(Note:

**All text in RED in this template is for instruction purpose only.**

**Delete/Replace this info in final report**

* Use font TIMES NEW ROMAN for the entire report.
* Use 18 font size for Chapter, 16 font size for Section and 14 font size for Sub-section,
* text in Section and Sub-section body should be in black colour and use font size 12
* Use “Center Justification” for chapter headings.
* Use “Left Justification for Section Heading / sub-section headings”
* Use “Justification” alignment for the text in the section and sub-sections.
* Each chapter should begin on new page.
* The page number should be in footer at middle.
* The page number of the Acknowledgement, Abstract, List of Tables, List of Contents, must be in Roman numerals (e.g. i, ii, iii, …….).
* All other pages must be assigned number (1, 2, 3, 4, …….) starting from 1. However, a page number should not appear on the title page.
* In the Table of Contents, four levels of heading can be used, inclusive of the chapter titles. There can be two subheadings within a level of heading.
* **Do not use AI tools like ChatGPT, Gemini, etc. for content, as AI tools often collate text from various sources and lead to high level of plagiarism.**
* The references must be arranged in the order of their appearance. Each entry of the reference must be referred to at least once in the body of the thesis by using a format as mentioned in Harvard referencing and citation style.
* **Refer Reference section for detailed instruction on references**
* Every figure must have the figure number, e.g. Figure 1.1, Figure 2.5, etc. and the figure caption is centered at bottom of the figure.
* Every table must have the table number, e.g. Table 1.1, Table 2.10, and so on and the table caption is centered at the top of the table.
* Note that, in Figure 2.5 or Table 1.3, the number 2 and 1 before decimal represents chapter number in which the figure or table belongs, while the number 5 and 3 after decimal indicates their order within the chapter.
* Every Figure and Table must be in-text cited in the body of sections or sub-section
* The appendixes should be used to contain lengthy or detailed supplementary documentation that may not be appropriate if inserted in the body of the thesis. Examples are data sheets of components, diagrams, models, sample documents, data dictionary, systems structured charts, program listing, user’s manuals, system’s manuals, and glossary of terms.
* Open source tools like **Draw.io, Dia**, etc. can be used to draw the block diagram, flowchart, domain model, etc.

RUBRICS for Project work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Excellent | Good | Satisfactory | Unsatisfactory |
| Abstract & Introduction | Abstract is a concise, powerful summary. Introduction defines the problem and objectives clearly, providing strong motivation and context. | Abstract and introduction are clear and functional. The problem and objectives are well-stated. | Abstract or introduction may be vague. Problem and objectives are present but lack specificity. | Abstract and introduction are missing or illogical. The report’s purpose is unclear. |
| Literature Review | Comprehensive and critical analysis of existing literature. Gaps are identified and justified. The project is effectively positioned within the field. | Covers relevant sources and provides a good overview. Connection to the project is clear. | Includes some relevant sources but may be superficial. The connection to the project is weak. | Absent, irrelevant, or a simple list of summaries without critical engagement. |
| Methodology | Described with exceptional clarity, allowing for full reproducibility. Strong, well-reasoned justification for the chosen approach, tools, and materials. | Described clearly and accurately; the project could likely be replicated. A logical justification for the methods is provided. | Explained, but some details are missing or unclear, making it difficult to fully replicate the work. | Missing or poorly described, providing insufficient detail to understand the project's execution. |
| Analysis and Design | Design choices are innovative, thoroughly justified, and directly address requirements. Robust analysis of alternatives and well-documented decisions. | The design is logical and effective, with a clear rationale for the choices made. Analysis of alternatives is present and sound. | Design is functional, but reasoning is not well-articulated. Limited or no discussion of alternative approaches. | Design is flawed or non-existent. No justification for the design or a poor understanding of design principles. |
| Implementation | Meticulous and well-structured account of hardware and software implementation. Code is well-commented, and schematics are clear. The implementation is robust and functional. | The implementation is clearly documented and functional. Adequate detail on the software and hardware components is provided. | Implementation section is present, but details are sparse or confusing. The hardware or software is not fully described. | Implementation is not described, or the documentation is inaccurate and cannot be used to replicate the project. |
| Project Management | The project plan is detailed and realistic, demonstrating exceptional foresight in managing scope, time, and resources. Includes a thoughtful reflection on challenges and lessons learned. | The project management approach is logical and well-documented. The report shows that the project was executed efficiently. | Provides a simple timeline without a deeper reflection on constraints or challenges faced. | Lacks any discussion of project management, or the project was clearly unmanaged. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evaluation and Results | Results are presented with impeccable clarity. The evaluation is rigorous and unbiased, with insightful analysis that directly addresses objectives. | Results are presented clearly and accurately. The evaluation is sound and addresses most objectives. | Results are presented but may be difficult to interpret due to poor organization. The analysis is basic and lacks depth. | Results are presented without evaluation or are illogical. The data is incomplete, inaccurate, or uninterpretable. |
| Social, Legal, Ethical, & Sustainability | Includes a comprehensive and insightful discussion of the project's broader implications. Potential risks are identified and mitigated. Demonstrates a strong sense of responsibility. | Includes a well-considered section on the broader impact of the project. Key considerations are discussed. | Includes a superficial or generic mention of social and ethical issues, without a deep analysis. | Completely ignores the social, legal, ethical, and sustainability dimensions of the project. |
| Conclusion | Provides a concise and powerful summary of key findings, answering all research questions. It discusses the project's significance and limitations, and provides specific recommendations for future research. | Accurately summarizes the main findings and addresses the research questions. Discusses the project’s significance and offers a few suggestions for future work. | A simple restatement of the results. May fail to answer research questions or discuss implications. Recommendations are vague. | Missing, inaccurate, or unrelated to the rest of the report. No recommendations for future work are provided. |
| References | All sources are meticulously cited using a consistent and correct academic style. The bibliography is comprehensive and includes all references mentioned. | References are mostly correct and follow a consistent style, with only minor errors. The bibliography is complete. | Several errors in the reference list or citations. The style is not consistently followed. | References are missing, incomplete, or incorrectly formatted. |
| Organization & Writing | Flawlessly organized with a logical flow. All sections are clearly labeled, transitions are smooth, and the writing is clear, concise, and free of grammatical errors. | Well-organized with a logical structure. Headings and subheadings are used effectively. Writing is generally clear and professional, with only minor errors. | Organization is acceptable but could be improved. Some sections may feel out of place. Writing contains several errors that may distract the reader. | Lacks a logical structure, making it confusing. Writing is difficult to understand due to numerous errors in grammar and sentence structure. |



**Computerized Cognitive Retraining Program for Home Training of Children with Disabilities**

**A PROJECT REPORT**

## 

***Submitted by***

Manoj Kumar K R- 20221CAI0085

Gnaneshwar D Y- 20221CAI0070

Vikranth Reddy Y- 20221CAI0063

### *Under the guidance of,*

**Ass Prof. Jhon Bennat**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING, AIML**

**PRESIDENCY UNIVERSITY**

**BENGALURU**

**DECEMBER 2025**



**PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

Certified that this report “**Computerized Cognitive Retraining Program for Home Training**

**of Children with Disabilities**” is a bonafide work of “Manoj Kumar KR (20221CAI0085), Gnaneshwar DY (20221CAI0070), Vikranth Reddy Y(20221CAI0063)”, who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING, AIML during 2025-26.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ass Prof. Jhon Bennat**  Project Guide  Presidency School of Computer Science and Engineering  Presidency University | **Dr. Zafar Ali Khan**  Head of the Department Presidency School of Computer Science and Engineering  Presidency University | **Dr. Shakkeera L** Associate Dean  Presidency School of Computer Science and Engineering  Presidency University | **Dr. Duraipandian N**  Dean  PSCS & PSIS  Presidency University |

### Name and Signature of the Examiners

### 1)

2)

**PRESIDENCY UNIVERSITY**

**PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

DECLARATION

We the students of final year B.Tech in COMPUTER SCIENCE ENGINEERING, AIML at Presidency University, Bengaluru, named < Manoj Kumar KR, Gnaneshwar DY, Vikranth Reddy Y >, hereby declare that the project work titled **“<Computerized Cognitive Retraining**

**Program for Home Training of Children with Disabilities >”** has been independently carried out by us and submitted in partial fulfillment for the award of the degree of B.Tech in COMPUTER SCIENCE ENGINEERING, AIML during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

<< Manoj Kumar KR > USN: 20221CAI0085 < Manoj Kumar KR>

< Gnaneshwar DY > USN: 20221CAI0070 < Gnaneshwar DY >

< Vikranth Reddy Y > USN: 20221CAI0063 < Vikranth Reddy Y >

PLACE: BENGALURU

DATE: XX-December 2025

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide Ass Prof**. Jhon Bennat** , Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Zafar Ali Khan, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators**, : Ms. Suma NG, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

