Hackathon Plan-

1. Team Members

|  |  |
| --- | --- |
| Thursday (5mins) – Part 1 |  |
| **Name** | Annamalai.S |
| **Employee ID** | H224346 |
| **Honeywell Mail ID** | [annamalais.s@honeywell.com](mailto:annamalais.s@honeywell.com) |
| **Gmail ID** | [s.annamalaikarthick@gmail.com](mailto:s.annamalaikarthick@gmail.com) |
| **Phone Number** | 9626137884 |
| **Known Programming Languages** | HTML, CSS, Javascript, JQuery, Basics of C, C++, C#, PHP |
| **Known IDE** | Visual Studio, DevC++, Notepad++, Sublime text editor |
|  |  |
| **Name** | Gokulvani.S |
| **Employee ID** | H224243 |
| **Honeywell Mail ID** | [gokulvani.s@honeywell.com](file:///C:\Users\gokulavani_2\Downloads\gokulvani.s@honeywell.com) |
| **Gmail ID** | [gokulavani81@gmail.com](file:///C:\Users\gokulavani_2\Downloads\gokulavani81@gmail.com) |
| **Phone Number** | 9791584371 |
| **Known Programming Languages** | Java,HTML,C,C++ |
| **Known IDE** | Netbeans,Eclipse, |
|  |  |
| **Name** |  |
| **Employee ID** |  |
| **Honeywell Mail ID** |  |
| **Gmail ID** |  |
| **Phone Number** |  |
| **Known Programming Languages** |  |
| **Known IDE** |  |
|  |  |

1. Problem Statement Analysis

|  |  |
| --- | --- |
| Thursday (2hrs) – Part 2 |  |
| **Problem Statement** | ? |
| **Programming Language** | ? |
| **Suitable IDE** | ? |
| **Front-End Languages** | ? |
| **Back-End Languages** | ? |
| **Database** | ? |
| **Install all the required Software’s in each system** | ? |

1. Understanding the Problem Statement – Home Work

|  |  |
| --- | --- |
| Thursday (3hrs: Home) – Part 3 |  |
| **Stay Connected** | **Home** : Gmail Hangouts (Group)  **Office** : Skype Group |
| **Learn and try GIT** | Since we use three different system, it would be useful for version control, and we can update our codes. |
| **Clone this GIT Repo** | **https://github.com/annamalais/myteam.git** |
| **Understand the problem statement well and come out with a Process or Work Flow of our project individually.** | For example, our problem statement is “Create an application like facebook” – we have to identify each of the **modules** or **functionalities** like,   1. Login 2. User-Database 3. News Feeds 4. Chat-Box 5. Groups 6. Contacts, etc   And with this, we can start and then later we can dig deep into each category and add new features.  On the next day we all can discuss and combine each of our idea/flow into a single one. |
| **Search in Google about the problem statement and understand** | If you couldn’t think of any idea or flow, search in google and prepare the flow. |
| **Check out available libraries and code snippets that matches our project** | Keep a track of these, as it would help us while we are coding. |

1. Workflow Discussion

|  |  |
| --- | --- |
| Friday (2hrs) – Part 1 |  |
| **One Workflow** | Let us discuss our ideas and come to a conclusion. |
| **Decide your part of work in the project** | Choose a part that you’re confident with and good in. |
| **Annamalai** | ? |
| **Gokulvani** | ? |
| **Teammate3** | ? |
| **Check whether you have all the necessary software’s installed based on your part of work** | If not, install those software’s asap. |
| **Break** | Once we complete the above things |

1. Check List

|  |  |
| --- | --- |
| Friday (30min) – Part 2 – TODO |  |
| **Prepare your own TODO List for your part of work** | Be specific and first do only the basic thing, then in future we can upgrade/enhance/add features, check each task once you complete it. Prioritize your task correctly. |

|  |  |
| --- | --- |
| Annamalai | Time taken for each task |
| **Task 1** | **30 minutes** |
| **Task 2** | **1 hour** |
| **Task 3** | **20 minutes** |
| **Task 4** | **45 minutes** |
| **Task 5** | **2 hours** |
| **Task 6** | **5 minutes** |
| **Task 7** | **1:30 hours** |
| **Task 8** | **20 minutes** |
| **Task 9** | **20 minutes** |
| **Task 10** | **1 hour** |

|  |  |
| --- | --- |
| Gokulvani | Time taken for each task |
| **Task 1** | **30 minutes** |
| **Task 2** | **1 hour** |
| **Task 3** | **20 minutes** |
| **Task 4** | **45 minutes** |
| **Task 5** | **2 hours** |
| **Task 6** | **5 minutes** |
| **Task 7** | **1:30 hours** |
| **Task 8** | **20 minutes** |
| **Task 9** | **20 minutes** |
| **Task 10** | **1 hour** |

|  |  |
| --- | --- |
| Teammate3 | Time taken for each task |
| **Task 1** | **30 minutes** |
| **Task 2** | **1 hour** |
| **Task 3** | **20 minutes** |
| **Task 4** | **45 minutes** |
| **Task 5** | **2 hours** |
| **Task 6** | **5 minutes** |
| **Task 7** | **1:30 hours** |
| **Task 8** | **20 minutes** |
| **Task 9** | **20 minutes** |
| **Task 10** | **1 hour** |

