Group No 38 Project2

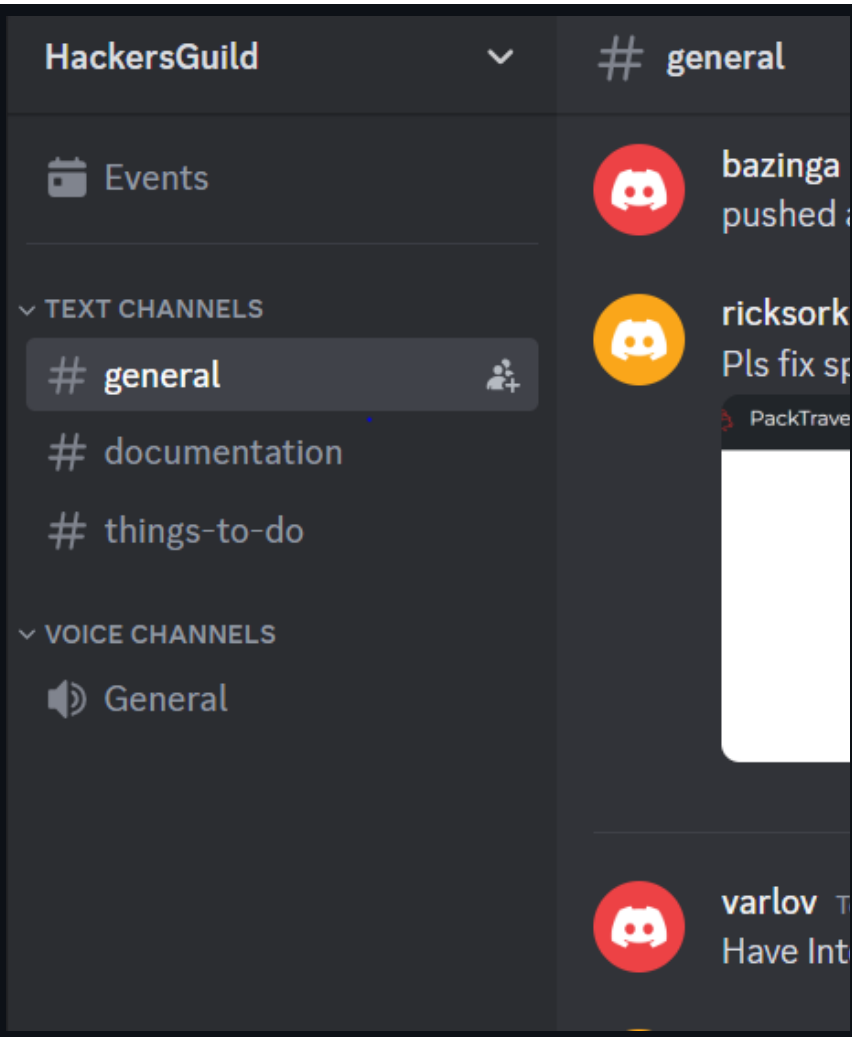
Makarand Pundlik (mpundli)
Michelle Varghese (mmvarghe)
Varun Varatharajan (vvarath)

Github Repo : Link

Poster submitted on moodle.
Rubric – (also added in github)

Rubric	Score (258)	Evidence
Workload is spread over the whole team (one team member is often Xtimes more productive than the others...	3	Git Commits Distribution 
but nevertheless, here is a track record that everyone is contributing a lot)	3	GH
Number of commits	3	evidence in GH
Number of commits: by different people	3	evidence in GH
Issues reports: there are many	3	evidence in projects - issue board https://github.com/orgs/GradHackersGuild/projects/2/views/1

Issues are being closed	3	evidence in README badges and project board
Docs: doco generated, format not ugly	3	in GH
Docs: what: point descriptions of each class/function (in isolation)	3	Docstrings in code
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	demo video entries
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	Poster
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	Video in readme
Use of version control tools	3	commit history
Test cases exist	3	tests/ under each dir
Test cases are routinely executed	3	Github Actions
Issues are discussed before they are closed	3	Discussed on discord
Chat channel: exists	3	Discord Channel



Discord channel		
Test cases: a large proportion of the issues related to handling failing cases.	3	Failed tests are fixed in the next commit
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	same codebase. everyone pulls and spins the main_new version to test on local
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	same config files for keys and db details
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	share screen

Evidence that the members of the team are working across multiple places in the code base	3	commit history
Short release cycles	3	commit history
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	in GH
The file INSTALL.md lists how to install the code	3	in GH
The file LICENSE.md lists rules of usage for this repo	3	in GH
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	in GH
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		in GH
The file README.md contains all the following		
Video	3	2min video of new functionality, showing a significant delta from prior.
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	badges in README
Badges showing your code formatters.	3	badges in README
Badges showing your syntax checkers.	3	badges in README