<Manoj Kumar KR >

<Gnaneshwar DY>

<Vikranth Reddy Y>

1. Abstract

* Cognitive development plays a vital role in a child’s ability to learn, interact, and adapt to different situations. However, children with disabilities often face difficulties in acquiring these skills due to neurological, developmental, or learning impairments. Challenges such as reduced attention span, weak memory retention, and limited problem-solving abilities make it harder for them to cope with conventional educational approaches. Traditionally, cognitive retraining programs for such children have been administered in hospitals, therapy centers, or specialized institutions. While these programs are effective, they are often restricted by geographical distance, high costs, limited availability of therapists, and the inability of families to make frequent visits. As a result, many children are unable to receive continuous and personalized support, leading to uneven progress in their cognitive development.
* This project introduces a **Computerized Cognitive Retraining Program for Home Training of Children with Disabilities**, which addresses the limitations of traditional therapy methods. The primary aim is to create a home-based, technology-driven system that combines **interactive digital exercises, gamified learning modules, and adaptive algorithms** to strengthen core cognitive skills. The program emphasizes memory enhancement, attention control, problem-solving, and language development through activities that are both engaging and developmentally appropriate. Unlike static exercises, the computerized system dynamically adjusts the level of difficulty based on the child’s progress, ensuring that the training is neither too easy nor too challenging. In addition, the interface is designed to be visually appealing and child-friendly, with intuitive navigation that allows children to interact independently or with minimal assistance.
* Another significant component of the system is the **caregiver and therapist integration module**. Parents or caregivers can track a child’s daily progress through dashboards and receive recommendations for further practice. At the same time, educators and therapists can remotely access performance data, monitor improvements, and suggest modifications to training schedules. This dual-level support ensures that the child receives professional guidance while being trained in the comfort of their home. The platform also incorporates motivational elements such as rewards, progress badges, and audio-visual encouragement to sustain interest and prevent dropout, which is a common issue in long-term training programs
* The implementation of this program demonstrates several advantages. Firstly, it ensures **continuous accessibility** by eliminating barriers of distance and availability. Secondly, it provides a **cost-effective alternative** to frequent therapy center visits, making cognitive training affordable for families across different socio-economic backgrounds. Thirdly, it offers **personalized and adaptive training**, which is critical in addressing the unique needs of each child. Finally, the integration of data-driven insights allows therapists to base their interventions on measurable outcomes rather than generalized assumptions.
* In conclusion, the **Computerized Cognitive Retraining Program** represents a **scalable and sustainable solution** for supporting children with disabilities in their cognitive development journey. By combining technology with therapeutic principles, it not only promotes inclusivity and accessibility but also aligns with broader global objectives of improving quality education and reducing inequalities.

1. INTRODUCTION
2. Background:

* Children with disabilities often face challenges in memory, attention, and problem-solving, which affect learning and daily life. Cognitive retraining programs aim to improve these skills but are usually clinic-based, costly, and inaccessible to many families. This creates the need for **affordable, home-based solutions** using technology.

1. Statistics of Project:

* As per WHO, around **15% of the world’s population lives with some form of disability**, and a large proportion are children. In India, nearly **7.8 million children** live with disabilities, but only a small fraction have access to specialized training. This highlights the importance of computerized home-based retraining programs.

1. Prior Existing Technologies:

* Existing tools like **Cogmed, Lumosity, and BrainHQ** provide cognitive exercises but are expensive, internet-dependent, and not designed specifically for children with disabilities. In India, such programs are limited, creating a gap for customized, affordable, and adaptive solutions.

1. Proposed Approach:

* Gamified exercises for engagement.
* Adaptive difficulty levels.
* Caregiver dashboards to track progress.
* Remote therapist access for feedback

1. Objectives:

* Develop an interactive home-based program for cognitive retraining
* Implement adaptive learning to suit each child’s pace.
* Provide monitoring tools for caregivers and therapists.
* Make the system cost-effective and widely accessible.

1. Sustainable Development Goals (SDGs):

* SDG 3: Good Health and Well-Being
* SDG 4: Quality Education
* SDG 10: Reduced Inequalities
* SDG 9: Innovation and Infrastructure

1. Overview of Project Report:

* Chapter 1 introduces the project. Chapter 2 reviews related literature. Chapter 3 presents the methodology. Chapter 4 covers project management. Chapter 5 details analysis and design. Chapter 6 explains hardware and software implementation. Chapter 7 discusses evaluation and results. Chapter 8 explores social, legal, and ethical aspects. Chapter 9 concludes with findings and future work.



1. Literature Review

* Computerized cognitive training (CCT) has been widely investigated as an intervention to improve working memory, attention, and related executive functions in children. Early work showed that targeted working-memory training can produce measurable gains in working memory and some executive measures in children with ADHD and other conditions, although transfer to broader academic outcomes has been inconsistent.
* Several randomized controlled trials and meta-analyses indicate that CCT produces reliable improvements on trained tasks (e.g., working memory tasks) and modest improvements on untrained neuropsychological measures, particularly in attention-deficit populations; however, evidence for durable school-level or generalized academic gains is mixed and depends on training dose, program design, and outcome measures.
* Research specific to children with developmental or intellectual disabilities (IDD) reports promising results from multi-domain computerized programs: short multi-week CCT interventions have shown improvements in intellectual performance and adaptive functioning in children with IDD in recent trials. These studies suggest that appropriately designed multi-domain training can benefit children beyond neurotypical samples.
* Feasibility and effectiveness of **home-based** and remote delivery have been explored in multiple settings. Telerehabilitation and remote CCT have been found to be feasible and at least as effective as in-person interventions for certain outcomes; remote delivery improves access and adherence, which is crucial in low-resource or geographically dispersed populations.
* Several disorder-specific studies show feasibility in special groups: computerized programs were successfully trialed with children with autism spectrum disorder, fragile X syndrome, Down syndrome, and acquired brain injury — demonstrating feasibility, acceptable adherence, and improvements on domain-specific measures in many cases. Nevertheless, effect sizes and generalization vary across conditions and study designs.
* Limitations across the literature include heterogeneity in program content, outcome measures, small sample sizes, variable follow-up durations, and inconsistent reporting of real-world functional outcomes (e.g., school performance, daily living skills). Several systematic reviews emphasize the need for larger, well-controlled trials with standardized outcomes and attention to transfer and maintenance of gains. This identifies a gap for culturally adapted, low-cost, adaptive home programs evaluated on both cognitive and functional endpoints.
* This literature motivates your proposed project: a low-cost, gamified, adaptive, home-based retraining system with caregiver and therapist

1. Methodology
2. Research Design:

* The study adopts an **experimental design** to evaluate the effectiveness of a computerized cognitive retraining program. The focus is on developing a home-based software application tailored for children with disabilities, enabling personalized training sessions in a supportive environment. Both qualitative and quantitative approaches are integrated to assess improvements in cognitive functions such as attention, memory, problem-solving, and processing speed.

1. Participants:

* The participants include children aged between 6 and 14 years with documented cognitive or developmental disabilities. A purposive sampling technique is used to select participants in collaboration with rehabilitation centers and special educators. Prior consent from parents/guardians and ethical approval from concerned authorities are obtained.

1. Materials and Tools:

* **Software Platform:** The retraining program is implemented using Python/JavaScript frameworks with a user-friendly interface.
* **Hardware Requirements:** Standard home-use personal computers or tablets with internet connectivity.
* **Cognitive Training Modules:** Interactive exercises targeting attention span, memory enhancement, visual-spatial skills, and problem-solving abilities.

1. Procedure:

* **Requirement Analysis:** Conducted with therapists, educators, and parents to identify essential training needs.
* Participants engage in daily sessions (20–30 minutes) over a period of 8–12 weeks at home.
* Progress is tracked automatically by the software and remotely supervised by therapists through cloud-based dashboards.
* Pre-test and post-test assessments are compared to measure cognitive improvements.

1. Data Collection and Analysis:

* Performance scores, response times, and error rates recorded automatically by the program.
* Feedback from parents and educators regarding usability, motivation, and behavioral changes.

1. Ethical Considerations:

* Informed consent and confidentiality are strictly maintained.
* Parental involvement is ensured for guidance during training.
* The program is designed to be non-invasive, safe, and child-friendly.

Table of Content

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title** | **Page No.** |
|  | Declaration |  |
|  | Acknowledgement |  |
|  | Abstract |  |
|  | List of Figures |  |
|  | List of Tables |  |
|  | Abbreviations |  |
| 1. | Introduction  1.1 Background  1.2 Statistics of project  1.3 Prior existing technologies  1.4 Proposed approach  1.5 Objectives  1.6 SDGs  1.7 Overview of project report |  |
| 2. | Literature review |  |
| 3. | Methodology |  |
| 4. | Project management  4.1 Project timeline  4.2 Risk analysis  4.3 Project budget |  |
| 5. | Analysis and Design  5.1 Requirements  5.2 Block Diagram  5.3 System Flow Chart  5.4 Choosing devices  5.5 Designing units  5.6 Standards  5.7 Mapping with IoTWF reference model layers  5.8 Domain model specification  5.9 Communication model  5.10 IoT deployment level  5.11 Functional view  5.12 Mapping IoT deployment level with functional view  5.13 Operational view  5.14 Other Design |  |
| 6. | Hardware, Software and Simulation  6.1 Hardware  6.2 Software development tools  6.3 Software code  6.4 Simulation |  |
| 7. | Evaluation and Results  7.1 Test points  7.2 Test plan  7.3 Test result  7.4 Insights |  |
| 8. | Social, Legal, Ethical, Sustainability and Safety Aspects  8.1 Social aspects  8.2 Legal aspects  8.3 Ethical aspects  8.4 Sustainability aspects  8.5 Safety aspects |  |
| 9. | Conclusion |  |
|  | References |  |
|  | Base Paper |  |
|  | Appendix |  |

List of Figures

(include list of all figure and page no

e.g.

Figure Caption Page no

Fig 1.1 Sustainable development goals 12

Fig 3.1 The V model methodology 14

Fig 3.2 Another example of the V model methodology 14

…..

List of Tables

(include list of all tables and page no

e.g.

