**TRACK 2 - CODE FOR MENTAL WELLNESS**

**TECHNICAL GUIDELINES**

Congratulations on being part of MANAS Codeathon 2023 and your desire to contribute to app ideas for Mental wellbeing for our society.

1. *Interaction with your allotted Mentor and team member(s):* Create effective communication channel with your Mentor and team members through online platforms, phone and/or WhatsApp and active participation.
2. *Understand the Problem Statement assigned to your team:* Carefully read and comprehend the problem statement or challenge requirements. Seek clarifications if anything is unclear with your mentor.
3. *Plan and Strategize:* Spend time planning your approach before diving into designing. Discuss the problem with your team and brainstorm possible solutions. Define a clear strategy, and outline the implementable steps.
4. *Choose the Right Tools and Technologies:* Select appropriate frameworks, libraries, or tools based on the problem requirements. Opt for technologies that you are familiar with, as MANAS Codeathon is time-sensitive and not the best time to experiment with unfamiliar tools. Some opensource and free tools are listed here, however you are free to use any opensource/free tools that does not add their product watermark and adhere to the format
5. Format: zip folder consists of html/ css /js / jpeg /png/ mp4/ mp3

Some suggested tools:

* 1. **HTML Editor**
     + **HTML Builder: Layout it:** Access Link: <https://www.layoutit.com/build>

Access Link: <https://www.layoutit.com/>

* 1. **Integrated Development Environment (IDEs)**
     + **Visual Studio Code:** Download Link: <https://code.visualstudio.com/download>
     + **Intellij IDEA -** The leading Java and Kotlin IDE

Download Link: <https://www.jetbrains.com/idea/download/?section=windows>

* + - **Bluefish Editor:** Download Link: <https://bluefish.openoffice.nl/download.html>
    - **Notepad++:** Download Link: <https://notepad-plus-plus.org/downloads/>
  1. For Video Editing:
     + **VN Editor for Mobile app:** Download Link:

<https://play.google.com/store/apps/details?id=com.frontrow.vlog&hl=en_IN&gl=US&pli=1>

* + - **OpenShot for PC:** Download Link: <https://www.openshot.org/download/>

Tutorial: <https://www.youtube.com/watch?v=rLWXlc1BW-I>

* + - Display resolution: 1080p, 720p
  1. For Image Editing (one content per tool):
     + **Krita:** Download Link: <https://krita.org/en/>
     + **GIMP:** Download Link: <https://www.gimp.org/downloads/>
  2. For checking responsiveness of HTML template
     + **Responsive Design mode in Browsers**
     + **Google Chrome:** Open the developer’s tools (press F12 and select ‘inspect’ and toggle the device toolbar icon (Ctrl + Shift + M).
     + **Mozilla Firefox:** Open the developer’s tools (press F12 and select ‘Inspect Element’ and toggle the responsive design mode icon (Ctrl + Shift + M).
     + **Screenfly:** Access Link: <https://screenfly.org/>

1. *Modular and Maintainable Design:* Ensure clean, modular, and well-documented design. Use proper naming conventions, add comments and follow best practices for readability. Break down your design into functions or modules that perform specific tasks.
2. *Version of Code:* Ensure working on latest and correct version of code and no duplicity/repetition of code/tasks by other members, when working to facilitate seamless teamwork.
3. *Test and Validate:* Validate your design as you progress. Your design should be implementable. Write unit tests to ensure that individual components or functions are working correctly. Regularly run test cases to identify and fix bugs early on. Automate testing where possible to save time and ensure reliability.
4. *Time Management:* Time is limited, final submission is on 20-08-2023, manage your time wisely. Break your work into smaller milestones and set achievable goals within specific time frames. Regularly reassess your progress and adjust your strategy if needed, in concurrence with your mentor.
   1. Initial discussions, design, plan, and familiarizing with tools/ technologies – One week
   2. Development & Coding – Two weeks
   3. Developer Testing and Debugging – One week
   4. Submission of final Design in required format and documentation after validation from the mentor.
5. *Intellectual Property Rights (IPR):* Check the usage right of images/icons, size, color, type, etc. used in your app and include reference/ citations, when required, from the authentic and reliable source(s).
6. *Stay Calm and Manage Stress:* Enjoy the experience, learn from your peers, and celebrate your accomplishments, irrespective of the final result.

