THE UNIVERSITY OF MELBOURNE

ePortfolios that help you get organised to showcase your technical skills in future job applications

Dr Eduardo Oliveira
[eduardo.oliveira@unimelb.edu.au]

CIS Industry Awareness Series

Agenda

- Who am I?
- ePortfolio
- 7-steps to organize your code repository
- [a few other related things]
- Conclusion

Who am I?

Recife, Brazil

Computer Science

(Bachelor/Masters/PhD)

Senior System Engineer

Technical consultant

14 years: Motorola/Samsung/Gemalto/Fiat

Senior Lecturer

7 years: Universities in Recife, Brazil

COMPAL

MOTOROLA

SAMSUNG

eclipse

FIAT

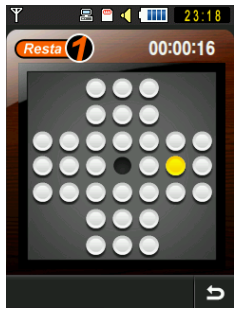
bematech

gemalto

security to be free


Google maps
Street View


Apple Campus
One Infinite Loop


23:18
Resta 1
00:00:16


15:45
Mobi-Farm
Mobi-Farm
LV2
29
-1
-42
Maize
7%


15:33
TENTATIVAS: 4
10/40


dematech


Recommended
Search All
PIG
SMART HUB
Search
Netflix
YouTube
Explore 3D
Facebook
Skype
Ad
Samsung Apps
Your Video
Fitness
Kids
Social TV
Signature Service
Channel List
Channel Manager
Schedule Manager
Web Browser
AllShare Play
My Applications
Source
Picasa
AccuWeather
Omok
TIME
TIME TV
SmartTV@samsung.com
Logout
WallPaper
Account Manager
Tools
Return


SAMSUNG
Widget
Message
Call
Keypad
Contacts
Messages
SAMSUNG

Computer Science

(post doctorate)

Melbourne,
Australia

Lecturer@CIS

ePortfolio

A portfolio is a compilation of materials that supports your skills, qualifications, education, training, and experiences. An eportfolio is generally a static website featuring work samples and it is closely related to online resume or CV.

Using an online portfolio allows you to show real examples of your work to back up your CV. Many candidates might only say they have experienced something but uploading your portfolio gives you the chance to showcase your work, thereby making your profile stand out.

ePortfolio

At the very least, your portfolio should include the following components:

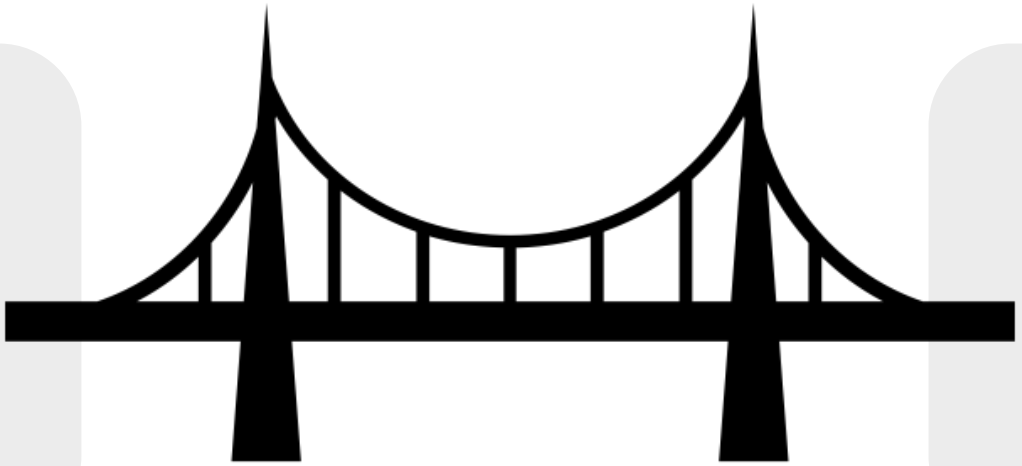
About Me – People want to know who you are, so make sure your portfolio includes some detail about that.

Projects – The critical component of any programming portfolio will be home to the very best samples of your work.