Table Caption Page no

Table 2.1 Summary of Literature reviews 15

Table 4.1 Project planning timeline 20

Table 4.2 Project implementation timeline 20

…….

Abbreviations

(include all abbreviations used in report in alphabetic order)

e.g.

IoT Internet of Things

SDG Sustainable Development Goal

….

Chapter 1

Introduction

(The Introduction should describe

* The background of the project,
* Aim of project,
* The need of the project and the statistics, and
* The of prior existing technologies)

1.1 Background

* Describe the background of the project and
* **ensure proper citation and referencing** the source.

1.2 Statistics

* Describe the statistics (preferably the regional) and the need of the project and
* **ensure proper citation and referencing** the source.

1.3 Prior existing technologies

* Describe the prior existing technologies of the project and
* **ensure proper citation and referencing** the source.

1.4 Proposed approach

* Aim of project
* Motivation
* Proposed approach
* Applications of the project
* Limitation of the proposed approach

1.5 Objectives

(The objectives should be 3 to 5.

* **Do not list benefits or applications** as objectives.
* Objectives should encompass various aspects listed below (but not limited to) of the project and

1. Behaviour -
2. Analysis -
3. System management –
4. Security –
5. Deployment -

* should be demonstrable in the project.)

e.g.

E.g. Interfacing light and temperature sensor to microcontroller to monitor ambient light and temperature in range of 100C to 600C)

1.6 SDGs

* Align the **project objectives** with UN SDGs as illustrated in Fig 1.1.
* assess how your project aligns with some UN SDGs and
* **ensure proper citation and referencing**



Fig 1.1 Sustainable development goals [1]

1.7 Overview of project report

(gives an Overview of the report should be in a single paragraph summarising all chapters.

e.g.

Chapter 1 provides an introduction on the project topic which is…Chapter 2 discusses the literature reviews …, Chapter 3 describes the methodology …. Chapter 4 covers the required project budgeting…, Chapter 8, entitled ‘ABC’, discusses the …)

Chapter 2

Literature review

* (Summarise in own word the context at least ten literature taken from conference or journal papers.
* **Note: Do not summarise from websites, wikipedia, blogs, etc**.
* Summarising includes understanding the context (like the concept/approach used in literature, the results obtained, the future work and recommendations) and **write review in own words without use of AI tools**. May also include illustrations from literature wherever essential / necessary.

• Discuss the concepts, approach, methods, analysis, and issues adapted in the project.

• Identify the limitations, inconsistencies, gaps, contradictions, differences.

• Suggest improvements.

• (**do not include detailed description, images, tables, data, illustration, results, graphs, etc. from the literature unless all that information is used as basis for your project**)

• (**do not use AI tools like ChatGPT, Gemini, etc. to summarise as AI tools often collate text from various sources and lead to high level of plagiarism.)**

• The key points should also be presented in a table.

* each as a single paragraph of about half page
* **Ensure proper citation and referencing**)

e.g.

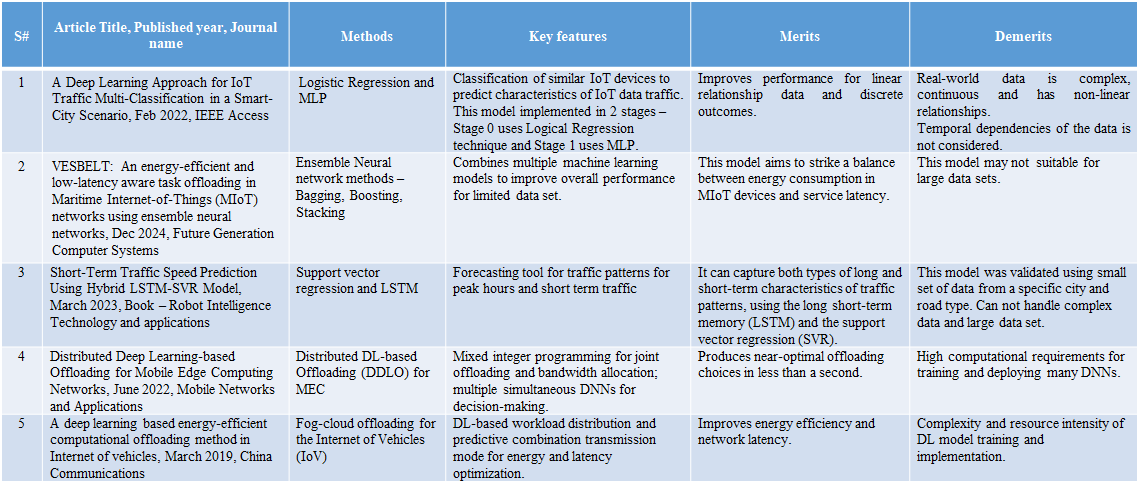
NarasimhaRao et al. implemented an IOT weather forecasting system that measures various parameters such as temperature, humidity, and rainfall and air pollution comparing with the traditional weather forecasting system. The Iot weather forecasting gives continuous and real time data. The information from the sensors is sent to the cloud server. To make sure that there is no external harm for the data, the data security in cloud provides with encryption and decryption of data while retrieving the information from cloud. This reduces corruption of data, minimizes communication overhead, enhances the security and also helps to give wise predictions. The work uses the traditional database system management is replaced by cloud, cloud has comparatively less maintenance cost, data storage cost and it increases the scalability. It provide enhanced security a homomorphic encryption technique is used, this provides a safer communication between the cloud and user. The homomorphic technique has many advantages compared to the present security systems [2].

Raju and Laxmi examine the use of supervised machine learning algorithm and short-term load forecasting for iot based online load forecasting. The data obtained from a research lab is used for training ML algorithms to short term load forecasting. Online forecasting is considered more effective as it uses recent data for forecasting and training. Different regression algorithms are implemented on cloud for forecasting power consumption. The effectiveness of ML algorithms is known by calculating the performance parameters such as RMSE, MSE and so on. The methodology presented in this paper offers real time prediction of the power consumption due to this its very much preferred in IOT based online home energy management system [3].

Summary of Literatures reviewed

e.g. summarise the literatures

Table 2.1 Summary of Literature reviews



Chapter 3

Methodology

(Identify a suitable methodology for the project, like V-model, Spiral, Devops, Onion, SDLC, Waterfall, Agile, Scrum, etc. illustrated below

* Describe the stages of the project by mapping to the stages of methodology viz. requirements (specification, literature), system design, functional design, unit design (hardware, software, cloud), unit testing, integration testing, Verification, Validation, etc.)
* **Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**
* **Ensure proper citation and referencing.**

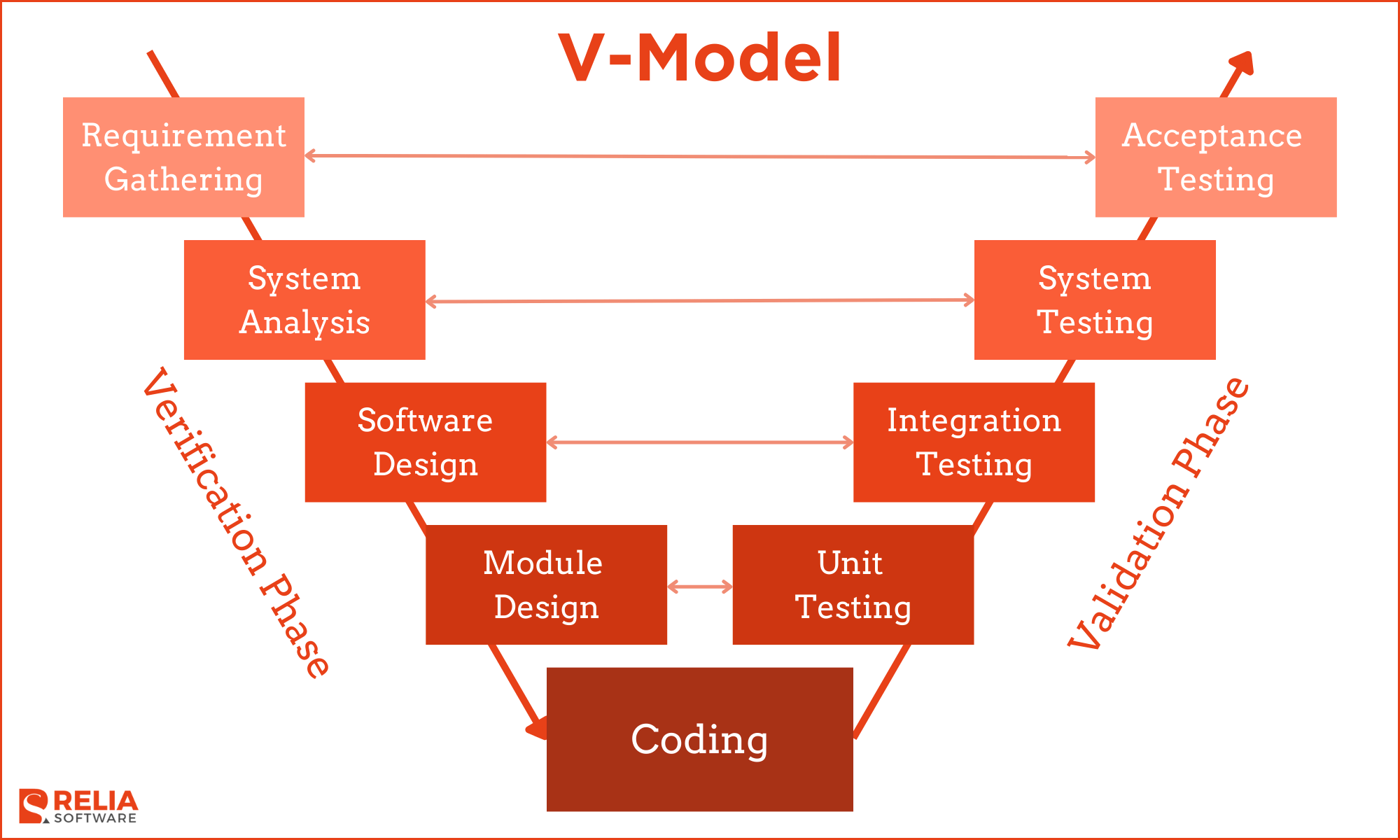
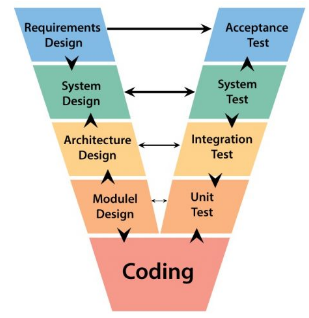


