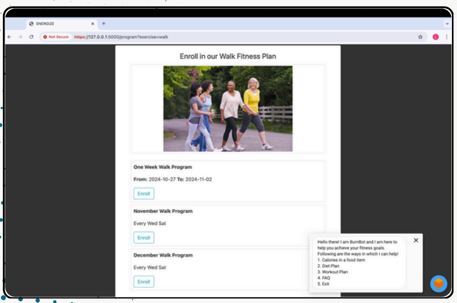
SCAN ME

ENERGIZE

Why Energize?

Energize is an easy-to-use web application designed to keep you motivated and in control of your fitness and health journey. With Energize, users can effortlessly monitor their progress, follow personalized workout plans, and gain valuable insights to live a healthier lifestyle. Whether you're new to fitness or an enthusiast, Energize offers the tools and guidance you need to achieve your goals, all within an intuitive and engaging interface.


ENERGIZE
Not Secure https://127.0.0.1:5000/program?exercise=walk

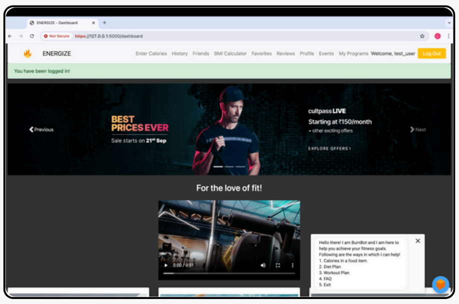
Enroll in our Walk Fitness Plan

One Week Walk Program
From: 2024-10-27 **To:** 2024-11-02
Enroll

November Walk Program
Every Wed Sat
Enroll

December Walk Program
Every Wed Sat
Enroll

Hello there! I am BurnBot and I am here to help you achieve your fitness goals. Following are the ways in which I can help!
1. Calories in a food item
2. Diet Plan
3. Workout Plan
4. FAQ
5. Exit


ENERGIZE - Dashboard
Not Secure https://127.0.0.1:5000/dashboard

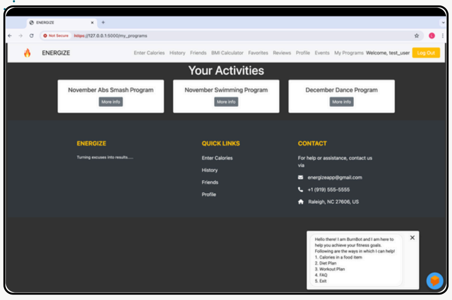
ENERGIZE | Enter Calories | History | Friends | BMI Calculator | Favorites | Reviews | Profile | Events | My Programs | Welcome, test_user | Log Out

You have been logged in!

Previous | BEST PRICES EVER | Sale starts on 21st Sep | cultpass LIVE | Starting at ₹150/month | + other exciting offers | EXPLORE OFFERS › | Next

For the love of fit!

Hello there! I am BurnBot and I am here to help you achieve your fitness goals. Following are the ways in which I can help!
1. Calories in a food item
2. Diet Plan
3. Workout Plan
4. FAQ
5. Exit


ENERGIZE
Not Secure https://127.0.0.1:5000/my_programs

ENERGIZE | Enter Calories | History | Friends | BMI Calculator | Favorites | Reviews | Profile | Events | My Programs | Welcome, test_user | Log Out

Your Activities

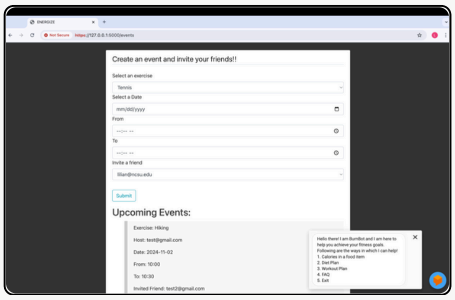
November Abs Smash Program — More info
November Swimming Program — More info
December Dance Program — More info

ENERGIZE
Turning excuses into results.....

QUICK LINKS
Enter Calories
History
Friends
Profile

CONTACT
For help or assistance, contact us via
energizeapp@gmail.com
+1 (919) 555-5555
Raleigh, NC 27606, US

Hello there! I am BurnBot and I am here to help you achieve your fitness goals. Following are the ways in which I can help!
1. Calories in a food item
2. Diet Plan
3. Workout Plan
4. FAQ
5. Exit


ENERGIZE
Not Secure https://127.0.0.1:5000/events

Create an event and invite your friends!!