Contact Me – Without this, you could be hurting your chances of getting job offers. Ideally, you should include a contact form and your social media channels. If you don't do that, at the bare minimum, add your email address plus links to your LinkedIn and GitHub profiles.

ePortfolio

CIS



INDUSTRY

ePortfolio

Good Practices for Creating a Programming Portfolio Website

1. Tailor Samples for the Role You Want
2. Include Extracurricular Work
3. Ensure You Have a Responsive Design
4. Appealing Visual Design and Layout
5. Custom Web URL
6. Minimize the Touchpoints
7. Include Social Proof

ePortfolio

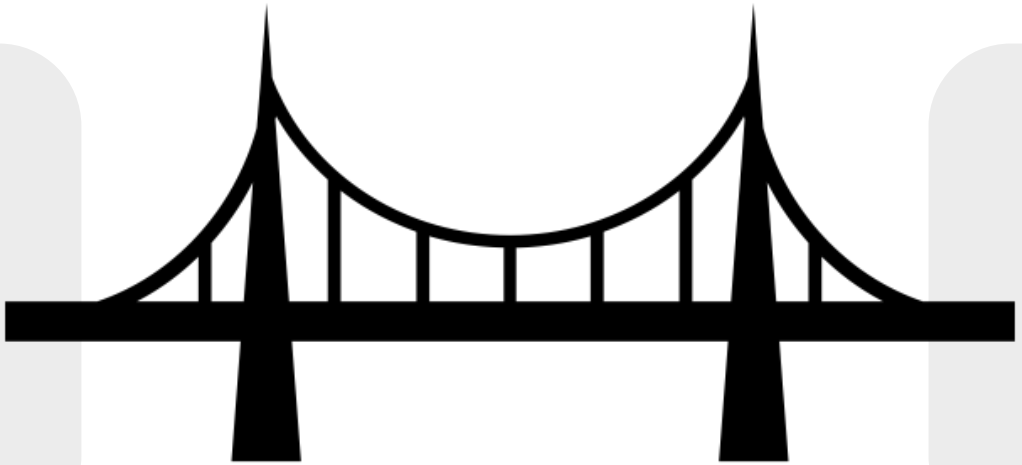
GitHub

Bitbucket

GitLab

stackoverflow

LinkedIn



INDUSTRY


GitHub

7-steps to organize your code repository

1. Keep your code repository professional
2. Keep your repo’s README.md updated
3. Commit often
4. Write good commit messages
5. Branch before you build
6. Structure your folders
7. Deploy your project

7-steps to organize your code repository

1. Keep your code repository professional

Treat this as a professional space

Add a photo to your profile (no avatars)

7-steps to organize your code repository

2. Keep your repo's README.md updated

What is that repository about?

Highlight the contents of your project (summary)

Setup and Configuration details

Is there a link to a live demo for that project?

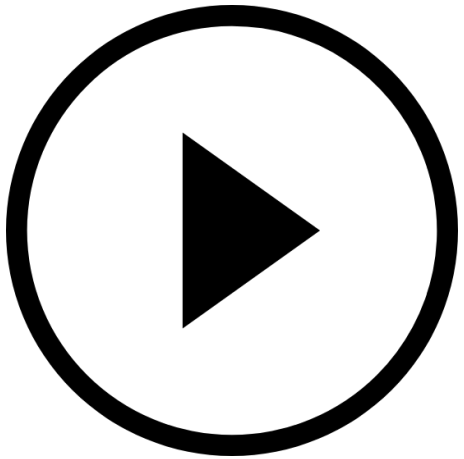
License details

The text of a license is usually stored in the LICENSE (or LICENSE.txt, LICENSE.md) file in the root of the project.