**SECURITY GUIDELINES**

1. First and foremost is to understand the rules and guidelines
2. Use secure devices and networks
3. Avoid utilizing external drives without antivirus scanning
4. Be cautious of security related attacks e.g Phishing attacks, because Codeathon will be the prime targets for phishing attempts.
5. Vigilant of suspicious emails, messages or links that may try to trick you into revealing sensitive information.
6. Protect sensitive data by avoid using the real user data.
7. Use secure communication platforms when discussing sensitive information
8. Do not use or incorporate copyrighted materials without proper authorization.
9. During the event, notify the organizers or designated security personnel of any security issues or vulnerabilities found.
10. Avoid exploiting the known vulnerabilities and expose the environment.
11. Before installing any tools, carefully review the terms and conditions and the data privacy statement.
12. Use malware or antivirus protection.
13. Follow *OWASP Top 10 Secure Coding Practices* to prevent common vulnerabilities such as XSS Attack, Injection Attacks, etc. Properly sanitize the user inputs and validate the user data. (Refer next page).
14. Secure third-party integrations: If your application integrates with external APIs, ensure that proper authentication and authorization mechanisms are in place.
15. Handle errors properly: Implement proper error handling mechanism and log errors securely without revealing sensitive information.

**COMMON SECURE CODING PRACTICES**

1. Input Validation: All user inputs, including form submissions, query parameters, and file uploads, must be validated and sanitized. Use server-side validation to check for the type, format, and length of expected input.
2. Parameterized queries and prepared statements: Instead of concatenating user-supplied data straight into SQL queries, use parameterized queries or prepared statements with placeholder values. This prevents SQL injection Attacks.
3. Ensure Strong authentication methods, such as password hashing and salting, multi-factor authentication, and secure session management, should be implemented. For handling user credentials and sessions, use industry-standard methods and libraries.
4. Cross-Site Request Forgery (CSRF) protection: To protect against CSRF attacks, use CSRF tokens. Tokens should be generated for each user session and included in forms or AJAX queries. Verify the token on the server side to ensure the request is legitimate.
5. Secure communication: Use secure protocols like HTTPS/TLS to encrypt data in transit between the client and the server. Avoid transmitting sensitive information, such as passwords or authentication tokens, over unsecured channels.
6. Error handling and logging: Use secure error handling procedures to avoid revealing sensitive information to users. Log errors and exceptions in a secure manner, and avoid logging sensitive data like as passwords or user information or version of the software used etc.
7. Implement proper file upload handling mechanisms. Validate file types, limit file size, and store files in a location outside the web root directory. Ensure proper access controls are in place for uploaded files.
8. Ensure with the latest security patches and updates for all frameworks, libraries, and components used in the application.
9. XSS Attack Prevention (Cross-site Scripting): To prevent XSS attacks, your application must validate all the input data, make sure that only allow listed data is allowed, and ensure that all variable output in a page is encoded before it is returned to the user. When you encode variable output, you substitute HTML mark up with alternative representations that are called entities. The browser displays the entities but doesn't run them. When a browser renders the entities, they're converted back to HTML and printed but they aren't run.
10. Least Privilege: Ensure that each component of your web application has the minimum level of access privileges required to perform its functions. This reduces the potential impact of a security breach.
11. Secure Configuration Management: Ensure that your web application and its underlying components are securely configured, adhering to best practices and deleting unneeded functionality or default credentials.