Fig 3.1 The V model methodology [4]

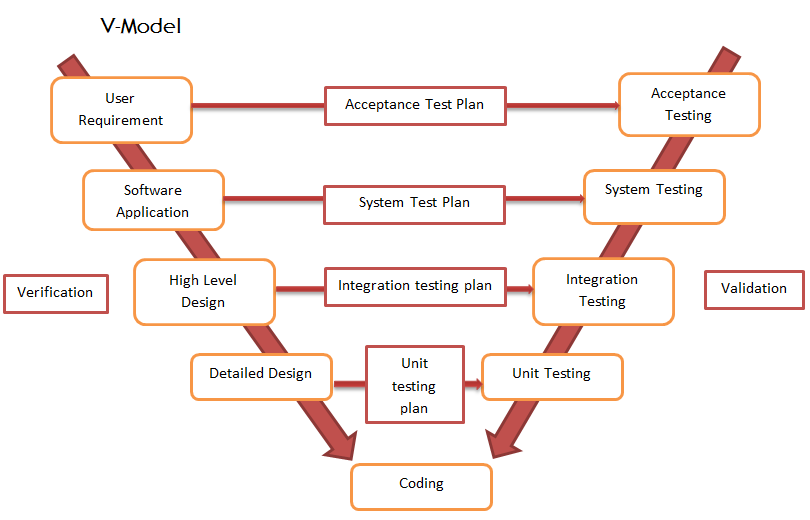


Fig 3.2 Another example of the V model methodology [5]

Fig 3.1 and Fig 3.2 shows the V-model methodology. The verification stage includes phases …., whereas the validation phase includes ….

Fig 3.3 illustrates the W- model methodology. Each phase of the W-model has a corresponding test requirements and testing phase…..

**Note:**

* **Every Fig must be captioned below the figure.**
* **Every figure must be described along with in-text citation in body of sections as shown above paragraphs.**
* **Also it should be added to the list of figures section**
* **Ensure proper citation and referencing for figures taken from sources.**

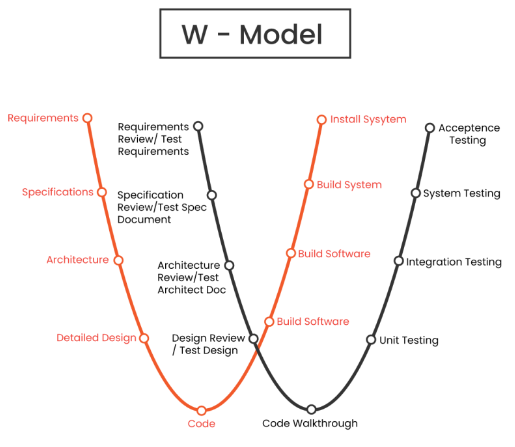


Fig 3.3 The W model methodology [6]

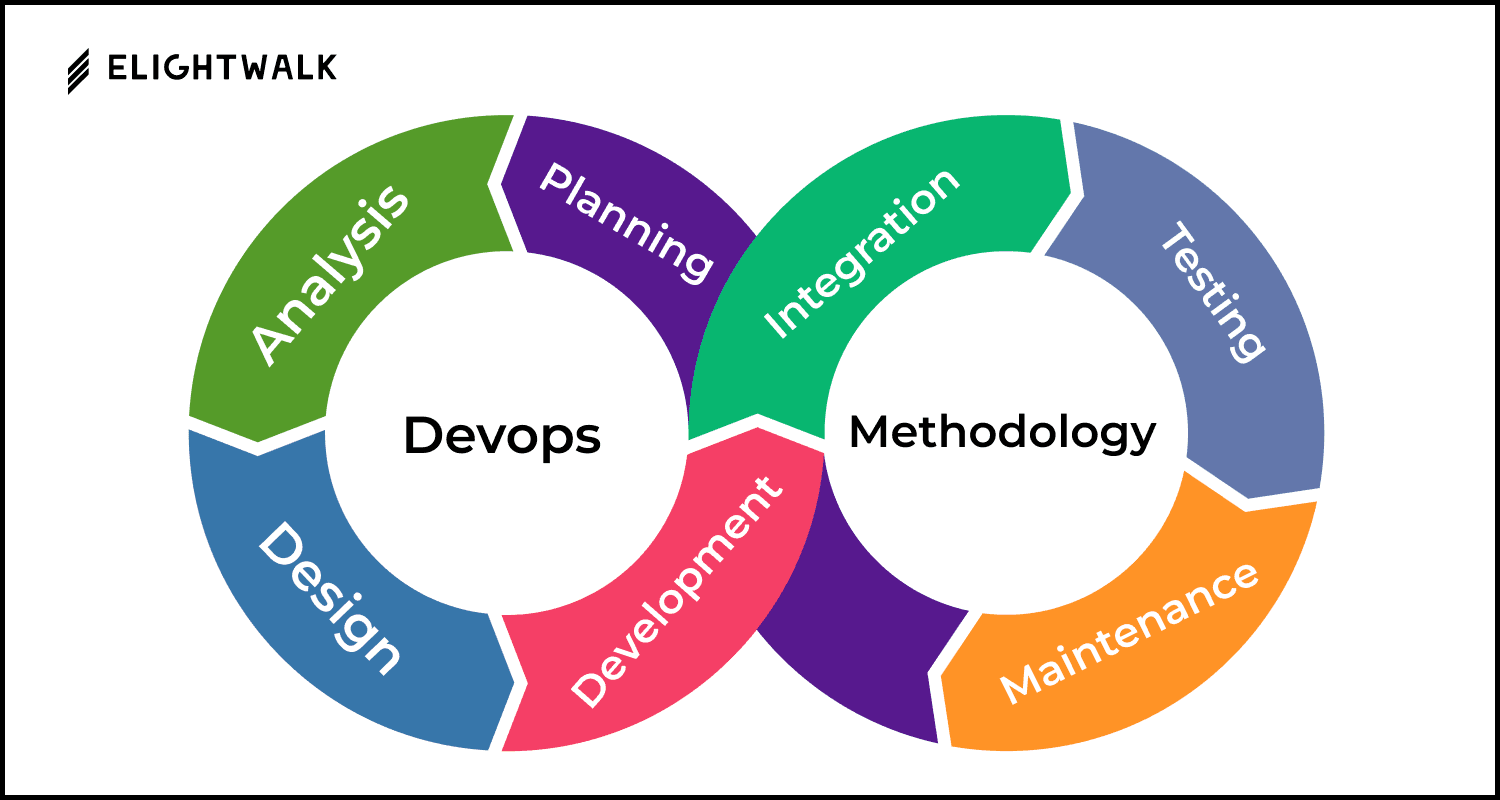
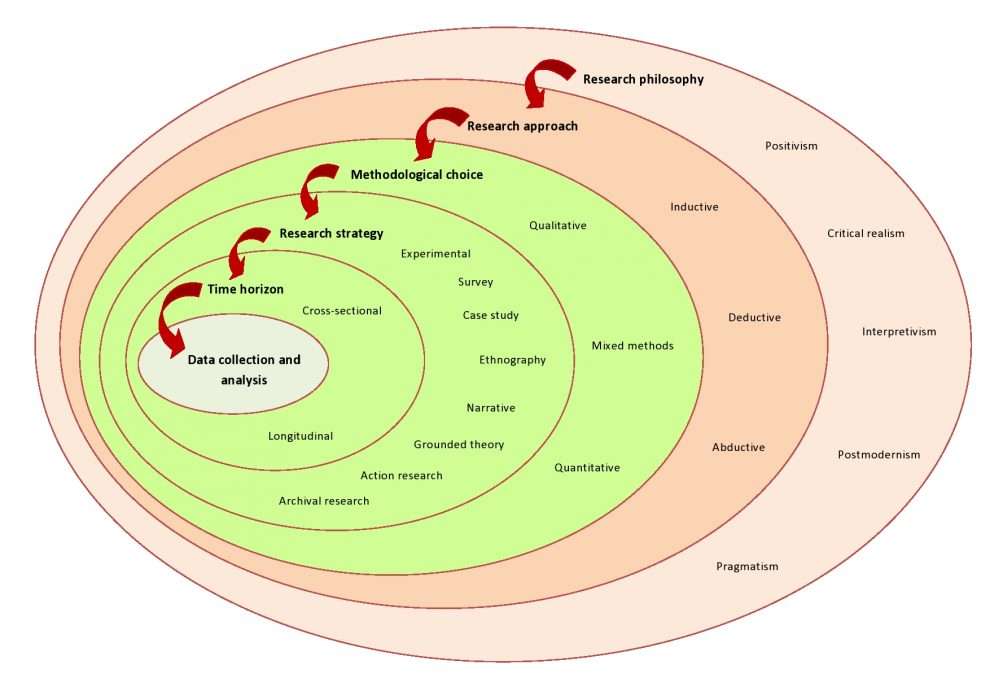