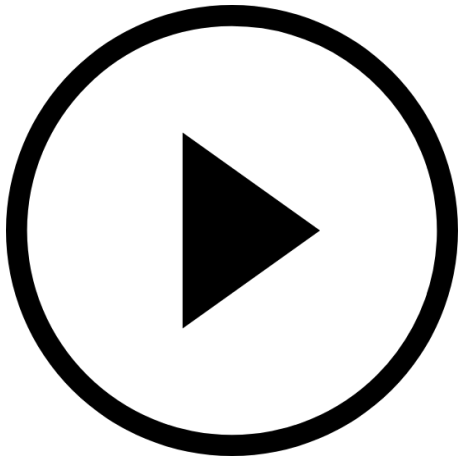
7-steps to organize your code repository

2. Keep your repo's README.md updated

Example/demonstration



react



strapi

7-steps to organize your code repository

3. Commit often

The smaller the changes, the easier it is to integrate the code later. Also, if everyone in the team is committing regularly, everyone is more likely to have the most up to date code

Each commit should be a single conceptual unit of change. Even if two changes happen in the same file, if they relate to two different bugs for example, they should go in different commits. Conversely, if changes in two separate files are related to the same bug, they should go together in the same commit

7-steps to organize your code repository

4. Write good commit messages

Use the imperative mood in the subject line

A properly formed Git commit subject line should always be able to complete the following sentence:

If applied, this commit will *your subject line here* (this should be your commit message. You do not need to write 'If applied…')

For example:

If applied, this commit will *refactor subsystem X for readability*

If applied, this commit will *update getting started documentation*

If applied, this commit will *remove deprecated methods*

7-steps to organize your code repository

5. Branch before you build

The naming convention simply adds prefixes to branch names, so that branches of the same type get the same prefix