**Ready References for User/Style guides**

1. Oreilly’s The little book on HTML/CSS Coding Guidelines <PDF attached>
2. <https://developer.mozilla.org/en-US/docs/Learn/HTML>

<https://docs.ckan.org/en/2.10/contributing/html.html>

1. <https://developer.mozilla.org/en-US/docs/Web/CSS>

<https://docs.ckan.org/en/2.10/contributing/css.html>

1. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>

<https://docs.ckan.org/en/2.10/contributing/javascript.html>

**SAMPLE TEMPLATE FOR CONTENT DEVELOPMENT**

**Content Theme:** Self Regulation

**Content Name**: **Daily Self Care**

**Technologies Used**: Figma/ Quant-UX

**Problem Statement**: Your goal is to create an interface that promotes emotional well-being, encourages self-care, and provides access to helpful resources.

**Stage 1: DESIGN for “Daily Self Care”**

**Questions to ponder for Mock UI Design**

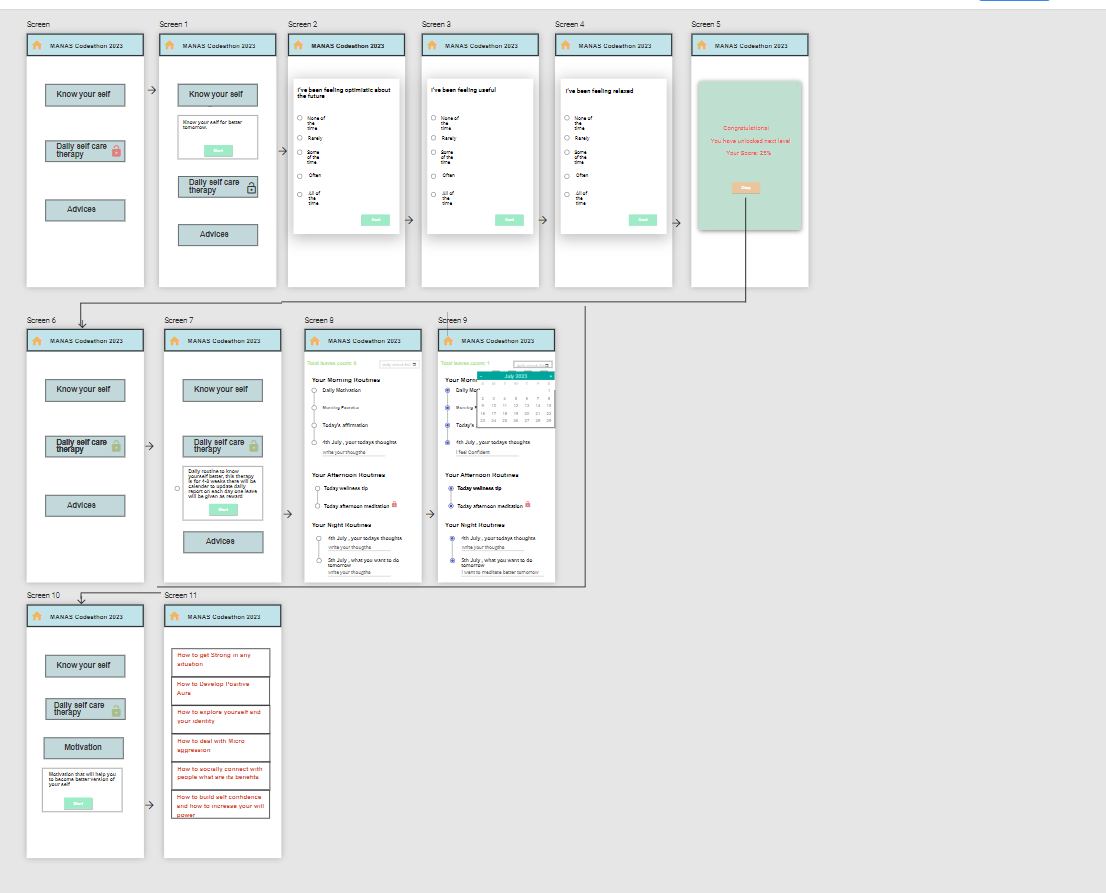
1. Purpose for creating “Self Regulation” and defining its Scope
2. Benefit to the user – Justify the reason user should use this module/app
3. Design the app - Steps for creating the app (screen-wise)
4. Familiarize with Quant-UX/ Figma
   * How to Use Quant UX for User Interface Design Development?

