

Push Your Application to GitHub

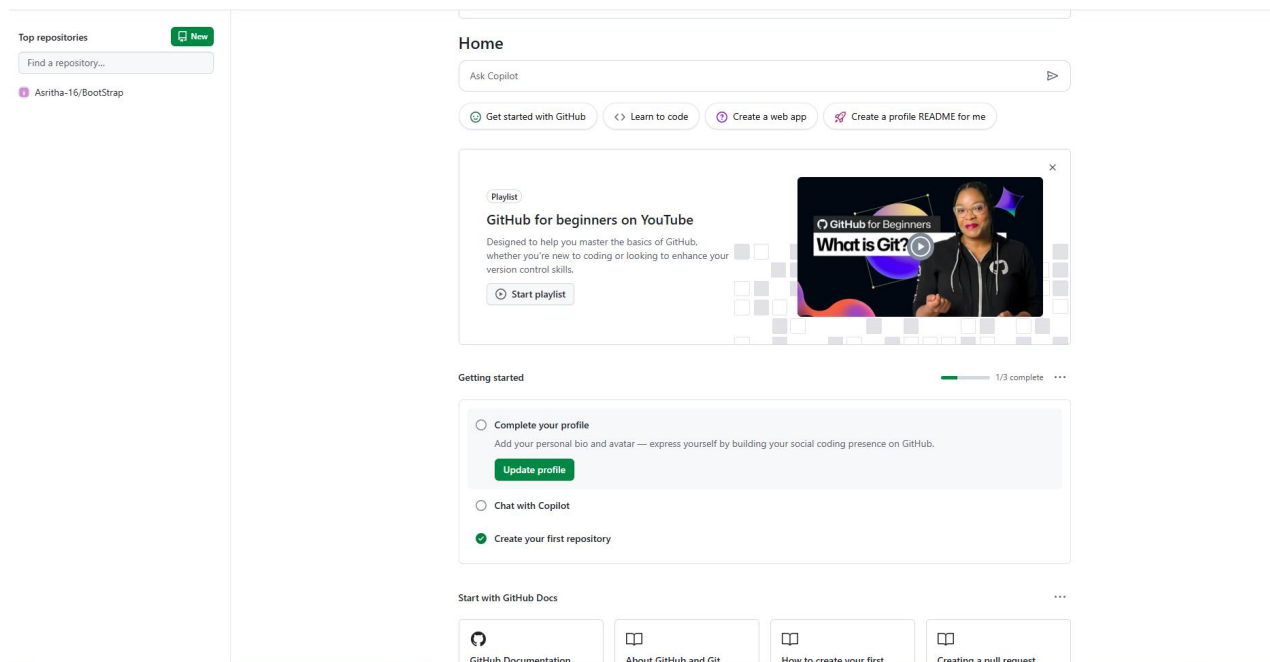
Goal:

To store your application in Git and upload it to GitHub so it's available online or as version-controlled code.

Step-by-Step Process:

Step 1: Open GitHub And Create a Repository on GitHub

Go to <https://github.com/Asritha-16>



Step 2: Click "New" to create a repository

Create a new repository [Preview](#) [Switch back to classic experience](#)

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * Asritha-16 / Repository name *

✔ Project_Bootstrap1 is available.

Great repository names are short and memorable. How about **fluffy-couscous**?

Description

0 / 350 characters

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

[Create repository](#)

Step 3: Set repository name and visibility (public/private)