feature/username/task_name

bugfix/username/task_name

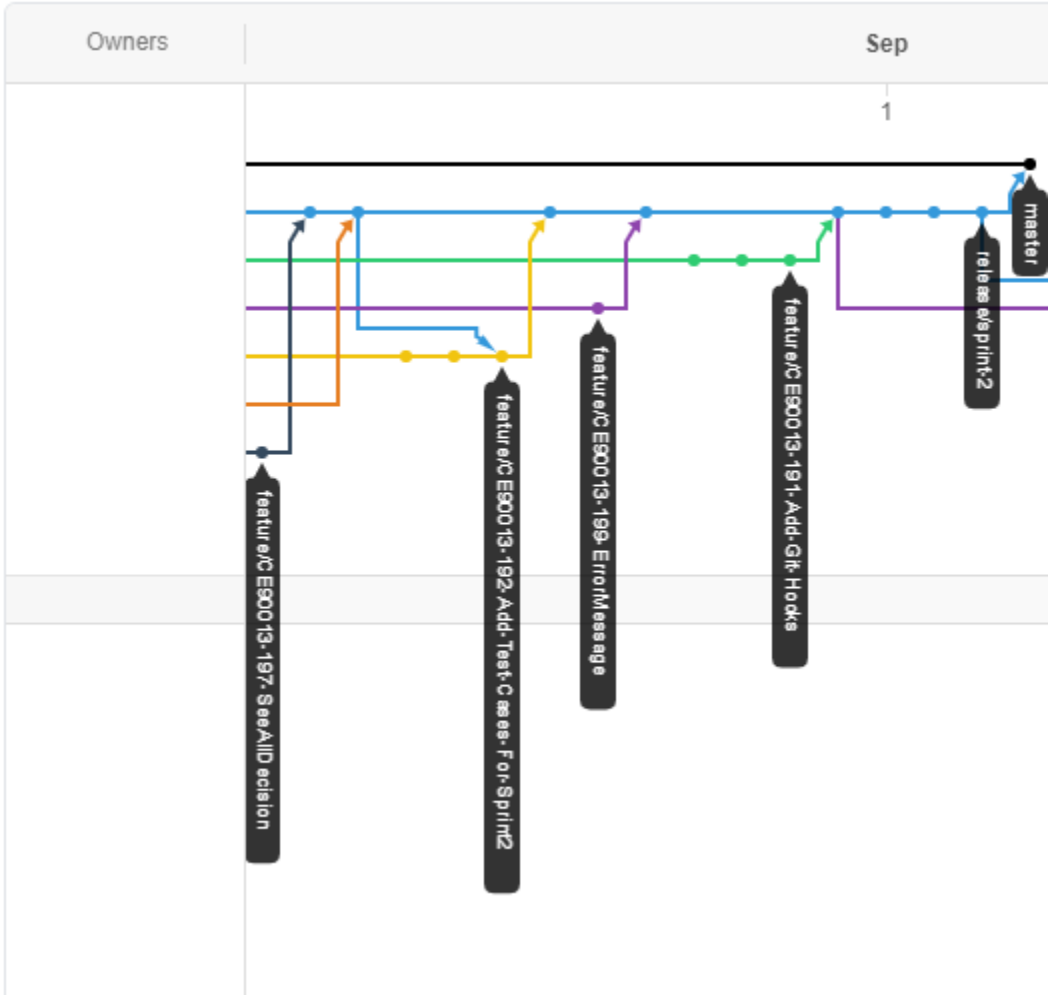
test/username/task_name

This lets you search for branches in many git commands, like this:

```
git branch --list "feature/*"
git log --graph --oneline --decorate --branches="feature/*"
git --branches="feature/*"
```

7-steps to organize your code repository

5. Branch before you build



7-steps to organize your code repository

6. Structure your folders

```
├── build                    # Compiled files (alternatively `dist`)
├── docs                     # Documentation files
├── src                      # Source files (alternatively `lib` or `app`)
├── test                     # Automated tests
├── tools                    # Tools and utilities
├── LICENSE
└── README.md
```

Use short lowercase names at least for the top-level files and folders except LICENSE, README.md.

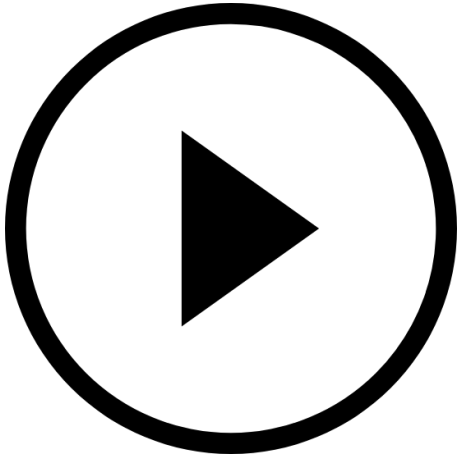
Automated tests are usually placed into the **test** or, less commonly, into the **spec** or **tests** folder.

```
├── ...
├── test                     # Test files
│   │
│   ├── benchmarks           # Load and stress tests
│   │
│   ├── integration          # End-to-end, integration tests
│   │
│   └── unit                 # Unit tests
└── ...
```