**Tutorial**: <https://www.youtube.com/watch?v=_47HCDPKa4g>

1. Break down problem statement into three sections for “Self Regulation”:
   * Section 1 - Promotes emotional well-being - **Motivation (Information)**
   * Section 2 - Provide access to helpful resources - **Know yourself (Measurement)**
   * Section 3 - Encourages self-care – **Daily Self Care (Intervention)**

**Section 1 - Motivation (Information) for “Daily Self Care”**

The mentor shall guide the team to appropriate resources with references and citations on “Self Regulation”. Identify crucial points on “Self Regulation” that appear for each screen of the app. List the number and order of references with appropriate contents. Refer Figure 1, Screen 10 to 11 (starting from Screen or Screen 5).



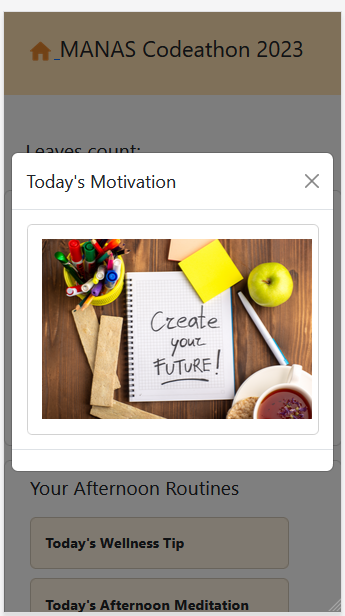
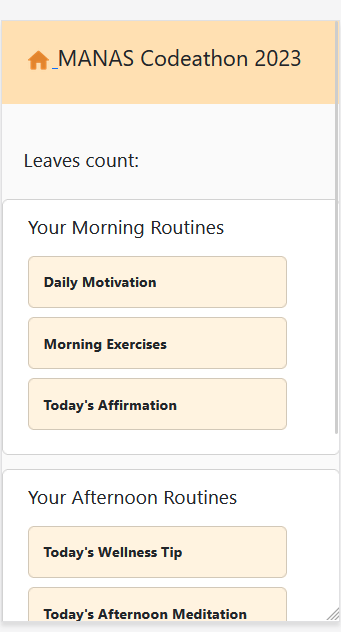
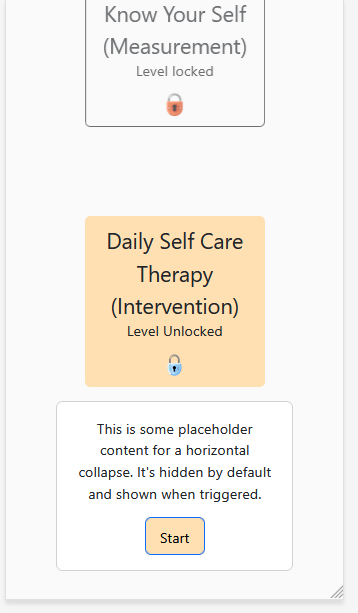
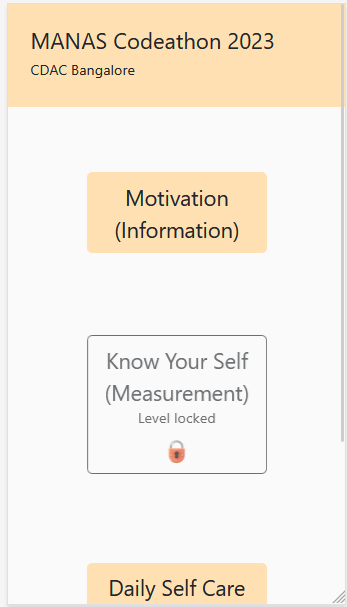
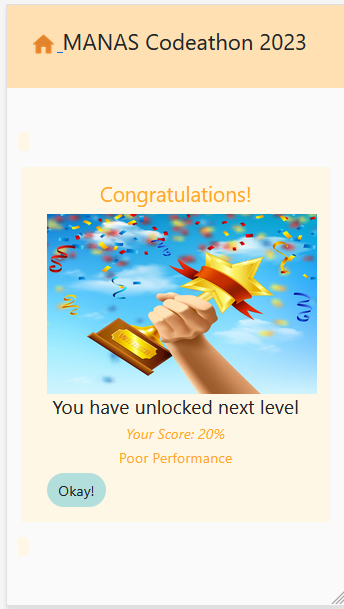
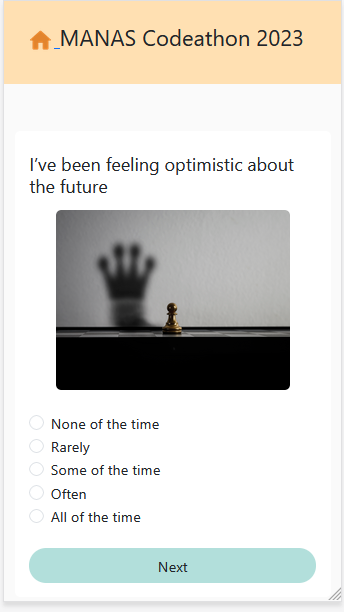
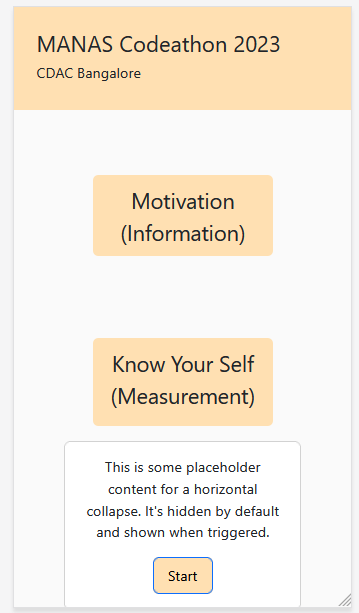
*Figure 1: UI Design for “Self Regulation”*

