

# Software Engineering Project

Name:	Email ID
Prateek Kamath	<a href="mailto:pnkamath@ncsu.edu">pnkamath@ncsu.edu</a>
Akarsh Reddy	<a href="mailto:aeatham@ncsu.edu">aeatham@ncsu.edu</a>
Himashu Singh	<a href="mailto:hsingh25@ncsu.edu">hsingh25@ncsu.edu</a>

## Link To the Github repository:

<https://github.com/Software-Engineering-Project-PKHSAK/To-Done>

## Poster:



# TO-DONE

Task Management Made Simple: Collaborate, Organize, and Achieve!  
Group -53 Akarsh Reddy, Himanshu Singh, Prateek Kamath

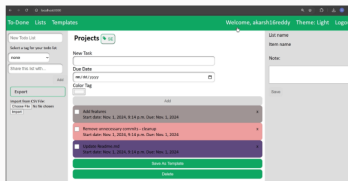


Repo link

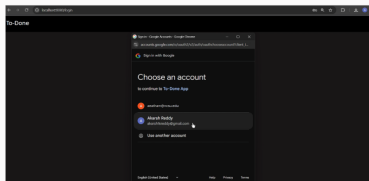
## INTRODUCTION

- To-Done is the ultimate task management app designed to streamline your day and supercharge your productivity.
- With To-Done's intuitive interface, organizing, prioritizing, and tracking tasks has never been easier.
- Whether you're a student juggling assignments, a professional managing projects, or just looking to simplify your to-do list, To-Done fits your needs.
- Designed to help you stay on track, To-Done gives you all the tools you need to maximize your focus and efficiency.
- Start transforming the way you manage tasks —join To-Done today and elevate your productivity game!

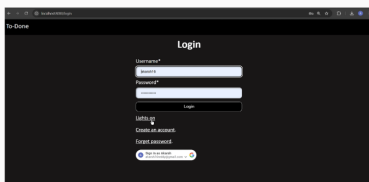
## OUTPUTS



Homepage with all tasks



Sign-in via Google

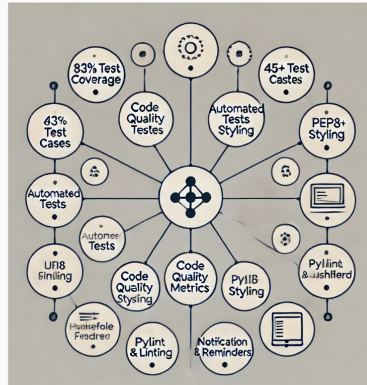


Page in Dark Mode

## JOIN THE TO-DONE REVOLUTION!

Boost your productivity with To-Done! Take control of your tasks and streamline your daily routine in one organized space. Say goodbye to chaos and prioritize effectively to achieve your goals. We welcome your feedback to help us enhance To-Done as the ultimate task management solution. Start your journey to a more efficient life today!

## ACHIEVEMENTS



- 83% Test Coverage:** Reliable, tested code with extensive coverage.
- 45+ Test Cases:** Rigorous testing ensures robust performance.
- Code Quality Metrics Powered by SonarCloud:** Automated metrics for quality you can count on.
- Automated Tests with GitHub Actions:** Seamlessly integrated and tested regularly.
- PEP8 Code Styling:** Professional code that follows industry standards.
- PyLint Linting:** Every line meets the highest linting standards.
- Fully Documented Code & Tests:** Clear, detailed documentation for easy onboarding.
- Makefile for Fast Setup & Running:** Get started instantly with a single command.
- Exciting Future Scope!**
  - UI Revamp
  - Timer Features
  - Notification & Reminder Integration
  - Dashboard Implementation
  - Mobile App Extension



## FEATURES

- Current Version (1):**
  - Register and Login
  - Create, Update, Delete Todo Lists
  - Quickly Create Todo Lists From Existing Templates
  - Create Your Own Templates
  - Shared List
  - Add Due Date To Tasks
  - Due Date Alerting Mechanism
  - Add Reminder Message to Task Completed
  - Customized Color Tag
  - Add Tags To Todo Lists For Customizable Grouping
- Our Version (1+1)**
  - Social Login - Google Sign-in
  - Bug Fixes (General maintenance and improvements)
  - Dark Mode
  - Import/Export Todo Feature.
- Delta**
  - New Features Introduced:**
    - Import/Export Todo Feature:** Users can import existing todo lists from other platforms or export their current lists for backup or sharing purposes.
    - Social Login - Google Sign-in:** Streamlined login process for users using Google accounts.
    - Dark Mode:** Enhanced user interface for better usability in low-light conditions.
  - Maintenance:**
    - Bug Fixes:** Addressing existing issues to improve overall performance and stability.
  - Testing:**
    - Added about 30 testcases

## FUTURE SCOPE

- 3-Month Tasks:**
  - Enhance task creation interface with due dates and priority levels.
  - Implement basic task editing and deletion functionalities.
  - Conduct user testing to gather feedback on the current UI.
- 6-Month Tasks:**
  - Develop shared tasks functionality for project collaboration.
  - Implement a commenting system for tasks.
  - Introduce tagging and categorization for tasks.
- 12-Month Tasks:**
  - Develop and launch a mobile application for iOS and Android.
  - Implement a robust search feature for tasks.
  - Develop filtering options based on date, priority, and category.
  - Allow users to customize their dashboard layout and displayed information.