7-steps to organize your code repository

6. Structure your folders

Example/demonstration



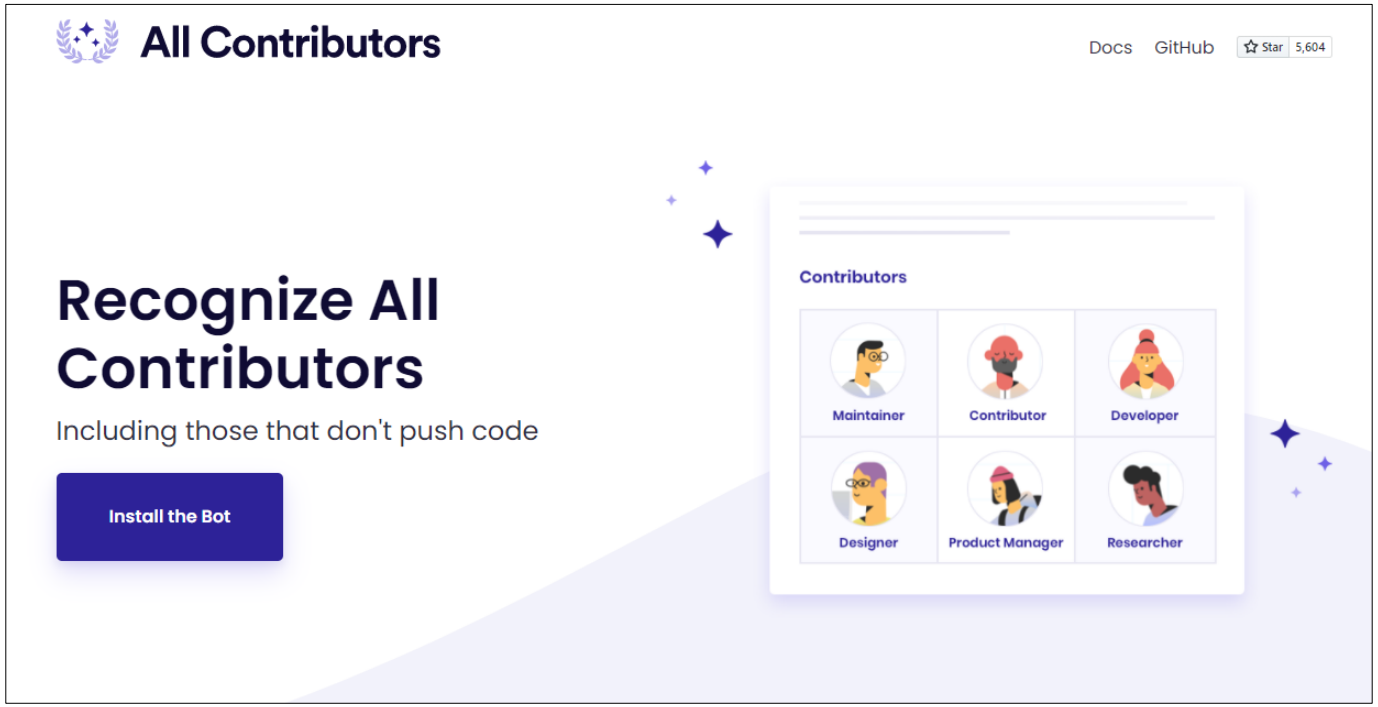
7-steps to organize your code repository

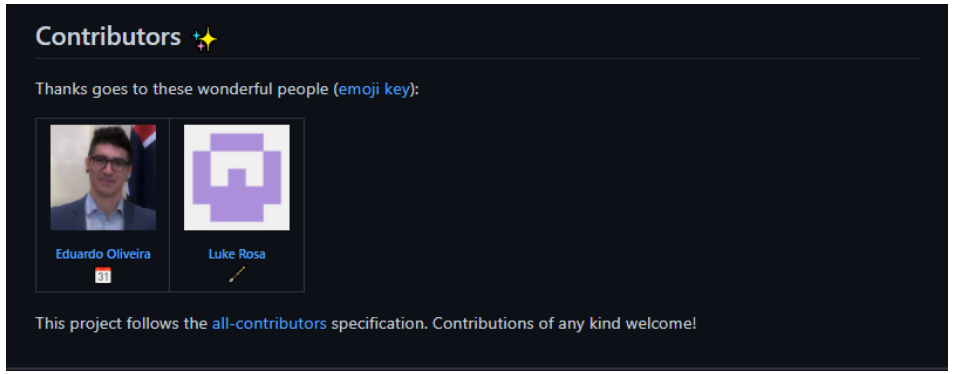
7. Deploy your project

Have your project ready to show! One of the worst possible things is to show up to an interview and offer to show your super cool website and then have to do a bunch of debugging or configuration in order to launch it (true story).

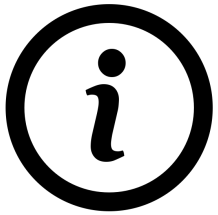
Deploy your project and have it publicly available via URL (keep the URL simple and easy to type for non-tech people).

apps


All Contributors
Docs
GitHub
Star 5,604
Recognize All Contributors
Including those that don't push code
Install the Bot
Contributors
Maintainer
Contributor
Developer
Designer
Product Manager
Researcher