Badges showing your code coverage tools	3	badges in README
Badges showing any other Other automated analysis tools	1	Line Count badge
What is the name of your software?*		PackTravel
Q1 - What your software does		
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software?*	3	Poster in Documentation
Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?*	3	Poster
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?*	0	No
Q2 - Your project's and software's identity		
Question 2.1: Is the name of your project/software unique?*	3	Yes
Question 2.2: Is your project/software name free from trademark violations?*	3	Checked on Go to the United States Patent and Trademark Office (USPTO) website.
Q3 - Availability of your software		
Question 3.1: Is your software available as a package that can be deployed without building it?*	0	
Question 3.2: Is your software available for free?*	3	Yes

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	GH repo
Question 3.4: Is your software hosted in an established, third-party repository likeGitHub (https://github.com), BitBucket (https://bitbucket.org),Launch Pad (https://launchpad.net) orSourceForge (https://sourceforge.net)?*	3	GH
Q4 - Your software's documentation		
Question 4.1: Is your documentation clearly available on your website or within your software?*	3	Yes
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 Install
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?*	3	Readme Install
Question 4.4: Do you provide a comprehensive guide to all your software’s commands, functions and options?*	3	Video
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error	0	

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?*	0	No
Question 4.7: Do you store your documentation under revision control with your source code?*	3	Readme
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 on GH
Q5 - How you support your software		
Question 5.1: Does your software describe how a user can get help with using your software?*	3	contact developers
Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?*	3	Developers info
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?*	3	Contributing.md
Question 5.4: Are e-mails to your support e-mail address received by more than one person?*	0	
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?*	3	issues 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	Public Link https://github.com/orgs/GradHackersGuild/projects/2/views/1
Q6 - Your software's maintainability		
Question 6.1: Is your software's architecture and design modular?*	3	Django is MVC
Question 6.2: Does your software use an accepted coding standard or convention?*	3	1. SOLID Principles. 2. DRY not WET 3. UUID added to index database (db principles)
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	0	No
Question 7.2: Does your software allow communications using open communications protocols?* e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones	3	HTTP
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	Platform Independent, browser independent

Q9 - Your software and accessibility		
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?*	0	no
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?*	0	WCAG (Web Content Accessibility Guidelines) 2.1, Section 508 compliance
Q10 - How you manage your source code		
Question 10.1: Is your source code stored in a repository under revision control?*	3	GH
Question 10.2: Is each source code release a snapshot of the repository?*	3	Releases provide source code download
Question 10.3: Are releases tagged in the repository?*	3	Check releases section
Question 10.4: Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)*	3	main_new branch is stable
Question 10.5: Do you back-up your repository?*	3	stored locally and on gdrive
Q11 - Building and installing your software		
Question 11.1: Do you provide publicly-available instructions for building your software from the source code?*	3	Install in readme
Question 11.2: Can you build, or package, your software using an automated tool?* e.g. Make (https://www.gnu.org/software	3	Makefile

/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)		
Question 11.3: Do you provide publicly-available instructions for deploying your software?*	3	Install in readme
Question 11.4: Does your documentation list all third-party dependencies?*	3	Requirements.txt and installation steps in readme
Question 11.5: Does your documentation list the version number for all third-party dependencies?*	3	Same as above
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	Same as above
Question 11.7: Can you download dependencies using a dependency management tool or package manager?*	3	pip install -r requirements.txt
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/pack	3	pip install -r requirements.txt Setup Venv

rat/)		
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	GH Workflows
Q12 - How you test your software		
Question 12.1: Do you have an automated test suite for your software?*	3	GH Workflows
Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?*	3	
Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?*	3	Run tests on push run_test_cases.yml workflow file
Question 12.4: Are your test results publicly visible?*	3	GH WOrkflows
Question 12.5: Are all manually-run tests documented?*	0	
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?*	0	
Question 13.2: Does your website state how many	1	Included contributors

projects and users are associated with your project?*		
Question 13.3: Do you provide success stories on your website?*	0	
Question 13.4: Do you list your important partners and collaborators on your website?*	3	Developers / contributors
Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?*	0	
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?*	0	
Question 13.7: Can users subscribe to notifications to changes to your source code repository?*	3	GH 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?*	0	
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	forking
Question 14.2: Do you have a contributions policy?*	3	Contributing.md
Question 14.3: Is your contributions' policy publicly	3	Same as above

available?*		
Question 14.4: Do contributors keep the copyright/IP of their contributions?*	0	No
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	code of conduct
Question 15.2: Does each of your source code files include a copyright statement?*	3	Source code file headers
Question 15.3: Does your website and documentation clearly state the licence of your software?*	3	License.md
Question 15.4: Is your software released under an open source licence?*	3	MIT License
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	MIT License header
Question 15.7: Do you have a recommended citation for your software?*	3	CITATION.rff
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	3	Poster and video

the next 3, 6 and 12 months)?*		
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?*	0	N/A
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?*	0	