## Rubric.md:

	Score	Evidence
Workload is spread over the whole team (one team member is often Xtimes more productive than the others...	3	git commits
but nevertheless, here is a track record that everyone is contributing a lot)		
Number of commits	3	git commits
Number of commits: by different people	3	git commits
Issues reports: there are many	3	project board
Issues are being closed	3	project board
Docs: doco generated, format not ugly	3	README.md & Documentation website posted in README
Docs: what: point descriptions of each class/function (in isolation)	3	Documentation website posted in README.md
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	GIFs in README.md
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	
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	Available in README.md
Use of version control tools	3	git
Test cases exist	3	unit_tests.yml workflow and coveralls code coverage
Test cases are routinely executed	3	unit_tests.yml workflow
Issues are discussed before they are closed		
Chat channel: exists	3	Google Chat,Whatsapp
Test cases: a large proportion of the issues related to handling failing cases.		
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	VScode, Python, Pip, Django
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	editorconfig
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	
Evidence that the members of the team are working across multiple places in the code base	3	Git commits
Short release cycles	1	
The file .gitignore lists what files should not be saved to the repo. See [examples]i( <a href="https://github.com/github/gitignore">https://github.com/github/gitignore</a> )	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	Score	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		
Video	3	README.md
DOI badge: exists. To get a Digital Object Identifier, register the project at Zenodo. DOI badges look like this: Zenodo doi badge	3	README.md
Badges showing your style checkers	3	README.md - pylint
Badges showing your code formatters.	3	README.md - autopep8
Badges showing your syntax checkers.	3	README.md - pyflakes
Badges showing your code coverage tools	3	README.md - coveralls
Badges showing any other Other automated analysis tools	3	README.md - sonarcloud quality gate
Q1 - What your software does		
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software? *	3	README.md
Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?*	3	README.md
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?*		
Q2 - Your project's and software's identity		
Question 2.1: Is the name of your project/software unique?*	3	README.md
Question 2.2: Is your project/software name free from trademark violations?*	3	ToDo is a unique name
Q3 - Availability of your software		
Question 3.1: Is your software available as a package that can be deployed without building it?*		
Question 3.2: Is your software available for free?*	3	Yes, under MIT License
Question 3.3: Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?*	3	Github Repo
Question 3.4: Is your software hosted in an established, third-party repository like GitHub ( <a href="https://github.com">https://github.com</a> ), BitBucket ( <a href="https://bitbucket.org">https://bitbucket.org</a> ), LaunchPad ( <a href="https://launchpad.net">https://launchpad.net</a> ) or SourceForge ( <a href="https://sourceforge.net">https://sourceforge.net</a> )?*	3	on GitHub
Q4 - Your software's documentation		
Question 4.1: Is your documentation clearly available on your website or within your software?*	3	Readme.md, Docstrings in Repository & Documentation website link located in Readme.md
Question 4.2: 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?*	3	Readme.md

Question 4.3: If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?*	Score 3	Evidence Readme.md
Question 4.4: Do you provide a comprehensive guide to all your software's commands, functions and options?*	3	Makefile
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?*		
Question 4.6: If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?*	3	Developer documentation link available in Readme.md
Question 4.7: Do you store your documentation under revision control with your source code?*	3	Readme.md
Question 4.8: 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?*	3	Releases
Q5 - How you support your software		
Question 5.1: Does your software describe how a user can get help with using your software?*	2	README mentions they can get help via creating issues
Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?*		
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?*		
Question 5.4: Are e-mails to your support e-mail address received by more than one person?*		
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?*	3	Project board
Question 5.6: Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?*	3	Project board is public
Q6 - Your software's maintainability		
Question 6.1: Is your software's architecture and design modular?*	3	uses Django, MVC Architecture
Question 6.2: Does your software use an accepted coding standard or convention?*	3	mandates code lint, style, formatting & docstrings. Dedicated workflows to verify
Q7 - Open standards and your software		
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	3	Feature to import & export CSV formatted TODO lists is built, demo in README.md
Question 7.2: Does your software allow communications using open communications protocols?* e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones		
Q8 - Your software's portability		
Question 8.1: 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?	3	Browser-compatible
Q9 - Your software and accessibility		
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?*		
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?*		