Click Create Repository

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

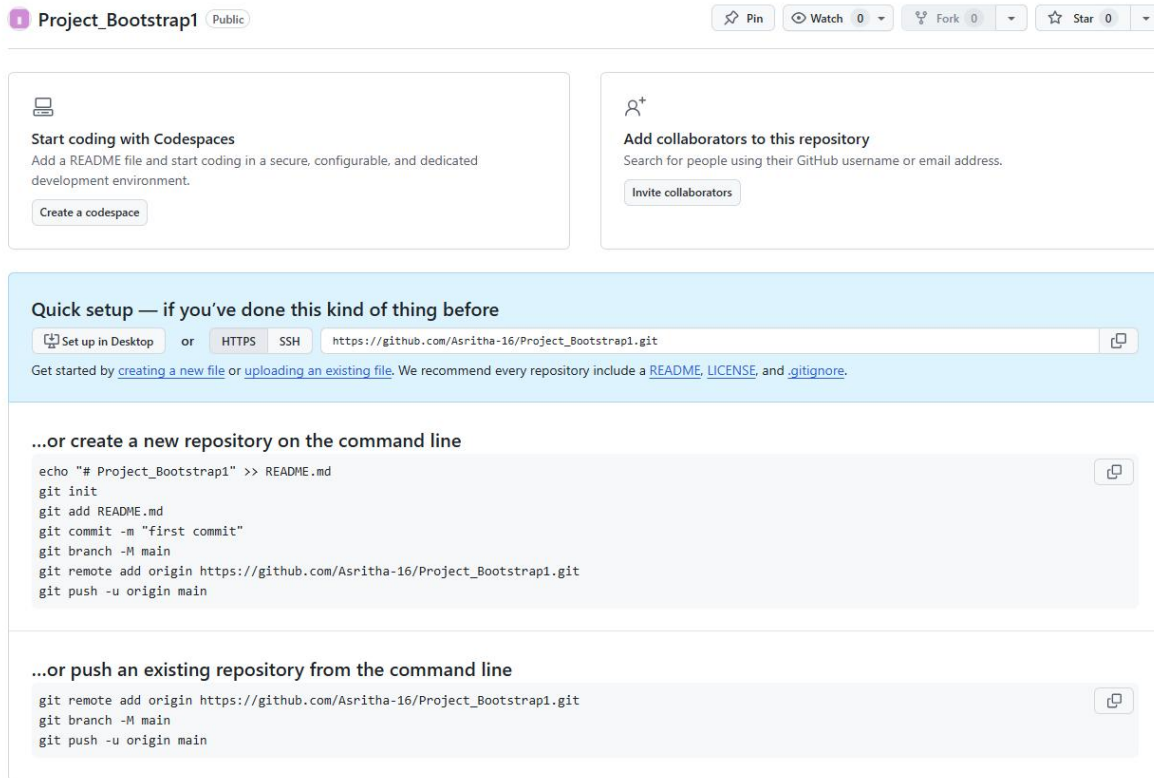
Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

[Create repository](#)

Step 4: Copy the repository link to run this in Git Bash:



The screenshot shows the GitHub repository page for 'Project_Bootstrap1'. It includes options to 'Set up in Desktop', 'HTTPS', and 'SSH'. A code block provides the commands to create a new repository on the command line. Another code block shows the commands to push an existing repository from the command line.

Project_Bootstrap1 Public

Pin Watch 0 Fork 0 Star 0

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
Create a codespace

Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators

Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH https://github.com/Asritha-16/Project_Bootstrap1.git
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