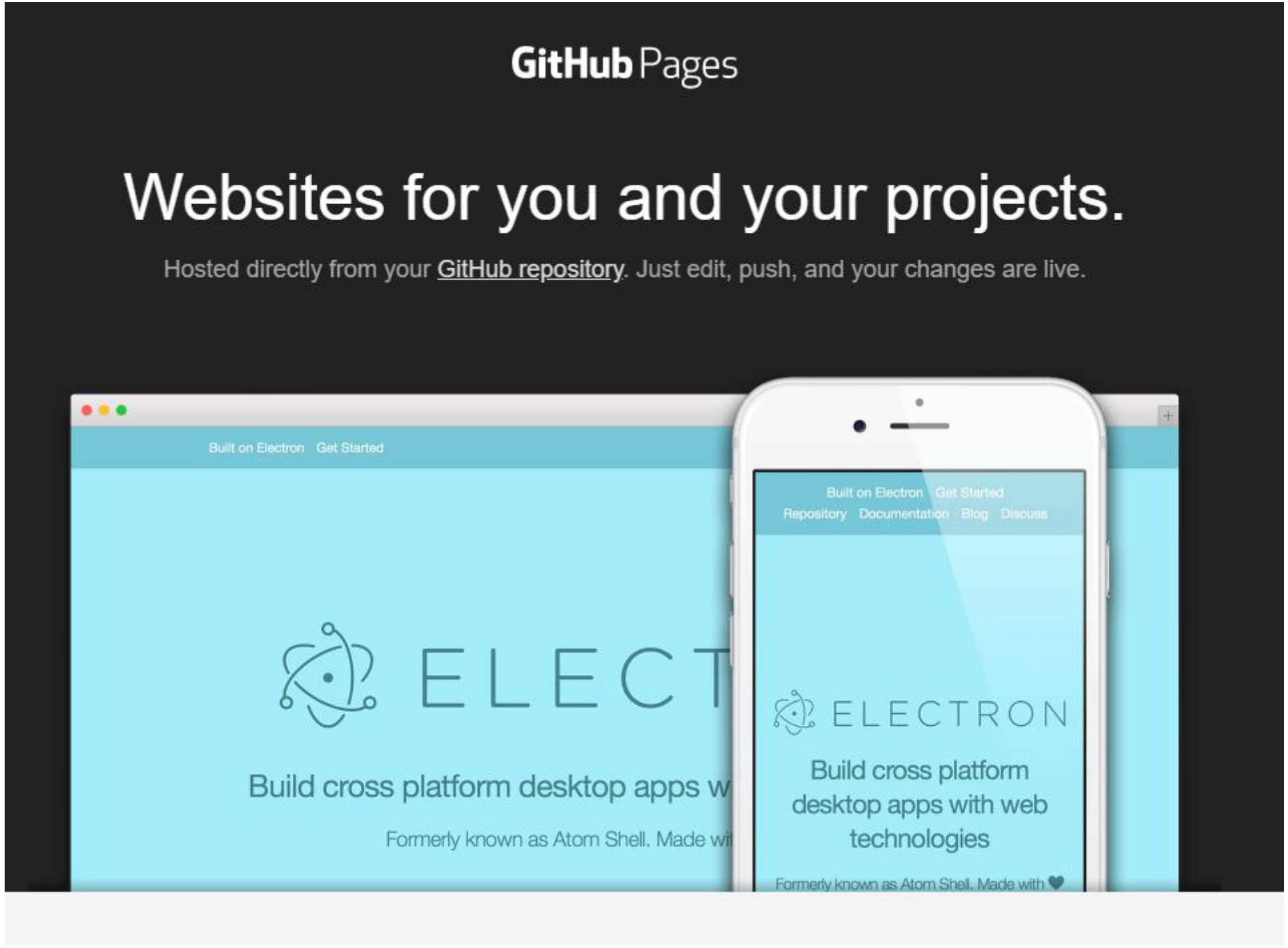

Contributors ✨
Thanks goes to these wonderful people (emoji key):
Eduardo Oliveira
Luke Rosa
This project follows the all-contributors specification. Contributions of any kind welcome!