Select an exercise: Tennis
Select a Date: mm/dd/yyyy
From: --:-- --
To: --:-- --
Invite a friend: lilian@ncsu.edu
Submit

Upcoming Events:
Exercise: Hiking
Host: test@gmail.com
Date: 2024-11-02
From: 10:00
To: 10:30
Invited Friend: test2@gmail.com

Hello there! I am BurnBot and I am here to help you achieve your fitness goals. Following are the ways in which I can help!
1. Calories in a food item
2. Diet Plan
3. Workout Plan
4. FAQ
5. Exit

What's New?

- **Enhanced Program Plans:** Our streamlined enrollment process makes signing up for programs more intuitive and user-friendly.
- **View Enrollments:** Conveniently track all your current enrollments in one place, simplifying participation management.
- **Event Sharing:** Create and invite your friends to join fitness events, like a 30-minute tennis session, fostering greater community engagement.
- **Email Reminders:** Stay organized with timely email reminders for upcoming fitness events, ensuring you never miss an important date.
- **One-Click Login:** Log in effortlessly using your Google account for quick access to all features without needing to remember multiple passwords.
- **UI Improvements:** Experience our newly redesigned user interface with an updated navbar and menu, enhancing navigation and providing improved page responsiveness for a seamless experience across devices.

Future Scope

- **Team Up on Your Goals:** Share your fitness journey! Invite friends to join program plans and motivate each other towards success.
- **Stay On Track, Every Step:** Seamlessly manage your fitness schedule and events to keep your goals in sight and progress unstoppable.
- **Find Your Fitness Crew:** Effortlessly locate and connect with friends using search feature, making your fitness journey a team adventure!

Group 40
Habib Mohammed
Lingjun Liu
Lawrence Arkoh

60 TESTS ENSURING RELEASE CONFIDENCE

Project 2

Group 40
Members:

Habib Mohammed - himohamm
Lingjun Liu - lliu39
Lawrence Arkoh - larkoh

GitHub link: https://github.com/CSC510-GROUP-40/FitnessApp/tree/develop

Total	105	
Notes	Grade	Evidence
Workload is spread over the whole team (one team member is often X times more productive than the others…	3	Commits
but nevertheless, here is a track record that everyone is contributing a lot)	3	Commits
Number of commits	3	248
Number of commits: by different people	3	Commits
Issues reports: there are **many**	3	9 issues reported
Issues are being closed	3	9 issues closed
Docs: doco generated, format not ugly	3	Readme Doc
Docs: what: point descriptions of each class/function (in isolation)	3	Doc string in implementation https://github.com/CSC510-GROUP-40/FitnessApp/pull/35/files
Docs: how: for common use	3	watch video

cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z		
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	3	Readme Doc
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	Watch video via Via Github Repo And Youtube
Use of version control tools	3	GitHub
Test cases exist	3	60 testcases
Test cases are routinely executed	3	See GitHub actions
Issues are discussed before they are closed	3	closed issues
Chat channel: exists	3	screenshot of google chat
Test cases: a large proportion of the issues related to handling failing cases.	3	Failing test cases: #25, #26, #27, #28 Other bug issues: #7, #8, #14, #15, #17
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	Demo
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	Demo

Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	Demo
Evidence that the members of the team are working across multiple places in the code base	3	Commits
Short release cycles	3	Releases
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	.gitignore
The file INSTALL.md lists how to install the code	3	INSTALL.md
The file LICENSE.md lists rules of usage for this repo	3	LICENSE.md
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	CODE-OF-CONDUCT.md
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	CONTRIBUTING.md
The file README.md contains all the following	3	Click to view on Github
Video	3	https://drive.google.com/file/d/1_qcWgYOCQ_Epj31jZ2Bs1_uOPPfvMzI

		6/view
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this:	3	in GH
Badges showing your style checkers	3	config files in GH showing your config README.md
Badges showing your code formatters.	3	config files in GH showing your this formatter's config README.md
Badges showing your syntax checkers.	3	config files in GH showing this checker's config README.md
Badges showing your code coverage tools	3	config files in GH README.md
Badges showing any other Other automated analysis tools	3	config files in GH README.md
Software Sustainability Evaluation		
Does your website and documentation provide a clear, high-level overview of your software?	Y	This is clearly described in the README.md and WEBSITE
Does your website and documentation clearly describe the type of user who should use your software?	Y	We are targeting users who want muscle gain and weight loss. This can found in the website and documentation.