Fig 3.4 The Devops methodology [7]

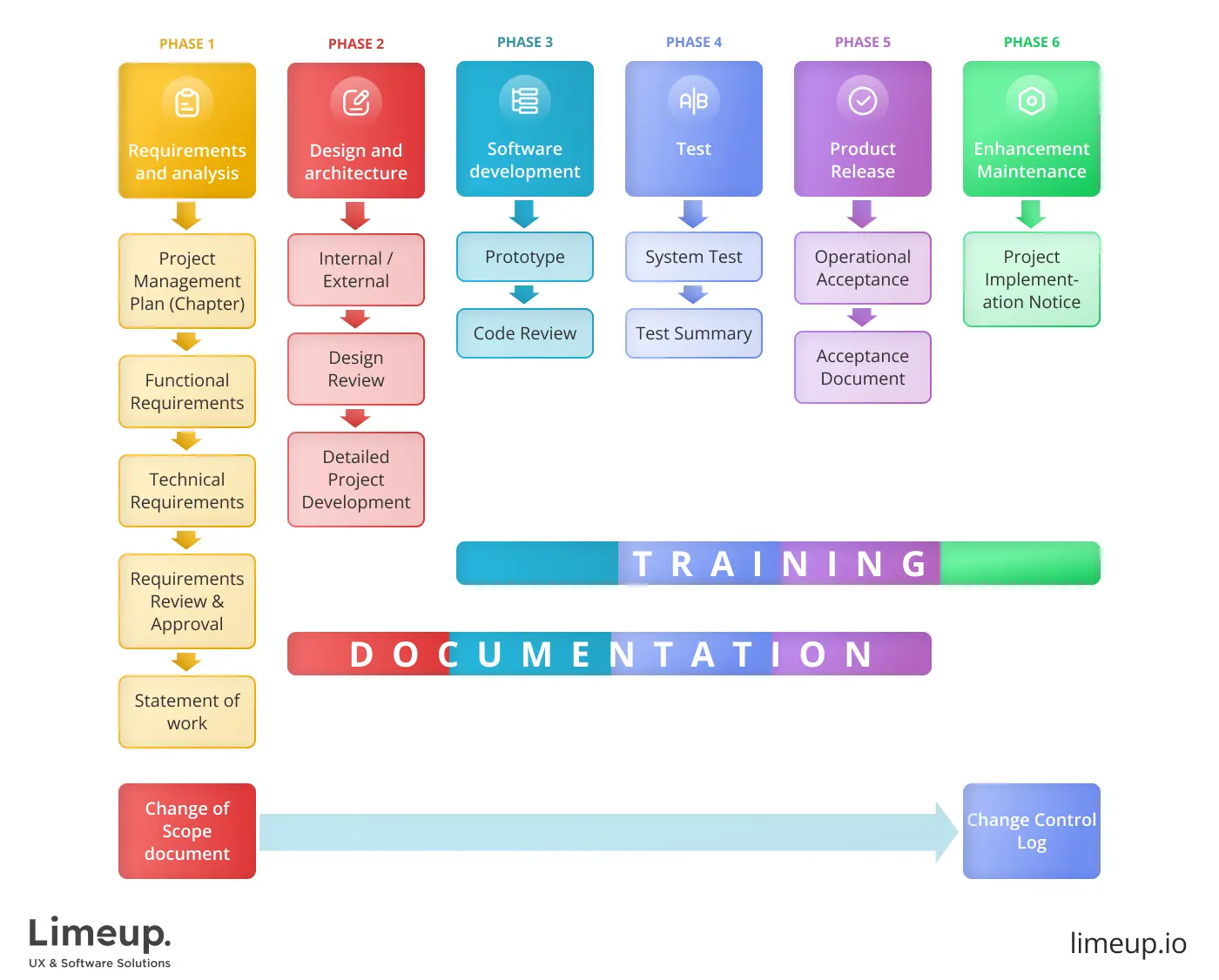


(a)



(b)

Fig 3.5 (a) Onion methodology, (b) another example [8]



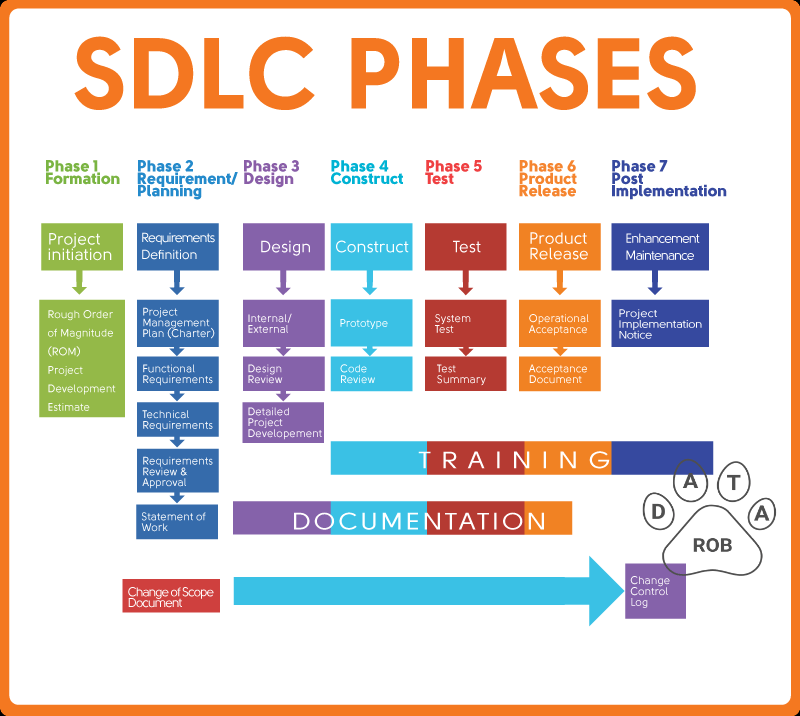


Fig 3.6 SDLC phases [9]

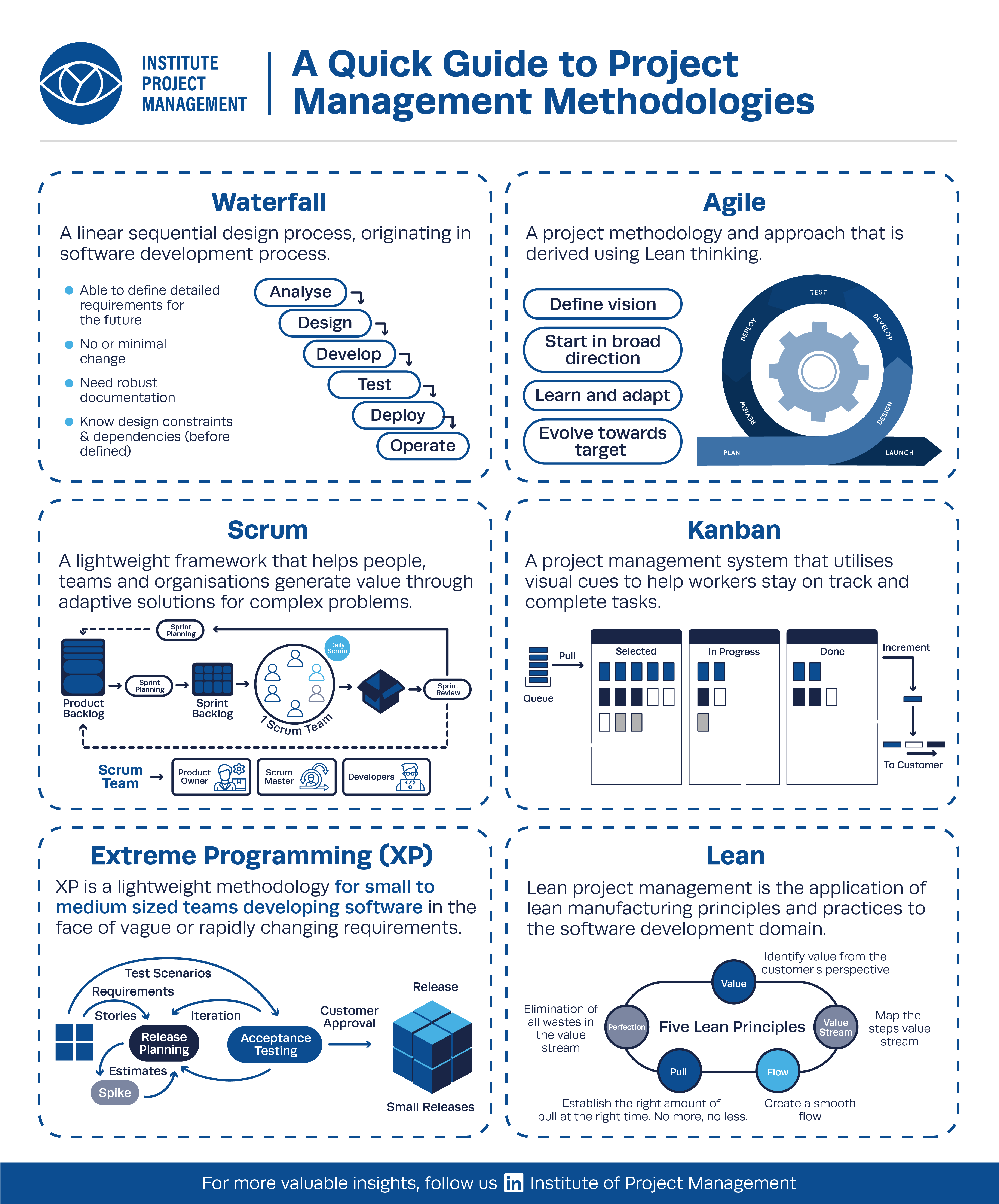


Fig 3.7 Summary of various methodology [10]