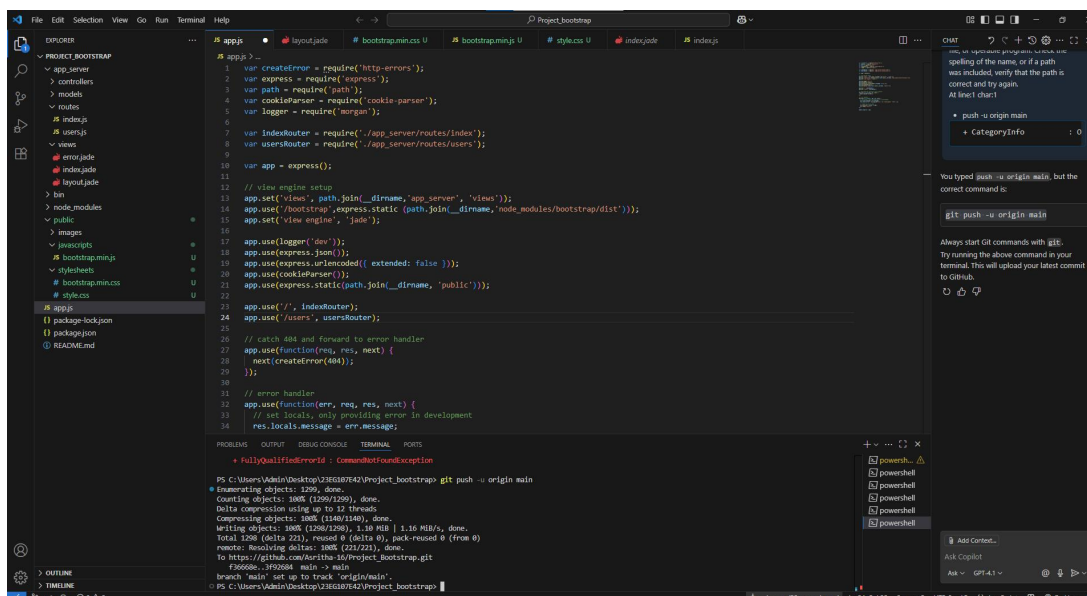
...or create a new repository on the command line

```
echo "# Project_Bootstrap1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Asritha-16/Project_Bootstrap1.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Asritha-16/Project_Bootstrap1.git
git branch -M main
git push -u origin main
```

Step 5: Goto VS code & open git bash:



The screenshot shows a VS Code editor with a project structure on the left. The main editor displays the 'app.js' file. The terminal window at the bottom shows the output of the 'git push' command, indicating that the repository has been successfully pushed to GitHub.

VS Code Explorer: PROJECT_BOOTSTRAP1, controllers, models, routes, index.js, users.js, views, error.js, index.js, layout.js, bin, node_modules, public, images, package-lock.json, package.json, README.md, bootstrap.min.js, stylesheets, bootstrap.min.css, styles.css.

app.js

```
1 var createError = require('http-errors');
2 var express = require('express');
3 var path = require('path');
4 var cookieParser = require('cookie-parser');
5 var logger = require('morgan');
6
7 var indexRouter = require('./app_server/routes/index');
8 var usersRouter = require('./app_server/routes/users');
9
10 var app = express();
11
12 // view engine setup
13 app.set('views', path.join(__dirname, 'app_server', 'views'));
14 app.use('bootstrap', express.static(path.join(__dirname, 'node_modules/bootstrap/dist')));
15 app.set('view engine', 'jade');
16
17 app.use(logger('dev'));
18 app.use(express.json());
19 app.use(express.urlencoded({ extended: false }));
20 app.use(cookieParser());
21 app.use(express.static(path.join(__dirname, 'public')));
22
23 app.use('/', indexRouter);
24 app.use('/users', usersRouter);
25
26 // catch 404 and forward to error handler
27 app.use(function(req, res, next) {
28   next(createError(404));
29 });
30
31 // error handler
32 app.use(function(err, req, res, next) {
33   // set locals, only providing error in development
34   res.locals.message = err.message;
```

Terminal:

```
PS C:\Users\Adin\Desktop\23EG107E42\Project_bootstrap1> git push -u origin main
Enumerating objects: 1299, done.
Counting objects: 386 (1250/1299), done.
Delta compression using up to 12 threads
Compressing objects: 180K (1140/1140), done.
Writing objects: 386 (1140/1299), 1.18 MiB | 1.16 MiB/s, done.
Total 1298 (delta 221), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 386 (221/221), done.
To https://github.com/Asritha-16/Project_Bootstrap1.git
  9560ae..3f92048 main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Adin\Desktop\23EG107E42\Project_bootstrap1>
```

A) Open Git Bash or your terminal inside the project folder:

```
"git init"
```

B) Commit the Changes:

```
"git commit -m "Initial commit""
```

C) Set the Branch Name:

```
" git branch -M main"
```

D) Connect Local Repository Of GitHub To Your Project:

```
"git remote add origin:https://github.com/Asritha-16/Project_Bootstrap.git"
```

E) Push Your Code to GitHub:

```
"git push -u origin main"
```