Q10 - How you manage your source code	Score	Evidence
Question 10.1: Is your source code stored in a repository under revision control?*	3	
Question 10.2: Is each source code release a snapshot of the repository?*	2	Releases - snapshot of specific tag
Question 10.3: Are releases tagged in the repository?*	3	Releases
Question 10.4: Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)*		
Question 10.5: Do you back-up your repository?*	3	into Google Drive & Locally
Q11 - Building and installing your software		
Question 11.1: Do you provide publicly-available instructions for building your software from the source code?*	3	Readme.md & Makefile
Question 11.2: Can you build, or package, your software using an automated tool?* e.g. Make ( <a href="https://www.gnu.org/software/make/">https://www.gnu.org/software/make/</a> ), ANT ( <a href="http://ant.apache.org/">http://ant.apache.org/</a> ), Maven ( <a href="https://maven.apache.org/">https://maven.apache.org/</a> ), CMake ( <a href="https://cmake.org/">https://cmake.org/</a> ), Python setuptools ( <a href="https://pypi.python.org/pypi/setuptools">https://pypi.python.org/pypi/setuptools</a> ), or R package tools ( <a href="https://cran.r-project.org/doc/manuals/r-devel/R-exts.html">https://cran.r-project.org/doc/manuals/r-devel/R-exts.html</a> )	3	MakeFile setup command does install, migrate and run
Question 11.3: Do you provide publicly-available instructions for deploying your software?*	3	Readme.md
Question 11.4: Does your documentation list all third-party dependencies?*	3	requirements.txt
Question 11.5: Does your documentation list the version number for all third-party dependencies?*	3	third_party.md
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?*	3	third_party.md
Question 11.7: Can you download dependencies using a dependency management tool or package manager?* e.g. Ivy ( <a href="http://ant.apache.org/ivy/">http://ant.apache.org/ivy/</a> ), Maven ( <a href="https://maven.apache.org/">https://maven.apache.org/</a> ), Python pip ( <a href="https://pypi.python.org/pypi/pip">https://pypi.python.org/pypi/pip</a> ) or setuptools ( <a href="https://pypi.python.org/pypi/setuptools">https://pypi.python.org/pypi/setuptools</a> ), PHP Composer ( <a href="https://getcomposer.org/">https://getcomposer.org/</a> ), Ruby gems ( <a href="https://rubygems.org">https://rubygems.org</a> ), or R PackRat ( <a href="https://rstudio.github.io/packrat/">https://rstudio.github.io/packrat/</a> )	3	pip install -r requirements.txt
Question 11.8: 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?*	3	unit_test.yml workflow on GitHub, 'test' command in MakeFile, also badge on README
Q12 - How you test your software		
Question 12.1: Do you have an automated test suite for your software?*	3	unit_test.yml workflow
Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?*		
Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?*	3	Django
Question 12.4: Are your test results publicly visible?*	3	via badges & actions
Question 12.5: Are all manually-run tests documented?*	2	Code comments inside tests
Q13 - How you engage with your community		
Question 13.1: 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		

Question 13.2: Does your website state how many projects and users are associated with your project?*	Score	Evidence
Question 13.3: Do you provide success stories on your website?*		
Question 13.4: Do you list your important partners and collaborators on your website?*		
Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?*		
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?*		
Question 13.7: Can users subscribe to notifications to changes to your source code repository?*	3	GitHub Watch Option
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?*		
Q14 - How you manage contributions		
Question 14.1: Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?*	3	Yes, details provided in CONTRIBUTING.md
Question 14.2: Do you have a contributions policy?*	3	CONTRIBUTING.md
Question 14.3: Is your contributions' policy publicly available?*	3	CONTRIBUTING.md
Question 14.4: Do contributors keep the copyright/IP of their contributions?*		
Q15 - Your software's copyright and licensing		
Question 15.1: Does your website and documentation clearly state the copyright owners of your software and documentation?*	3	LICENSE.md
Question 15.2: Does each of your source code files include a copyright statement?*	3	LICENSE.md
Question 15.3: Does your website and documentation clearly state the licence of your software?*	3	Readme.md
Question 15.4: Is your software released under an open source licence?*	3	Readme.md
Question 15.5: Is your software released under an OSI-approved open-source licence?*	3	MIT
Question 15.6: Does each of your source code files include a licence header?*	3	
Question 15.7: Do you have a recommended citation for your software?*	3	CITATION.md
Q16 - Your plans for the future		
Question 16.1: Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?*	3	Readme.md
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?*	3	Readme.md
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?*		