**Section 2 - Know Yourself (Measurements) for “Daily Self Care”**

The content when in questionnaire format, write down each question with appropriate options and submit button in concurrence with your mentor. The team should design keeping in mind, the font, color, size, line spacing, items (ex: Ratio buttons/ Checkboxes/ Submit buttons/Next or Previous button/ Lock and Unlock icon, if any), contents for every screen and navigation from each screen. Refer Figure 1, Screen 1 to 5.

The contents should be engaging for the users to earn scores. The detail of scoring system should be documented for calculation purpose. The user has to complete assessment and scores shall decide unlocking of Section 3.

When the user score is above 70 percent, Section 3 shall remain locked. The Section 3 shall be unlocked only when the user scores is below 70 percent for intervention. The user is requested to practice the instruction provided in Section 3.



*Figure 2: Sample screen for “Daily Self Care”*

**Section 3 – Intervention for “Daily Self Care”**

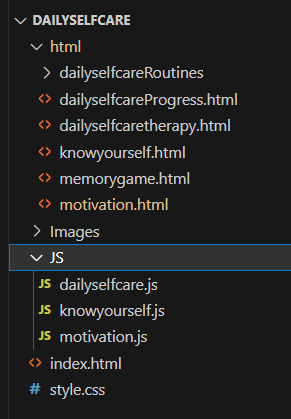
The team shall design the simple and effective intervention steps for “Daily Self Care”. The benefits of follow through should be concise for user to practice and follow through. The contents should be in agreement with your mentor. The team should list number of interventions, simplicity, time frame (example: number of days/weeks) for user to practices and reevaluation through Section 2. Refer Figure 1 Screen 6 to 9.

