**Pushing an Angular Application to GitHub Repository: A Step-by-Step Guide**

The project is to push the Angular application to a GitHub repository, ensuring that the project is properly version-controlled and shared with the team. To accomplish this, we used Git for version control, GitHub as the remote repository, and Angular with Node.js as the development framework. Here's a detailed writeup on how the task was completed:

**Step 1: Set Up The Local Environment**

Install Git: Download and install Git on the local system from https://git-scm.com/downloads.

Install Node.js and Angular CLI: Ensure to have Node.js installed on the system. Then, install Angular CLI by running the command npm install -g @angular/cli in the terminal or command prompt.

**Step 2: Create the Angular Application**

Open the terminal or command prompt.

Navigate to the directory where you want to create the Angular application.

A new Angular application using Angular CLI with the command ‘*ng new AngularApplicationOnGithub’* was created.

Change into the newly created application directory with cd YourAppName.

**Step 3: Initialize a Git Repository**

Run git init to initialize a new Git repository in the Angular application's root directory.

**Step 4: Create a .gitignore File**

In the root directory of the Angular project, create a .gitignore file using a text editor.

Add the following content to the .gitignore file:

*# Node.js*

*node\_modules/*

*npm-debug.log*

*yarn-error.log*

*# Angular build artifacts*

*/dist/*

*/src/environments/environment.prod.ts*

This will ensure that Node.js dependencies, build artifacts, and environment-specific files are excluded from version control.

**Step 5: Stage and Commit The Changes**

Stage the changes in the Git repository by running git add . . This will add all files and changes to the staging area.

Commit the changes with a meaningful message by running git commit -m "Initial commit".

**Step 6: Create a GitHub Repository**

Go to https://github.com/ and sign in to your GitHub account.

Click on the "+" sign in the top-right corner and select "New repository."

Choose a repository name, add an optional description, and configure other settings as needed.

Click "Create repository."

**Step 7: Connect Local Repository to GitHub**

On the GitHub repository page, you'll see a set of instructions under the section "…or push an existing repository from the command line."

Follow the instructions to set the remote origin for the local repository:

*git remote add origin <repository\_url.git>*

**Step 8: Push The Code to GitHub**

Finally, push your local code to the remote GitHub repository using the following command:

*git push -u origin master*

**Step 9: Share the Repository Link**

Once the push is successful, go to the GitHub repository page.

Copy the URL from the address bar and share it.