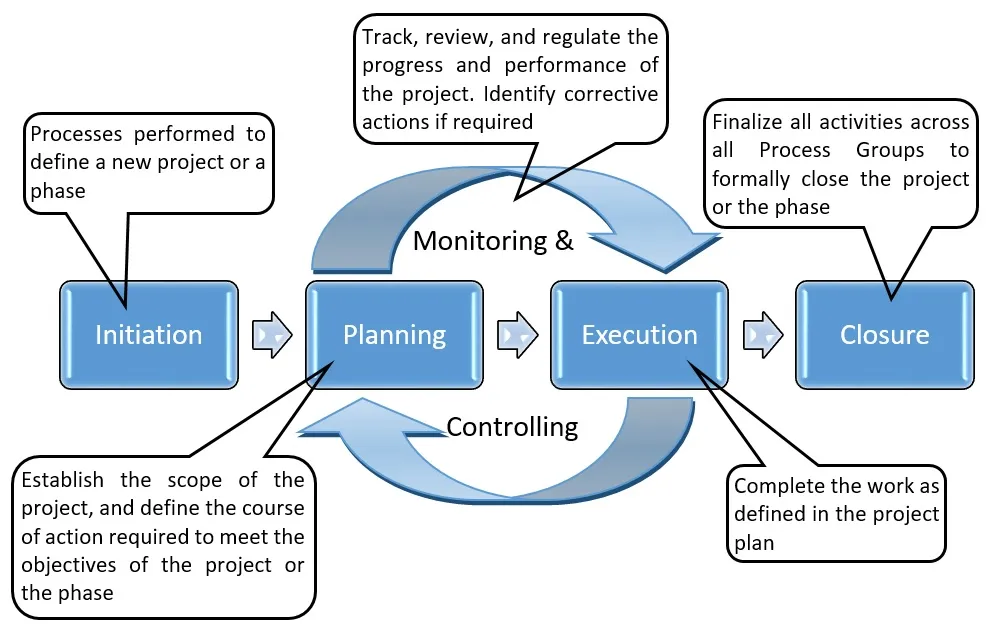
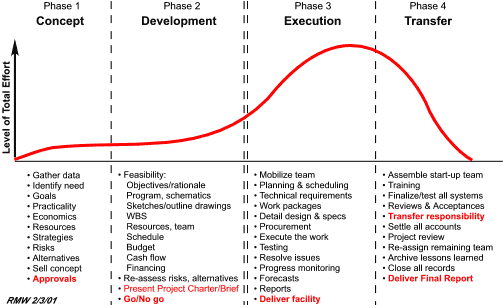


Fig 3.8 Some more methodologies [11]



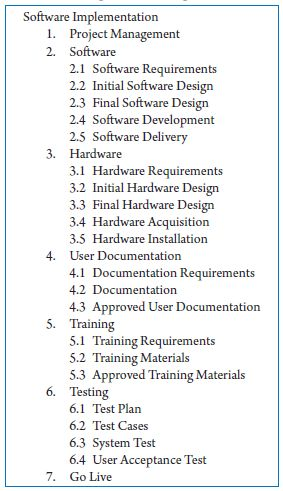


Fig 3.9 Summary of project breakdown to task [12]

Chapter 4

Project Management

4.1 Project timeline

(Show a tabular visual representation of project breakdown, project's schedule, outlining tasks, milestones, and deadlines in chronological order)

Also called **Gantt chart** is a visual, bar-chart-based project management tool that displays tasks, their duration, start and end dates, dependencies, and milestones along a timeline. It provides a clear overview of a project's schedule and progress, allowing teams to track work, identify potential bottlenecks, and ensure alignment on project goals.

**What a Gantt Chart Shows**

* **Tasks**: A vertical list of all the activities required to complete a project.
* **Timeline**: A horizontal axis that represents time, showing the duration of the project.
* **Bars** (Gantt Bars): Horizontal bars on the timeline, with their length indicating a task's duration and position showing its start and end dates.
* **Dependencies**: Lines or arrows connecting tasks to show which tasks must be completed before others can begin.
* **Milestones**: Important points or due dates in the project, often marked by a diamond or star symbol.
* **Progress**: The bars can be shaded or filled to show the percentage of a task that has been completed.
* **Assignees**: The person or team responsible for each task can be shown on the chart.

**How It's Used**

* **Planning**: Helps in breaking down a project into manageable tasks and scheduling them in a logical sequence.
* **Scheduling**: Creates a visual roadmap of the project timeline, allowing for better time management.
* **Tracking Progress**: Provides a quick, at-a-glance view of project status and task completion.
* **Resource Management**: Allows teams to assign tasks and see workloads.
* **Communication**: Offers a clear, shared view of the project for all team members and stakeholders, ensuring everyone is aligned.

**To prepare Gantt chart Use open source tools GanttProject or Google Sheets. It can also be prepared using MS Excel, or MS Project that may require license**.

Project Planning

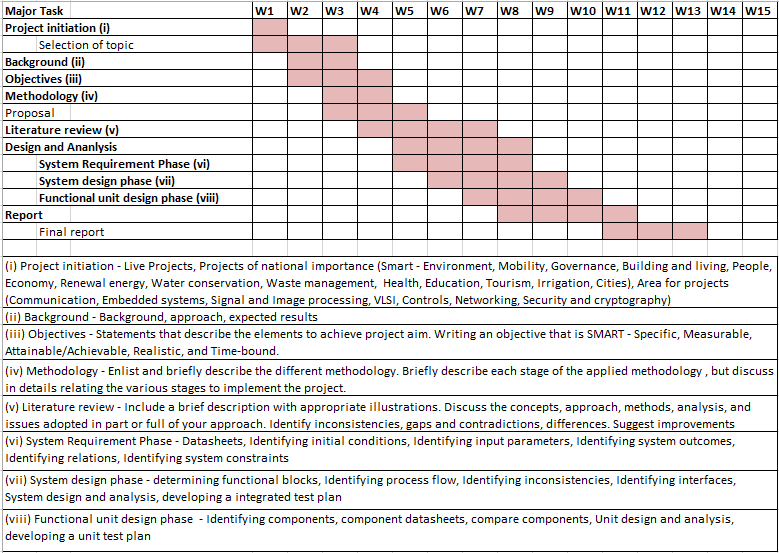
Table 4.1 summarises the timeline during the project planning phase. The milestones are identified and the deadlines are ……

Table 4.2 summarises the timeline during the project implementation phase……

**Note:**

* **Every Table must be captioned above the table**
* **Every table must be described along with in-text citation in body of sections as shown above paragraph.**
* **Also it should be added to the list of tables section**
* **Ensure proper citation and referencing for Tables taken from sources.)**

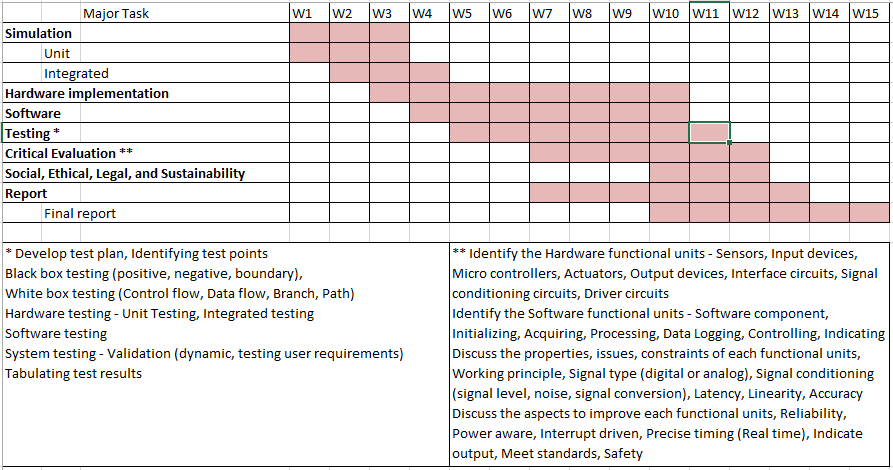
Table 4.1 Project planning timeline



(Describe the project timeline and describe the suitability for the project)

Project implementation

Table 4.2 Project implementation timeline



(Describe the project timeline and describe the suitability for the project)

4.2 Risk analysis

(PESTLE analysis - assess how these factors might impact a project's success and allows for proactive risk mitigation and opportunity maximization)

Table 4.3 Example of PESTEL analysis [13]