Do you publish case studies to show how your software has been used by yourself and others?	N	
Is the name of your project/software unique?	Y	The name of our project is ENERGIZE
Is your project/software name free from trademark violations?	Y	
Is your software available as a package that can be deployed without building it?	N	Our software is a flask app (python scripts) and requires RabbitMQ as the message queue services for handly background tasks
Is your software available for free?	Y	First it is open source and will be available for free if we are to host it on a live server
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	Y	Project source can be downloaded from Github
Is your software hosted in an established, third-party repository likeGitHub (https://github.com), BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?	Y	Click to view on Github

Is your documentation clearly available on your website or within your software?	Y	Readme Doc
Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	Y	Readme Doc
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	Y	Readme Doc (manual and how to deploy)
Do you provide a comprehensive guide to all your software’s commands, functions and options?	Y	The README and INSTALL markdown files gives information on commands for setting up and running the software
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	N	We do not have any of such scenarios yet
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	N	Our software is not a substitute for a library or package

Do you store your documentation under revision control with your source code?	Y	All our documentation is committed just like codes and pushed to github
Do you publish your release history e.g. release data, version numbers, key features of each release etc. on your web site or in your documentation?	Y	Releases
Does your software describe how a user can get help with using your software?	Y	This information is stated in the footer of the website
Does your website and documentation describe what support, if any, you provide to users and developers?	Y	We stated in our readme on how support should be requested and hence we will accept and respond to all types of requests
Does your project have an e-mail address or forum that is solely for supporting users?	N	
Are e-mails to your support e-mail address received by more than one person?	N	
Does your project have a ticketing system to manage bug reports and feature requests?	Y	Issues or new feature requests can be raised on this Github page
Is your project's ticketing system publicly visible to your	Y	We use github issues are our ticketing systems and this is publicly available

users, so they can view bug reports and feature requests		
Is your software's architecture and design modular?	Y	Views, html templates, tests, are grouped into individual modules
Does your software use an accepted coding standard or convention?	Y	We enforce the standards from autopep-8
Does your software allow data to be imported and exported using open data formats? e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones	N	
Does your software allow communications using open communications protocols? e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones	Y	Our software uses http and HTTPS
Is your software cross-platform compatible? e.g. does it run under two or more of Windows, Unix/Linux and Mac OS X, or can be used from within two or more of Internet Explorer, Chrome, Firefox and Safari?	Y	Yes, our software can be run on windows, linux or macos and accessed through any web browser.(chrome, firefox, edge, safari, opera)

Does your software adhere to appropriate accessibility conventions or standards?	Y	
Does your documentation adhere to appropriate accessibility conventions or standards?	Y	
Is your source code stored in a repository under revision control?	Y	Project is on github
Is each source code release a snapshot of the repository?	Y	We use github tags
Are releases tagged in the repository?	Y	https://github.com/CSC510-GROUP-40/FitnessApp/tags
Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	Y	main and develop branch
Do you back-up your repository?	Y	
Do you provide publicly-available instructions for building your software from the source code?	Y	This is stated in readme and installation markdown files

Can you build, or package, your software using an automated tool? e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setuptools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)	N	This project is based on python and cannot be package into executables like projects written in other languages like JAVA
Do you provide publicly-available instructions for deploying your software?	N	Deploying the software is deployment platform dependent and hence we cannot provide an exact instruction
Does your documentation list all third-party dependencies?	Y	We list it in requirements.txt
Does your documentation list the version number for all third-party dependencies?	Y	We list it in requirements.txt
Does your software list the web address, and licenses for all third-party dependencies and	N	This is not required for our use case.

