# Team Members

* BVA
	+ PM: Brayden Beverage <brayden@bvacommerce.com>
	+ Lead DEV: Daniel Lerman <daniel.lerman@bvacommerce.com>
* PURIUM
	+ Main POC: Debbie Darling <debbie@puriumcorp.com>
	+ Developer #1: Brian
	+ Developer #2: Jessica <jessicacalderon@organicbynatureinc.com>
	+ PUR developers have access to BVA Repo
		- Training on 4/6/21 - Link to Recording: <https://drive.google.com/file/d/1VKlaUtzqYUXgTSj9t2aerPt0_ekuFr8y/view?usp=drivesdk>

# Recap High Level Expectations/ Assumptions

* BVA is responsible for…
	+ Code Reviews/ Approving PUR’s (PRs) Pull Request
	+ Merging into Production
* Client is responsible for
	+ Creating PRs
	+ Merge into Develop/ testing theme
	+ QA in testing theme
	+ Creating Rollback theme
	+ Making Production live
* NOTES/ TBD Discussion
	+ PUR to provide some type of cadence of upcoming PRs
	+ If BVA ever is developing a feature, BVA/ PUR will need to coordinate so that we can ensure accountability for deployments
* Themes
	+ Develop | BVA - Preview Theme Link: <https://ishoppurium.com/?_ab=0&_fd=0&_sc=1&preview_theme_id=112978198600>

# Workflow

BVA follows an adjusted GitFlow pattern for codebase management. [You can find the basic rules and implementation for GitFlow here](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow). BVA switches the “master” branch for one named “production”. Please ignore the “release” and “hotfix” flows at this time.

1. PUR to create **feature** branches forked from the **develop** branch
2. PUR to create a (PR) Pull Request for **feature** >> **develop** & Assign to BVA/ Daniel for Approval
3. BVA/ Daniel will be notified of the PR & will perform a high level code review, if the PR passes, he will Approve it
	1. General Timeline for BVA Code Review Approvals ~ 48 hours
	2. If it fails → he will send it back to PUR for remediation
4. PUR dev will be notified if their PR is approved → Then PUR is responsible for merging it into develop branch which will automatically trigger a deployment to the [Develop | BVA](https://ishoppurium.com/?_ab=0&_fd=0&_sc=1&preview_theme_id=112978198600) Theme in shopify
	1. If PUR has multiple features that they want to get deployed out together, then they can create multiple PRs and then wait till they are all approved till they merge into testing, or they can just do them one at a time.
5. PUR team is responsible for perform QA/ UAT in the Testing Theme (BVA is not liable for any QA issues from PUR code)
6. If PUR has approved the feature(s) in [Develop | BVA](https://ishoppurium.com/?_ab=0&_fd=0&_sc=1&preview_theme_id=112978198600) Theme in shopify & are ready for deployment then PUR team has to do the following 2 things
	1. STEP #1 (\*important this 1st step, must be done first) PUR to duplicate the Production Theme and name the duplicate “[Rollback] Production <Date: YYYY-MM-DD>” then publish the Rollback theme
		1. While this Rollback theme is live, PUR team needs to make they don’t make any updates to the Shopify Customizer (Theme settings/ sections settings) if they need to make updates, they must do so in the Rollback them & Production theme
		2. [Rollback theme docs](https://bvaccel.atlassian.net/wiki/spaces/TECH/pages/833617921/Developer%2BThemes%2BRollback%2BThemes)
	2. STEP #2 Then PUR Dev submits another PR for Develop >> Production & Assigns it to BVA/ Daniel
7. BVA/ Daniel will do another code review and Approve it, then BVA will merge it into Production Theme, which will be unpublished at that point as long as PUR followed step #1 w/ creating the Rollback theme (only BVA will have permission to merge into the Production branch). Once approved & merged, the deployment to the unpublished Production theme will automatically begin.
8. PUR will be notified of the PR Approval to production, then PUR is responsible for smoke-testing the Production theme → if all is good PUR team can make the Production Theme Live by re-publishing it.
	1. If any live issues occur from the most recent deployment, then PUR team can republish the most recent Rollback theme, then they’d restart the process

# **Project Setup**

## **Workflow Tools**

* **Bitbucket** - Git source control
* **Gulp** - Shopify theme asset upload while developing
* **Buddy** - code deployment

## **Setup a Dev Theme + Private App**

#### **Dev Theme**

* Navigate: Shopify Admin > Online store > Theme
* Duplicate live theme
* Rename new theme using this format: **[DEV] <name> | <company>**
	+ **Sample: [DEV] John | BVA**
* Get theme ID: dev theme (actions) > edit code, number at the end of the url
[save theme ID for Env setup]

#### **Dev Private App**

* Navigate: Shopify Admin > Apps > Manage private apps (bottom)
* Click: Create new private app (top right)
* Name app using this format: **DEV <name> | <company>**
* Set admin API permissions:
	+ Theme templates and theme assets - Read and write
* Click: Save (bottom)
* Get app password: click button on right of password field to copy
[save app password for Env setup]

## **Running Gulp Workflow**

#### **System Prerequisites**

* Install Node
* Install Gulp

#### **Pulling down the repo**

* Terminal: **git clone <git-url>**
* If this project uses bower, be sure to run bower install to fetch the project dependencies

#### **Env Setup**

After running bower install, an ENV file should be present in your project root. Edit the **.env** file to follow this format:

API\_KEY=<private-app-key>

PASSWORD=<private-app-password>

URL=<shopify-domain>.myshopify.com

THEME\_ID=<dev-theme-id>

#### **Gulp Commands:**

* **gulp**: runs a build then watches the files for changes
* **gulp -o**: Same as above, plus opens a tab with your dev theme
* **gulp build**: Rebuilds the /deploy folder
* **gulp <task>**: runs a single gulp task (see gulpfile.js for the tasks)

While running gulp, changes that you make will be compiled and uploaded into Shopify. You can preview your development theme to see changes reflected.