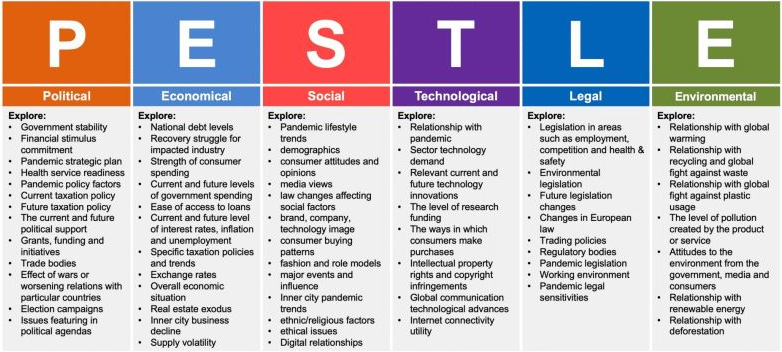
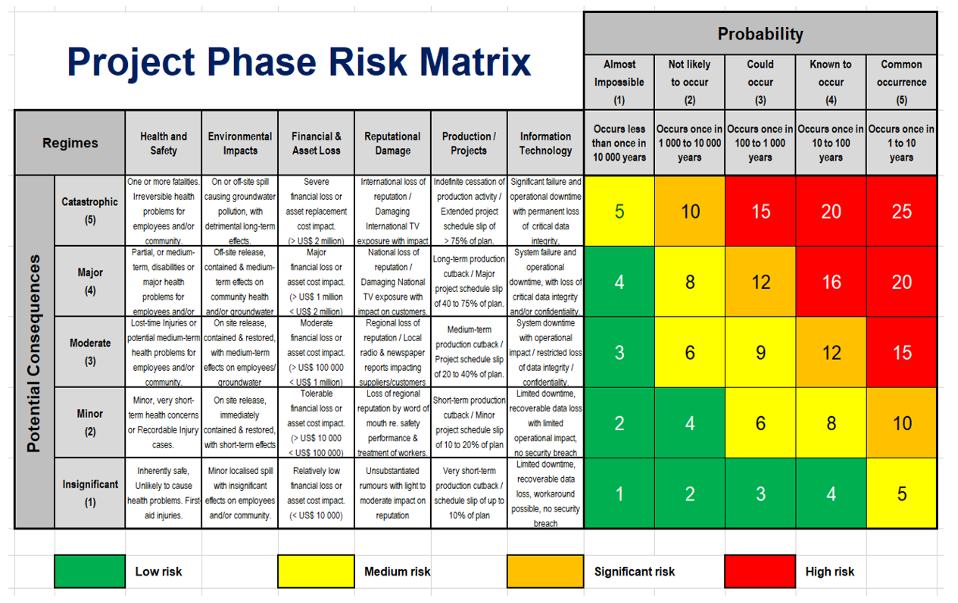
Table 4.4 Another example of PESTEL analysis [14] 

Table 4.5 Example of Project phase risk matrix [15]



(Describe the risk these factors might impact a project's success and the appropriate risk mitigation for the project)

4.3 Project budget

Step 1: List All Tasks and Resource Requirements

Step 2: Check Team Availability

Step 3: Estimate Task Duration

Step 4: Use Your Experience and Data

Step 5: Set the Project Budget

Step 6: Keep Track of the Project Budget and Assess the Team

Sample Template for project budget is shown in Table 4.6

Table 4.6 Example of project budget [16]



Chapter 5

Analysis and Design

(Analysis and design are distinct but interconnected phases in a systems development. Analysis focuses on understanding the problem and gathering requirements, while design focuses on creating a solution based on those requirements. Essentially, analysis identifies "**what**" a system needs to do, and design determines "**how**" it will be done.

Design is the process of creating a plan, concept, or sketch for an object, system, or process with a specific purpose or function. It involves conceiving, structuring, computing values to derive the appropriate components, and arranging elements to meet a particular need or goal, often incorporating both functional and aesthetic considerations. The term can also refer to the resulting blueprint, pattern, or the inherent arrangement of an object's components.

Analysis is the detailed, organized examination of a complex topic, substance, or situation by breaking it down into its smaller, constituent parts to understand its nature, essential features, or underlying causes. It involves carefully studying the pieces to gain deeper insight into the whole, often to make a decision, draw conclusions, or solve a problem.

* **Ensure proper citation and referencing**)

5.1 Requirements

the requirements of the system.

* In this step, the system purpose, behaviour and requirements are captured. Requirements can be:
* System HW Requirement Phase
  1. Identify Initial conditions
  2. Determine Input parameters
  3. System outcomes
  4. Formulate relations
  5. Identify system constraints
* System SW Requirement Phase
  1. Identify Initial conditions
  2. Determine Input parameters
  3. System outcomes
  4. Formulate relations
  5. Identify system constraints
* Data collection requirements
* Data analysis requirements
* System management requirements
* Security requirements
* User interface requirements

 e.g. For home automation system the purpose and requirements specification is in Table 5.1 as follows:

Table 5.1 Summarizing requirements

|  |  |
| --- | --- |
| **Purpose** | A home automation system that allows controlling the lights remotely using a web application |
| **Behaviour** | Home automation system should support two modes: auto and manual  **Auto:**System measures the light level in the room and switches on the light when it is dark  **Manual:**Allows remotely switching lights on and off |
| **System Management** | System should provide remote monitoring and control functions |
| **Data Analysis** | System should perform local analysis of the data |
| **Application Deployment** | Application should be deployed locally, but should be accessible remotely |
| **Security** | Should provide basic security like user authentication |

* System HW design phase
  1. Determine functional blocks
  2. Develop process flow
  3. Identify inconsistencies
  4. Design interfaces
  5. System design and analysis
  6. Developing an integrated test plan
* System SW design phase
  1. Determine functional blocks
  2. Develop process flow
  3. Identify inconsistencies
  4. Design interfaces
  5. System design and analysis
  6. developing an integrated test plan

5.2 Block diagram

(The functional block diagram.

* Show inputs blocks on the left, the processor in middle, the output block to the right. Show relation between blocks with appropriate direction arrows.
* Blocks should be the functions and not names of components.
* Circuit diagram is not block diagram)
* **Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

e.g.

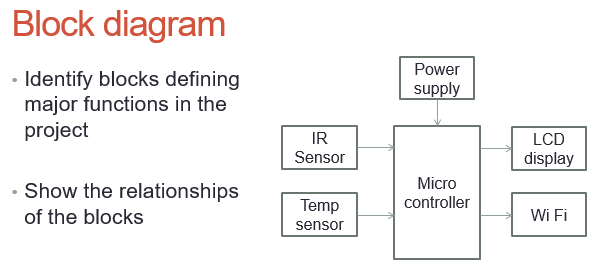


Fig 5.1 Functional block diagram

(Describe the block diagram and describe the suitability for the project)

Fig 5.1 shows the functional block diagram of the system. It consist of ……

5.3 System Flow chart

(The system flow chart

* Visualise the process flow of initialisation, inputs, processing, conditions, and outputs)
* **Use open source tools viz. Draw.io, Dia, or any other to draw flow diagrams.**

e.g.

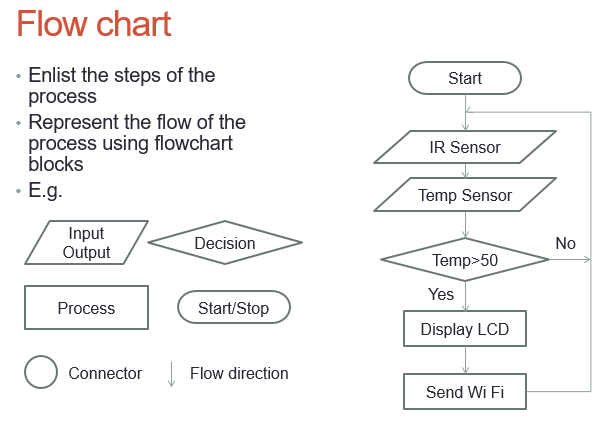


Fig 5.2 System flow chart

(Describe the flowchart and describe the suitability for the project)

Fig 5.2 shows the system flow chart. The flow begins with initialisation……

* Functional HW unit design phase
  1. Identify HW components
  2. HW Component datasheets
  3. Compare components
  4. Unit design and analysis
  5. Develop a unit Test plan
* Functional SW unit design phase
  1. Identify SW components
  2. HW Component datasheets
  3. Unit design and analysis
  4. Develop a unit Test plan

5.4 Choosing devices

(viz. microcontroller, sensor, actuators)

(Instructions

1. Search for multiple models of each type of device in columns
2. Summarize from **DATASHEETS** the feature in rows.
3. Add rows to include additional features/specifications not listed in table.
4. Leave table cells blank wherever data is not available
5. Below each table include the references for every datasheet referred along with the URL.)