The final reward for user is fully grown tree with lots of leaf. The user should receive a leaf as reward upon completion of each task given in Section 3 Invention for Daily Self Care. Daily Self Care is an example of content for which a game-like interface needs to be developed.

**Stage 2: Development of “Daily Self Care”**

The team shall start coding after the design is approved from their mentor. The coding for “Daily Self Care” involves following steps:

1. Create folder structure open with IDE (Visual Studio Code).



*Figure 3: Folder Structure for “Daily Self Care”*

1. Create one main html file *index.html* through which all other sections files shall be linked. Create a folder for HTML files, i.e., *dailyselfcare.html, motivation.html, knowyourself.html*. Create a folder for JavaScript files, i.e, *dailyselfcare.js, motivation.js, knowyourself.js* containing logic and functions call to be coded by the team and CSS *style.css.*
2. In the Section 2 (Know Yourself) of Stage 1 (Design), the user is given five options for each question. There are five questions in the questionnaire and the user is assigned the final scores upon selecting his/her response for each question. The next button takes the user to next question and the final submit button at the end of 5th question evaluates the user score. The same is explained through the code given below.
3. Inside the HTML <input/> tag, use value attribute <input value=”1”/>. When user clicks Submit button, the submit handler function is invoked and using parseInt(document.myform.answer1.value);
4. Store all values selected by user in variable “total”, calculate the percentage and display remarks according to the percentage.

let ans1 = parseInt(document.myform.answer1.value);

let ans2 = parseInt(document.myform.answer2.value);

let ans3 = parseInt(document.myform.answer3.value);

let ans4 = parseInt(document.myform.answer4.value);

let ans5 = parseInt(document.myform.answer5.value);

let total = 0;

total = ans1 + ans2 + ans3 + ans4 + ans5;

let percent = total \* 4;

knowyourselfResult = percent;

// percent < 20%

if (total < 13 || ans1 < 2 || ans2 < 2 || ans3 < 2 || ans4 < 2 || ans5 < 2) {

recommend.innerHTML = "Poor Performance ";

}

// percent 20% - 40%

else if (percent >= 20 && percent < 40) {

recommend.innerHTML = "Poor Performance";

}

// percent 40% - 60%

else if (percent > 40 && percent <= 60) {

recommend.innerHTML = "Average Performance";

} else if (percent > 60 && percent <= 80) {

recommend.innerHTML = "Good Performance";

}

// percent > 80%

else if (percent > 80) {

recommend.innerHTML = "Good Performance";

}

document.getElementById("resultImg").src = "../Images/trophy.jpg";

score.innerHTML = "Your Score:" + " " + percent + "%";

appData = {

question1: que1,

answer1: r1,

question2: que2,

answer2: r2,

question3: que3,

answer3: r3,

question4: que4,

answer4: r4,

question5: que5,

answer5: r5,

percent: percent,

comment: comment

};

The developer is requested to write appropriate comments for each module.

**Stage 3: Testing of “Daily Self Care”**

The team shall test the individual unit of code after development and complete testing after integration of the all modules. The testing of User Interface can be done using:

1. For Checking Responsiveness of HTML template: Responsive Design mode in browsers;
   * Google Chrome: Open the Developer’s tools (press f12 and select ‘inspect’ and toggle the device toolbar icon (Ctrl + Shift + M).
   * Mozilla Firefox: Open the Developer’s tools (press f12 and select ‘Inspect Element’ and toggle the responsive design mode icon (Ctrl + Shift + M).
2. Screenfly: Access Link: https://screenfly.org/

**Conclusion**

The team shall be responsible, under the guidance of their mentor, for proposing solution to their problem statement in simple and effective manner, with UI design for step-wise and screen-wise solution. After validation and concurrence from the mentor, the team shall submit the design to MANAS Codeathon 2023 team. Required consent forms has to be submitted along with the submission (format will be available in website [www.manasmitra.in](http://www.manasmitra.in) soon)