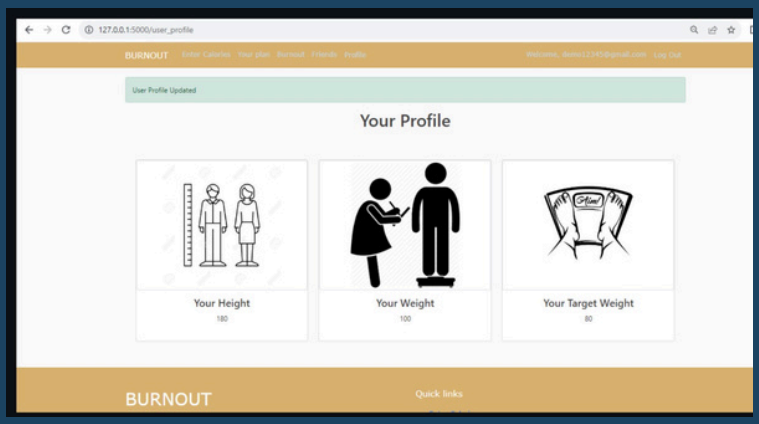
UNLOCK THE POWER OF METAFIT



What is MetaFit?

A platform-independent solution that provides AI-driven meal suggestions, real-time calorie tracking, and personalized health management. Achieve your fitness goals effortlessly, anytime and anywhere, with a scalable system designed to simplify your fitness journey.

5 ways that this software (Version i) is better

1. **Meal Plan Recommendation Feature:** Our Meal Plan Recommendation tool personalizes meal plans to match your unique goals—whether for weight loss, muscle gain, or maintenance. It analyzes your calorie and macronutrient preferences using advanced machine learning, creating meal suggestions that are both nutritious and diverse. Say goodbye to guesswork and make your dietary journey effortless, as this feature keeps your meals exciting and on target.
2. **Streak Feature:** The Streak Feature boosts engagement by rewarding users for daily logins, encouraging consistent activity over time. Each day you log in continues your streak, displayed with visual indicators and notifications that celebrate your progress. By adding a gamified element to the app, users stay motivated and develop a habit of daily tracking, making it a fun and engaging experience.
3. **Quiz Feature:** This Quiz Feature makes fitness learning interactive and competitive, offering users a chance to test their knowledge on topics like diet, calories, and exercise. Questions are dynamically drawn from a database, and users earn points for correct answers, building their score over time. A final score summary allows users to track their learning progress and performance, adding a layer of challenge and personal achievement.
4. **Social Sharing:** This feature enables users to share milestones and fitness progress to social media with ease. It generates personalized, shareable messages based on progress, like calories burned or challenges completed, allowing users to broadcast their achievements to friends and followers. Weekly email summaries are also sent automatically every Monday via Flask-APScheduler, formatted with HTML for easy sharing through social buttons, helping users stay motivated by sharing their journey highlights.
5. **Daily Challenge:** The Daily Challenge feature presents three random fitness challenges each day from a curated list, offering a mix of tasks that enhance daily workout variety. Users complete and track each challenge in MongoDB, which records completion status and gives personalized feedback. Once all challenges are completed, users receive a ready-to-share message to post on social media, making it easy to celebrate achievements with friends.

Implementation milestones for Version I + 1

1. **Real-Time Fitness Sync: Auto-Sync Your Wearables!** Seamlessly connect your fitness tracker (like Fitbit) to automatically log calories, workouts, and activity levels in real-time, providing a holistic view of your fitness journey. Enjoy instant updates without the hassle of manual entries, so you can focus on achieving your goals.
2. **Advanced Analytics: Predictive Health Insights!** Leverage smart analytics to understand your calorie trends and track your weight goals with personalized, predictive insights. Identify patterns over time and receive tailored recommendations that empower you to stay on track and reach your desired outcomes with confidence.
3. **Water Intake Tracking: Stay Hydrated, Stay Healthy!** Easily log your daily water intake, monitor hydration levels, and set personalized hydration goals. Receive timely reminders throughout the day, helping you stay refreshed, energized, and in peak condition to support your health and wellness goals.
4. **Goal-Driven Workout Recommendations: Tailored Workouts Just for You!** Define your fitness goals—whether it's strength, endurance, or flexibility—and receive custom workout recommendations suited to your objectives. Follow progression-based routines that adapt as you improve, making it easier to see results and stay motivated on your journey.