1. Coding Phase

|  |  |
| --- | --- |
| Friday (1hrs) – Part 3 |  |
| **Happy Coding** | Complete each task asap |
| **Lunch** | Try to complete at least some tasks before lunch |

|  |  |
| --- | --- |
| Friday (4hrs) – Part 4 |  |
| **Task completed – Commit in GIT** | Save your work once it’s done |
| **Have Doubt in your work?** | 1. If, Doubt matches with your team mates skills, then feel free to ask and disturb 2. Else if, Google it 3. Else, Change your idea/code to easy one. |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Create a NEW TASK List** | Before you leave, update your task list with tomorrow’s task. Save your work. |
| **Leave Office** | Try to complete that day task and If you want to work for more time, stay and complete the current tasks and leave office. |

|  |  |
| --- | --- |
| Saturday (4hrs) – Part 1 |  |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Test your Code while you Code (TESTING)** | While you code, check whether it satisfies all the inputs and shows similar output in any case, |
| **Task completed – Commit in GIT** | Save your work once it’s done |
| **Break** | Try to complete at least some tasks before break |

|  |  |
| --- | --- |
| Saturday (4hrs) – Part 2 |  |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Lunch** | Try to complete at least some tasks before lunch |
| **Happy Coding** | Be fast and try to complete more task |
| **Happy Coding** | Be fast and try to complete more task |
| **Test your Code while you Code (TESTING)** | While you code, check whether it satisfies all the inputs and shows similar output in any case, |
| **Task completed – Commit in GIT** | Save your work once it’s done |
| **Merge everyone’s work** | Start merging everyone’s work |
| **Leave Office** | Try to complete that day task and If you want to work for more time, stay and complete the current tasks and leave office. |

1. Deployment Phase

|  |  |
| --- | --- |
| Sunday (3hrs) – Part 1 |  |
| **Merge everyone’s work** | Start merging everyone’s work |
| **Merge everyone’s work** | Start merging everyone’s work |
| **Merge everyone’s work** | Start merging everyone’s work |
| **Test the application** | Check whether it’s working as before merging, |
| **Task completed – Commit in GIT** | Save your work once it’s done |
| **Break** | Complete Merging and test the application |

|  |  |
| --- | --- |
| Sunday (2hrs) – Part 2 |  |
| **Develop Presentation Slides** | Let us create individual slides based on each of our work simultaneously and then merge it to a single Power Point Presentation. |
| **Lunch** | Once we complete our ppt. |

1. Knowledge Transfer

|  |  |
| --- | --- |
| Sunday (2hrs) – Part 3 |  |
| **Make others to understand your Work** | Discussion between ourselves about our work, so that one can help other during the review in case of any difficult/unknown questions are raised by the reviewers. |
| **Break and Leave Office soon** | Take complete rest on Sunday evening and get ready for tomorrow’s presentation. |

All the Best !

Tutorial

**Git Tutorial-Work Flow**:

https://www.youtube.com/watch?v=3a2x1iJFJWc

**Demo Git**:

https://www.youtube.com/watch?v=9pa\_PV2LUlw

https://www.youtube.com/watch?v=HVsySz-h9r4

**Git Commands**:

1. Create a directory

mkdir myrepo

2. Navigate to that directory

cd myrepo

3. Clone the remote repository

git clone <https://github.com/annamalais/myteam.git>

4. Navigate to myteam directory

cd myteam

5. List the files using 'ls'

6. Create your user name and email

git config --global user.name "Anna"

git config --global user.email [s.annamalaikarthick@gmail.com](mailto:s.annamalaikarthick@gmail.com)

7. Open your file or create your file in this folder.

8. Make changes in these files.

-----------**Push and Pull**--------

1. Check status and difference

git diff

git status

2. Add to local repos

git add -A

git status

3. Commit

git commit -m "type your message"

4. Before pushing this to remote repo, pull from remote repo first

git pull origin master

5. Push your changes to remote repo

git push origin master

------- **Create three branches** -------

1. To view existing branch

git branch

2. Create a new branch and switch to it

git branch branchname

git checkout branchname

1. Follow above ---**Push and Pull Procedure**---
   1. Replace master with your branch\_name and follow that.

4. Push your branch to remote

git push -u origin branchname

5. At last when everything is done, we have to merge.

- First checkout to master,

git checkout master

-then pull master

git pull origin master

-merge branchname

git merge branchname

-check whether its merger or not

git branch –merged

-then push your master branch to remote repo.

git push –u origin master

6. If that branch is not required then delete it,.

**Locally delete**: git branch -d branchname

**Remote delete**: git push origin --delete branchname

git branch -a : to check remote branches

Software Links and References

IDE

1. **Sublime Text Editor**: <https://www.sublimetext.com>
2. **Android Studio**: <https://developer.android.com/studio/index.html>
3. **Visual Studio**: <https://www.visualstudio.com/vs/community/>
4. **Dev**-**C++**: <https://sourceforge.net/projects/orwelldevcpp/>
5. **Net Beans**: <https://netbeans.org/downloads/>
6. **Eclipse**: <http://www.eclipse.org/downloads/eclipse-packages/>

Dependencies and Libraries

1. **JDK & JRE**: <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
2. **Wamp Server**: <http://www.wampserver.com/en/>
3. **Oracle Db**: <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>
4. **SQLite**: <https://www.sqlite.org/index.html>
5. **Web Platform Installer (Microsoft)**: <https://www.microsoft.com/web/downloads/> - IIS, .Net Framework, SQL server etc.

References

1. **SQL**: <https://www.sqlteaching.com> , <https://www.tutorialspoint.com/sql/> , <https://www.w3schools.com/sql/sql_intro.asp>
2. **Web Development reference**: <https://www.w3schools.com/>