The Best GitHub Marketplace Apps You Should Use



Top 10 Github Apps on Marketplace to Use as Junior JavaScript Developers

pages


GitHub Pages
Websites for you and your projects.
Hosted directly from your GitHub repository. Just edit, push, and your changes are live.
Built on Electron Get Started
Build cross platform desktop apps w
Formerly known as Atom Shell. Made wi
Built on Electron Get Started
Repository Documentation Blog Discuss
ELECTRON
Build cross platform desktop apps with web technologies
Formerly known as Atom Shell. Made with

CIS-Resources

UOM CIS Learning Resources

View the Project on GitHub
agogear/CIS-Resources

This project is maintained by agogear

Hosted on GitHub Pages — Theme by orderedlist

UOM CIS Repository

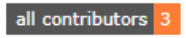


Table of contents

Goals

- Provide additional learning resources to CIS staff (so they can reuse/circulate these with their students)
- Make getting started with technologies and projects easier for everyone (staff/students)

Documentation

Documentation includes tutorials, workshops, project details, samples of code and so on (for various subjects).

What's Included

Within this repo download you'll find the following directories and files, logically grouping common resources. You'll see something like this:

```
├── README.md
├── checklists
|   └── <subject_code>
├── course_notes
|   └── <subject_code>
|       └── <year>
├── how_to
|   ├── README.md
|   └── <subject_code>
|       ├── <year>
├── project
```

./ Resources

View on GitHub

SWEN90007 Software Design and Architecture Repository

all contributors 2

Table of Contents

Goals

>> Provide additional learning resources to SWEN90007 students
>> Make getting started with SWEN90007 technologies and project easier
>> More details can be found here: SWEN90007 Handbook

Documentation

SWEN90007's documentation includes tutorials, workshops, project details, samples of code and so on.

The documents are related to our subject planning, available on Canvas

What's Included

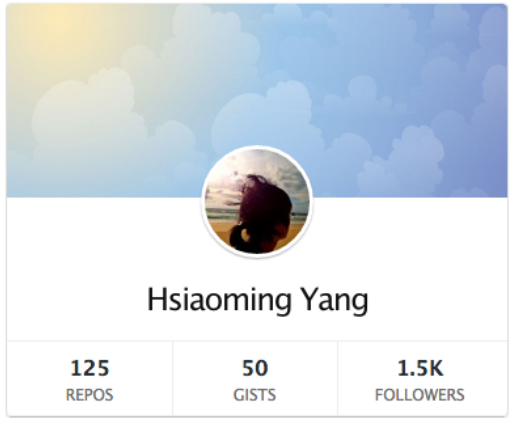
Within this repo download you'll find the following directories and files, logically grouping common resources. You'll see something like this:

```
├── README.md
```

cards

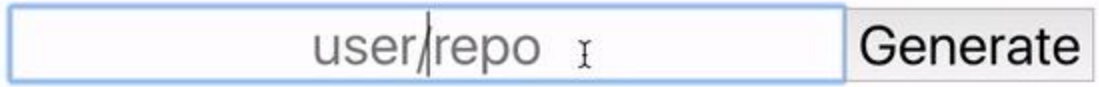
User Card

Show your GitHub profile.



Repo Card

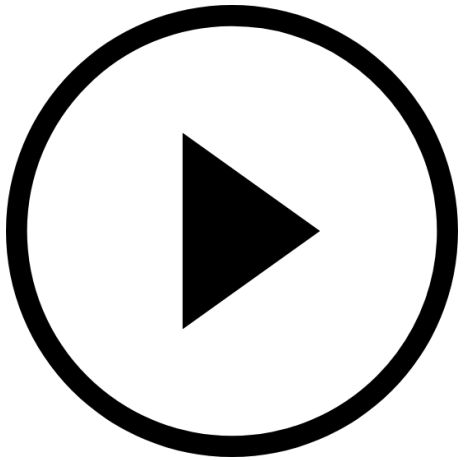
Show your GitHub repository.