say whether the dependencies are mandatory or optional?		
Can you download dependencies using a dependency management tool or package manager? e.g. Ivy (http://ant.apache.org/ivy/), Maven (https://maven.apache.org/), Python pip (https://pypi.python.org/pypi/pip) or setuptools (https://pypi.python.org/pypi/setuptools), PHP Composer (https://getcomposer.org/), Ruby gems (https://rubygems.org), or R PackRat (https://rstudio.github.io/packrat/)	Y	Our dependencies can be downloaded with PIP
Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	Y	We have our tests here: https://github.com/CSC510-GROUP-40/FitnessApp/tree/develop/tests
Do you have an automated test suite for your software?	Y	We used github CI actions https://github.com/CSC510-GROUP-40/FitnessApp/actions

Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	Y	We used github CI actions https://github.com/CSC510-GROUP-40/FitnessApp/actions
Do you use continuous integration, automatically running tests whenever changes are made to your source code?	Y	We used github CI actions https://github.com/CSC510-GROUP-40/FitnessApp/actions
Are your test results publicly visible?	Y	https://github.com/CSC510-GROUP-40/FitnessApp/actions/workflows/unit_test.yml
Are all manually-run tests documented?	N/A	All of the tests can be run automatically. There are no manually-run tests in our repo.
Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software? e.g. release announcements, publications, workshops, conference presentations	Y	Releases
Does your website state how many projects and users are associated with your project?	N	

Do you provide success stories on your website?	N	We do not have this at the moment since project is not live
Do you list your important partners and collaborators on your website?	N	We do not have this at the moment since project is not live
Do you list your project's publications on your website or link to a resource where these are available?	N	
Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	N	
Can users subscribe to notifications to changes to your source code repository?	N	
If your software is developed as an open source project (and, not just a project developing open source software), do you have a governance model?	N	
Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	Y	

Do you have a contributions policy?	Y	
Is your contributions' policy publicly available?	Y	
Do contributors keep the copyright/IP of their contributions?	Y	We have copyright statement here: https://github.com/CSC510-GROUP-40/FitnessApp/commit/7213308cafb79cd844053558dcb3e6127ea2969b
Does your website and documentation clearly state the copyright owners of your software and documentation?	Y	We have listed copyright owners here: https://github.com/CSC510-GROUP-40/FitnessApp/commit/7213308cafb79cd844053558dcb3e6127ea2969b
Does each of your source code files include a copyright statement?	Y	We have copyright statement here: https://github.com/CSC510-GROUP-40/FitnessApp/commit/7213308cafb79cd844053558dcb3e6127ea2969b
Does your website and documentation clearly state the licence of your software?	Y	This is included in our repository: see on github
Is your software released under an open source licence?	Y	MIT (view license)
Is your software released under an OSI-approved open-source licence?	Y	MIT is an OIT license
Does each of your source code files include a licence header?	Y	We have license header here: https://github.com/CSC510-GROUP-

		40/FitnessApp/commit/7213308cafb79cd844053558dcb3e6127ea2969b
Do you have a recommended citation for your software?	Y	Liu, L., Arkoh, L., & Mohammed, H. (2024). ENERGIZE. Zenodo. https://doi.org/10.5281/zenodo.14025468
Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	Y	This was a short project for the semester. However, we had deliverables as shown in the poster with indications for future work highlighted.
Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	N	
Do you make timely announcements of the deprecation of components, APIs, etc.?	N	

SOFTWARE SUSTAINABILITY INSTITUTE

ENERGIZE - Sustainability Evaluation

This software evaluation report is for your software: ENERGIZE. It is a list of recommendations that are based on the survey questions to which you answered "no".

If no text appears below this paragraph, it means you must already be following all of the recommendations made in our evaluation. That's fantastic! We'd love to hear from you, because your project would make a perfect case study. Please get in touch (info@software.ac.uk)!

Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?
A great way of showing off your software is to write case studies about how yourself, and others, have used it. Case studies can help potential users learn about your software. They also act as a great advert for your software. If you can show happy users benefiting from your software, you are likely to gain more users.

Question 3.1: Is your software available as a package that can be deployed without building it?
Building software can be complicated and time-consuming. Providing your software as a package that can be deployed without building it can save users the time and effort of doing this themselves. This can be especially valuable if your users are not software developers.
You should test that your software builds and runs on all the platforms it is meant to support, which means you will already have created packages that can be distributed to your users!
See our guide on Ready for release? A checklist for developers (http://www.software.ac.uk/resources/guides/ready-release).
If you're interested in the consequences of ignoring your users' needs, see our guide on How to frustrate your users, annoy other developers and please lawyers (http://www.software.ac.uk/resources/guides/how-frustrate-your-users-annoy-other-developers-and-please-lawyers).

Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?
Troubleshooting information helps users quickly solve problems. It can help to reduce the number of support queries that you have to deal with.

Question 4.6: If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?
If your software includes support for Application Programming Interfaces (API) (https://en.wikipedia.org/wiki/Application_programming_interface), whether these be functions, data types, or classes offered by a library or a collection of REST (https://en.wikipedia.org/wiki/Representational_state_transfer) endpoints or web services, then

these need to be documented if you want them to be used. Code examples alone may not provide enough information on how someone can use your API in their own code.
From structured comments in the code, generating complete, structured API documentation can be done automatically with, for example, JavaDoc (http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html) (for Java), Doxygen (http://www.stack.nl/~dimitri/doxygen/) (for C, C++ , Fortran or Python), Sphinx (http://sphinx-doc.org/) (for Python). Certain REST frameworks, such as Django (http://www.django-rest-framework.org/), also support auto-generation of API documentation.

Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?
E-mail is purpose-made for resolving users' problems. The user can provide a good description of their problem and can attach screenshots, log files or other supporting evidence, and it's easy for you to ask for follow-up information, if required. You should always try to have an e-mail address for support queries.
It's best if the support e-mail address is clearly labelled as such e.g. myproject-support@myplace.ac.uk. This makes it easy for users to identify the e-mail address on your website or within your documentation, and it helps you to separate your support queries from all of your other e-mail. However, a personal e-mail address is better than nothing if you don't have the means to provide a dedicated support address.
See our guide on Supporting open source software (http://software.ac.uk/resources/guides/supporting-open-source-software). Its advice applies to supporting closed source software too.

Question 5.4: Are e-mails to your support e-mail address received by more than one person?
It's easy to forget about an e-mail, especially one that's asking difficult questions, so your e-mails to your support e-mails address should always be received by more than one person. One person should still have the primary responsibility of handling users' e-mails, but others can step up to handle e-mails if necessary, so that a user's query will be acknowledged even if one of you is on holiday, ill or otherwise indisposed.
See our guide on Supporting open source software (http://software.ac.uk/resources/guides/supporting-open-source-software). Its advice applies to supporting closed source software too.

Question 7.1: Does your software allow data to be imported and exported using open data formats? (e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones)
Supporting open data formats, data formats which are publicly available, as a complement to any proprietary ones you need to support, has many benefits. If data can be imported and exported in an open format, then it can be used with any software that uses this format. Software that supports open formats do not lock users into that software, because they will be able to use other software to access their data, if necessary. This can make it easier for users to swap between software that uses open formats. This means that users of other software may switch to your software, if your software is more efficient, robust, scalable or functional than that of your competitors'.
Try and adopt open formats that are mature, ratified standards, if possible. Standards can go through many iterations, because they evolve as ideas are proposed and debated and the scope, remit and intent of the standards are agreed. If a standard changes, then any software that uses the standard needs to be changed to keep up to date. Most of the big changes occur early in the lifetime of a standard. Mature and ratified standards are less likely to change significantly or

frequently, which reduces the risk of you having to modify your software in response.

Question 11.2: Can you build, or package, your software using an automated tool? e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setuptools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)?
Typing in lots of instructions is both time-consuming and prone to error. An automated build/packaging tool can make building or packaging your software easier, and less error-prone. Automation is also useful for developers: it makes it easier for them to rebuild or repackage code after implementing extensions, enhancements or bug fixes.

Question 11.3: Do you provide publicly-available instructions for deploying your software?
If there are no instructions for deploying your software, how will your users deploy it? At best, you'll end up dealing with lots of queries about how to deploy your software. At worst, you'll get no queries, nor any users, as if they can't deploy it they can't use it. Unless your software is a standalone EXE file or a single Linux/UNIX executable, then you need to provide deployment instructions.

Question 11.6: Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?
Users don't want to have to search the web for your third-party dependencies to find the information they need to package or deploy your software. You already know all the information that your users will need about suitable versions, licences and suchlike, so you should make it available to your users. In particular, licence information is very important, because users need to understand the terms and conditions of third-party dependencies so that they can determine whether they are legally permitted to use them, and, so, use your software.