Tests, QR Code and Tech Stack

Ensuring Stability: 24 test modules (Link) guarantee reliable performance across key features with 6 new test modules including 60 test cases.

SCAN ME

CSS HTML JavaScript

GROUP60: APURV, PRABHUDATTA, SANKAR

Prabhudatta Mishra (pmishra4)
Sankar Raghuthaman (sraghut)

Apurv Choudhari(apchoudh)

GITHUB LINK: https://github.com/apurv-choudhari/calorieApp_server/

Requirements	**Self Assessment (Total = 283)**	**Links**
Workload is spread over the whole team (one team member is often Xtimes more productive than the others…	**3**	https://github.com/deekay2310/calorieApp_server/pulse
Number of commits	**3**	https://github.com/apurv-choudhari/calorieApp_server/graphs/commit-activity
Number of commits: by different people	**3**	https://github.com/apurv-choudhari/calorieApp_server/graphs/contributors
Issues reports: there are **many**	**3**	https://github.com/apurv-choudhari/calorieApp_server/issues
Issues are being closed	**3**	https://github.com/apurv-choudhari/calorieApp_server/issues?q=is%3Aissue+is%3Aclosed
Docs: doco generated, format not ugly	**3**	https://github.com/apurv-choudhari/calorieApp_server/tree/main/docs
Docs: what: point descriptions of each class/function (in isolation)	**3**	
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**	https://github.com/apurv-choudhari/calorieApp_server/edit/main/README.md
Docs: short video, animated, hosted on your repo. That convinces people why they want to	**3**	https://github.com/user-attachments/assets/ad8074ca-386d-49f0-b6ea-cd2ffd9b1d58

work on your code.		
Test cases exist	**3**	https://github.com/apurv-choudhari/calorieApp_server/tree/main/tests
Test cases are routinely executed	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/.github/workflows/Code_Coverage.yml

Issues are discussed before they are closed	**2**	Documented over Slack channel, whatsapp and in person
Chat channel: exists	**3**	https://join.slack.com/t/metafit-group/shared_invite/zt-2tbbnhyud-oBs4Bx31uYCPpnCpjMI2BA
Test cases: a large proportion of the issues related to handling failing cases.	**3**	https://github.com/apurv-choudhari/calorieApp_server/tree/main/tests
Evidence that the whole team is using the same tools: everyone can get to all tools and files	**3**	Github, Slack, Whatsapp

Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	**3**	https://github.com/apurv-choudhari/calorieApp_server/tree/main/.github/workflows
Evidence that the members of the team are working across multiple places in the code base	**3**	https://github.com/apurv-choudhari/calorieApp_server/graphs/commit-activity
Short release cycles	**2**	—----------
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/.gitignore
The file INSTALL.md lists how to install the code	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/INSTALLATION.md
The file LICENSE.md lists rules of usage for this repo	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/LICENSE
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/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**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/CONTRIBUTING.md
Video	**3**	video1302963082

DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this:	**3**	https://zenodo.org/records/10211870
Badges showing your style checkers	**3**	https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/Pylint.yml https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/flake8.yml
Badges showing your code formatters	**3**	https://github.com/apurv-choudhari/calorieApp_server/blob/main/.github/workflows/code_formatter.yml
Badges showing your syntax checkers	**3**	https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/flake8.yml https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/flake8.yml
Badges using code coverage tools	**3**	https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/Code_Coverage.yml
Badges for any other automated tools.	**3**	https://github.com/apurv-choudhari/calorieApp_server/actions/workflows/codeql.yml
Sustainibility Form		
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software?	**3**	Yes
Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?	**3**	Yes

Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?	**1**	Yes
Question 2.1: Is the name of your project/software unique?*	**2**	Yes
Question 2.2: Is your project/software name free from trademark violations?*	**2**	Yes
Question 3.1: Is your software available as a package that can be deployed without building it?	**3**	Yes
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**	yes
Question 3.4: 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)?	**3**	yes
Question 4.1: Is your documentation clearly	**3**	yes

available on your website or within your software?		
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?	**2**	yes
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**	yes
Question 4.4: Do you provide a comprehensive guide to all your software's commands, functions and options?	**3**	yes
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	**0**	No
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	**0**	No

store your documentation under revision control with your source code?		
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?	**0**	No
Question 5.1: Does your software describe how a user can get help with using your software?	**2**	Yes
Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?	**2**	Yes
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?	**0**	No
Question 5.4: Are e-mails to your support e-mail address received by more than one person?	**0**	No
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?	**0**	No
Question 5.6: Is your project's ticketing system	**0**	No

publicly visible to your users, so they can view bug reports and feature requests?		
Question 6.1: Is your software's architecture and design modular?*	**3**	Yes
Question 6.2: Does your software use an accepted coding standard or convention?	**2**	Yes
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**	Yes
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**	Yes
Question 8.1: Is your software cross-platform	**3**	Yes

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?		
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?	**1**	Yes
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?	**1**	Yes
Question 10.1: Is your source code stored in a repository under revision control?	**2**	Yes
Question 10.2: Is each source code release a	**0**	No

snapshot of the repository?		
Question 10.3: Are releases tagged in the repository?	**0**	No
Question 10.4: Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	**3**	Yes
Question 10.5: Do you back-up your repository?	**0**	No
Question 11.1: Do you provide publicly-available instructions for building	**3**	Yes

your software from the source code?		
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)	**3**	Yes
Question 11.3: Do you provide publicly-available instructions for deploying your software?	**3**	Yes

Question 11.4: Does your documentation list all third-party dependencies?	**3**	Yes
Question 11.5: Does your documentation list the version number for all third-party dependencies?	**3**	yes
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**	No

Question 11.7: 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/)	**3**	Yes
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?	**2**	Yes

Question 12.1: Do you have an automated test suite for your software?	**3**	Yes
Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	**0**	No
Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?	**3**	Yes

Question 12.4: Are your test results publicly visible?	**3**	Yes
Question 12.5: Are all manually-run tests documented?	**0**	No
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**	No
Question 13.2: Does your	**2**	Yes

website state how many projects and users are associated with your project?		
Question 13.3: Do you provide success stories on your website?	**0**	No
Question 13.4: Do you list your important partners and collaborators on your website?	**0**	No
Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?	**0**	Yes
Question 13.6: Do you list third-party publications	**0**	No

that refer to your software on your website or link to a resource where these are available?		
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**	No
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
Question 14.2: Do you have a contributions	**3**	Yes

policy?		
Question 14.3: Is your contributions' policy publicly available?	**3**	Yes
Question 14.4: Do contributors keep the copyright/IP of their contributions?	**3**	Yes
Question 15.1: Does your website and documentation clearly state the copyright owners of your software and documentation?	**3**	Yes
Question 15.2: Does each of your source code files include a copyright	**0**	No

statement?		
Question 15.3: Does your website and documentation clearly state the licence of your software? Question 15.4: Is your software released under an open source licence?	**3** **3**	Yes Yes
Question 15.5: Is your software released under an OSI-approved open-source licence?	**3**	Yes
Question 15.6: Does each	**1**	Yes

of your source code files include a licence header?		
Question 15.7: Do you have a recommended citation for your software?	**2**	Yes
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**	Yes
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	**0**	No

Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?	**0**	No