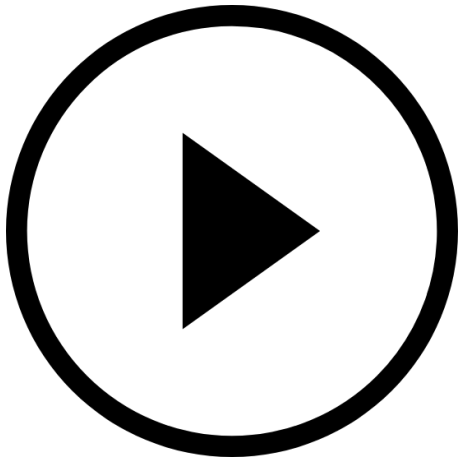

user/repo
Generate

portfolios

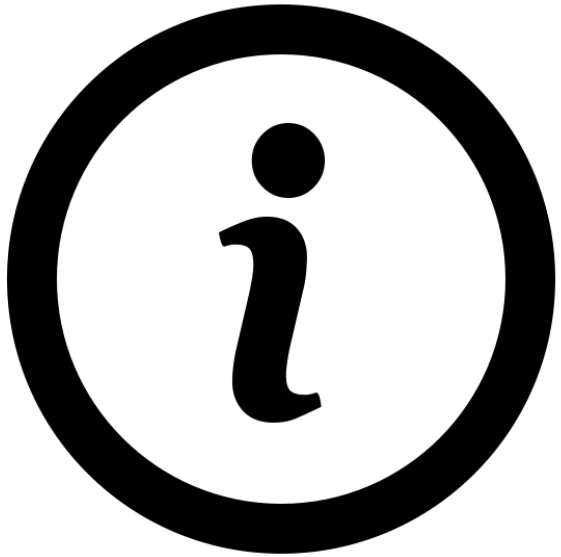
portfolios

Example/demonstration



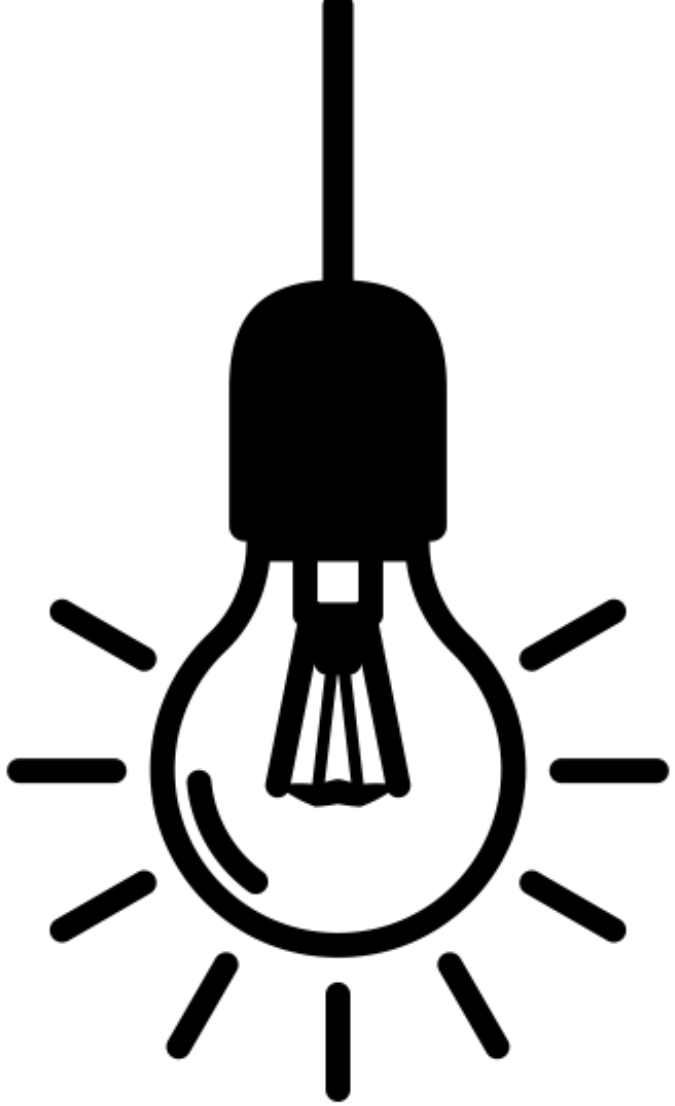


portfolios



List of [github] portfolios

What else



stackoverflow

Reputation and badges


LinkedIn

Conclusion

Thank you very, very much!

THE UNIVERSITY OF MELBOURNE

ePortfolios that help you get organised to showcase your technical skills in future job applications

Dr Eduardo Oliveira
[eduardo.oliveira@unimelb.edu.au]

CIS Industry Awareness Series