Question 13.2: Does your website state how many projects and users are associated with your project?
Where you have an active set of users and developers, advertising their existence is not just good for promoting the success and life of your project. If potential users see that there are a large number of users, they know that your project is thriving, your software is useful and is under active development. This may encourage them to use your software, knowing that if they run into problems, there may be people who can help, and who they can share experiences with.

Question 13.3: Do you provide success stories on your website?
A great way of showing off your software is to write case studies about the people who've used it and how they've used it. This helps potential users learn about the software but, more to the point, is a great advert for your software. If you can show happy users benefiting from your software, you are likely to gain more users.

Question 13.4: Do you list your important partners and collaborators on your website?
Providing a list of important partners and collaborators gives potential users valuable assurance that your software has a future. Also, the higher the scientific, academic or industrial reputation of those partners, the higher the perceived reputation of your software, and project, will be.

Publicly recognising partners' efforts in improving or working with your software also increases the likelihood they will continue to use, or develop, your software in the future. Credit where credit is due!

Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?
Listing your software publications provides an academic perspective on the value of your software. It can also help users, and other stakeholders (e.g. current and potential funders) to understand, in detail, how your software contributes to research, what scientific problems it has helped to solve. In addition, these can help to show where your software sits in relation to other software that fulfils a similar need, and what makes yours different, or better.
These publications also gives researchers something to cite when they write their own papers where your software has been used, which is of value to them and also increases your citation count for your papers, which helps you demonstrate your impact!

Question 13.6: Do you list third-party publications that refer to your software on your website or link to a resource where these are available?
Providing a list of third-party publications can show, academically, how the software is used by others, as well as promoting their efforts and successes.
It also gives potential users ideas for how they may choose to use the software, as well as providing assurance that the software can be used by people other than its original developers to achieve something. Having such a list also means you can cite these publications in your own papers, funding proposals and reports to show or justify its value and the impact you have made! As a matter of routine, you should always ask people to cite your software if they've used it in their research for these reasons, and to inform you if they have included such a reference in one of their papers.

Question 13.7: Can users subscribe to notifications to changes to your source code repository?
Keeping information on these notifications as open as possible helps you present the impression that your project is open and inclusive. If users are actively developing using your software, keeping them up to date with on-going development on your source code (e.g. implementation of enhancements, extensions or bug fixes) enables them to factor these into their own development plans. For example, users might want to use an up-to-date copy from the repository to benefit from bug fixes made since your last release. If the notifications include information on the changes made (e.g. excerpts of the source code showing additions, removals and alterations) then this may allow for rapid identification of bug as any subscriber may notice an issue in the changes made.
A lightweight, automated subscription process can reduce, even remove, from you the overhead of managing notifications. It also means that users aren't subject to long delays waiting for their membership to be approved.

Question 13.8: If your software is developed as an open source project (and, not just a project developing open source software), do you have a governance model?
A governance model sets out how a, open source project is run. It describes the roles within the project and its community and the responsibilities associated with each role; how the project supports its community; what contributions can be made to the project, how they are made, any conditions the contributions must conform to, who retains copyright of the contributions and the process followed by the project in accepting the contribution; and, the decision-making process in

within the project.
Though they are designed for open source projects, many of their concerns are relevant to any software project.
OSS Watch (http://oss-watch.ac.uk) provide an introduction to governance models (http://oss-watch.ac.uk/resources/governancemodels).

Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?
Especially on academic projects, users will view the active lifetime of software to be the duration of the software's project funding. If you want to persuade users that your software will be around in the future, it is a good idea to describe your funding model and the duration over which funding is assured.

Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?
It's never a good idea to remove components or features without giving your users an advance warning first. It could be there are users who are dependent on the feature(s) you plan to change or remove. Announcing such planned deprecations well in advance means users and developers can respond if a given feature is important to them.
If a feature is due to be superseded by a newer, better feature or component, including both for a suitable period within the software can allow your users to transition comfortably from the older version to the new version.
You could also consider developing and publicising a deprecation policy, stating how and when features or components in general are deprecated. This gives your users assurance that features will not be removed without warning. see, for example the Eclipse API deprecation policy. (https://wiki.eclipse.org/Eclipse/API_Central/Deprecation_Policy).