Choosing Processor

(While choosing the IoT device/Processor for the IoT application,

* Summarize (as table below) the specifications (like digital ports, analog port, Serial port, I2C, UART, etc) of various IoT devices referring to the **DATASHEETS** and
* Then choose the device that has the all necessary features for the project
* **Ensure proper citation and referencing.**)

(Example look for various processors (Arduino, Raspberry Pi, ESP32, etc.) and compare the specifications)

Table 5.2 Comparing features of different processors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features/ Specification** | **Arduino** | **Raspberry Pi** | **NodeMCU** | **ESP32 dev kit** |  |
| Power Supply | 5V |  |  |  |  |
| Number of Digital I/O ports | 14 |  |  |  |  |
| Voltage and Current rating for digital port | 5V  20mA |  |  |  |  |
| Number of Analog ports | 6 |  |  |  |  |
| Voltage range of analog port | 0 – 5V |  |  |  |  |
| Number of Serial ports | 1 |  |  |  |  |
| I2C | 1 |  |  |  |  |
| SPI | 1 |  |  |  |  |
| UART | 1 |  |  |  |  |
| PWM | 6 |  |  |  |  |
| RAM | 2KB SRAM, 32KB FLASH, 1KB EEPROM |  |  |  |  |
| Timers/  Counters | 2x 8-bit Timer/Counter 1x 16-bit Timer/Counter |  |  |  |  |
| Bluetooth | No |  |  |  |  |
| Wireless | No |  |  |  |  |
| Other features |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**References (include in the reference section at the end)**

Arduino, Arduino UNO3, <https://docs.arduino.cc/hardware/uno-rev3/#tech-specs>

Choosing devices (Sensors, driver circuits, Actuators)

For every device required in the project, while choosing the devices (Sensors, driver circuits, Actuators, etc) that are connected to the IoT devices,

* Summarize (as table below) the specifications (like working principle, output type, voltage range, sensitivity, etc) of various devices referring to the **DATASHEETS** and
* Then choose the device that meet the IoT device.
* **Ensure proper citation and referencing of datasheets.**

(Example look for various temperature sensor (LM35, DHT11, etc) and compare the specifications)

1. Temperature sensor

Table 5.3 Comparing features of different temperature sensors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features/ Specification** | **LM35** | **DHT11** | **Thermistor** | **Thermocouple** |
| Describe Working principle (state the relation between O/P to I/P) | Detects temperature and convert to proportional voltage |  |  |  |
| Sensitivity  (change in O/P for unit change in I/P) | 10mV / 0C |  |  |  |
| Resolution  (smallest value that can be measured) | 10mV |  |  |  |
| Accuracy  (how closely measured value matches the true or expected value of the quantity being measured.) | ±0.5°C at 25°C |  |  |  |
| Linearity  (O/P signal reflects a linear relationship with the I/P) | Non-Linearity Only ±¼°C Typical |  |  |  |
| Range  (the minimum and maximum values that it can accurately measure) | -55 0C to 150 0C |  |  |  |
| Output Type (analog, digital, Serial, etc) | Analog |  |  |  |
| Power Supply | 4 V to 30 V |  |  |  |
| Power consumption | 60 μA |  |  |  |
| Drift  (gradual change in O/P over time, even when the I/P remains constant) |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**References (include in the reference section at the end)**

**Ti, LM35 Specifications,** [**https://www.ti.com/product/LM35**](https://www.ti.com/product/LM35)

Similarly do for **all other devices** required in the project

1. Moisture Sensor
2. Humidity Sensor
3. Pressure Sensor
4. Motion Sensor
5. Distance Sensor
6. Location Sensor
7. Optical Sensor
8. Velocity Sensor
9. Level Sensor
10. Chemical Sensor
11. Relay Actuators
12. Motor drivers
13. Solenoids
14. Servo motors
15. DC motors
16. Stepper motors

5.5 Designing units

**Breakdown** the project into **sub-project or units** and design each unit of the project

Identify the interfacing circuits required for each unit and

compute the values for signal conditioning like level shifting, amplification, A/D conversion, etc.



Fig 5.3 Example of interfacing a one unit

e.g.

1. Design of the temperature unit

Computing values for LM35

Therefore, for temperature 200C

Using 16-bit ADC for converting Analog to digital

Computing Temperature value from digital value

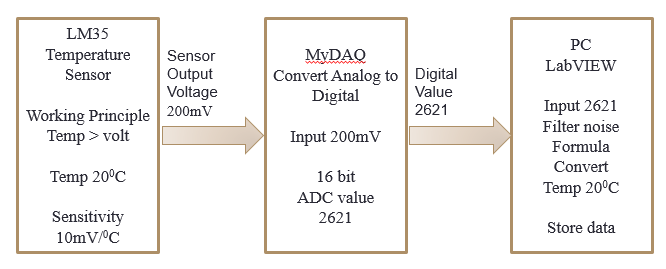


Fig 5.4 Example of computing the values for the temperature unit

**(Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.)**

Similarly do for **all other units** of the project

5.6 Standards

(**Discuss and describe the various standards that are relevant and required for the project**

* **Ensure proper citation and referencing.**

Standards are technical specifications that ensure interoperability, security, and data exchange between various devices and systems. They cover areas like communication protocols, data formats, security, and device management, allowing devices from different manufacturers to work together seamlessly.

Key aspects of IoT standards are Interoperability, Security, Data Formats and Protocols, Device Management.

* Communication Protocols: Standards like Wi-Fi, Bluetooth, ZigBee, MQTT, and CoAP.
* Data Formats: Standards like JSON, XML, and CBOR are used for exchanging data between devices and platforms.
* Security: Standards such as TLS for secure communication and standards related to data privacy and device security.
* Architecture and Interoperability: Standards related to layered architectures or specific device interactions.

Benefits of using IoT standards are

* Reduced Development Costs: Standards streamline development by providing a common framework and reducing the need for custom solutions.
* Improved Interoperability: Standards enable devices from different manufacturers to work together, creating a more flexible and dynamic IoT ecosystem.
* Enhanced Security: Standards provide guidance on security best practices, helping to mitigate risks and protect sensitive data.
* Increased Innovation: By providing a foundation for interoperability and security, standards enable innovation and the development of new IoT applications.

e.g.

IEEE develops standards for various aspects of IoT, including

Wireless communication (802.11, 802.15),

smart grids (1547), and

intelligent transportation (1609)

Key ISO standards for IoT devices include

Frameworks ISO/IEC 30141 (IoT Reference Architecture),

Secure design ISO/IEC 27400 and ISO/IEC 27402 (IoT Security & Privacy), and

Testing ISO/IEC 21823 series (Interoperability).

ITU: The ITU (International Telecommunication Union) focuses on establishing a global infrastructure for the IoT, ensuring interoperability and advanced services.

DDS (Data Distribution Service): A scalable IoT protocol for real-time and embedded systems, offering interoperable data exchange.

GS1: GS1 provides standards for identifying objects and capturing data, crucial for the IoT ecosystem.

Key ISO standards for IT include

ISO/IEC 27001 for information security management,

ISO 9001 for quality management, and

ISO 20000 for IT service management.

Other relevant standards helping organizations manage risks, improve efficiency, and ensure compliance in a complex IT landscape cover

business continuity (ISO 22301),

data privacy (ISO 27701), and

Artificial Intelligence management (ISO/IEC 42001)

communication standards – PCI, UART, I2C, SPI, CAN, etc. MQTT, CoAP, OPC UA: These are popular communication protocols used in IoT for different purposes.

Other standards like Power supply standards, interfacing standards, etc.)

5.7 Mapping with IoTWF reference model layers (in tabular form)

(The IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access.

• It provides a clear way of visualizing IoT from a technical perspective.

• Each of the seven layers is broken down into specific functions, and security encompasses the entire model.

Using this reference model, we are able to achieve the following:

1. Decompose the IoT problem into smaller parts

2. Identify different technologies at each layer and how they relate to one another

3. Define a system in which different parts can be provided by different vendors

4. Have a process of defining interfaces that leads to interoperability

5. Define a tiered security model that enforced at the transition points between levels)

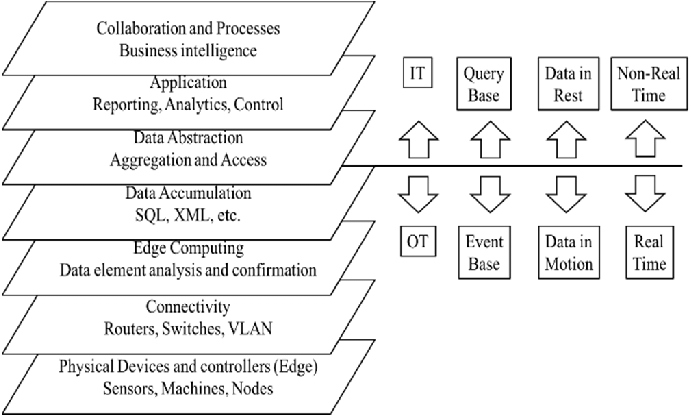


Fig 5.5 The IoT World Forum Reference Model

Table 5.4 Mapping Project layers with IoTWFRM

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **IoT World Forum Reference Model** | **Project Layer mapping** (Identify different technologies at each layer and how they relate.  Have a process of defining interfaces) | **Security**  (Tiered security model that enforced at the transition points between levels) |
| 7 | Collaboration and Processes  (involving people and business processes) |  |  |
| 6 | Application  (reporting, analytics, control) |  |  |
| 5 | Data Abstraction  (aggregation and access) |  |  |
| 4 | Data Accumulation  (Storage) |  |  |
| 3 | Edge Computing  (data element analysis and transformation |  |  |
| 2 | Connectivity  (communication and processing units) |  |  |
| 1 | Physical devices and Controllers  (things) |  |  |

5.8 Domain model specification

(The domain model describes the main concepts, entities and objects in the domain of the IoT system to be designed.

* Domain model defines the attributes of the objects and relationships between objects.
* The domain model is independent of any specific technology or platform
* It consist of physical entities, virtual entities, devices, resources, services as defined in table below

Table 5.5 Description of Domain model

|  |  |
| --- | --- |
| **Physical Entity** | • The physical identifiable objects in the environment  • IoT system provides information about the physical entity (using sensors) or performs actuation upon the physical entity |
| **Virtual Entity** | • Virtual entity is a representation of the physical entity in the digital world  • For every physical entity there is a virtual entity |
| **Device** | • Devices provide a medium for interaction between physical and virtual entities  • