
sw.js and main.js are included in the setup folder!		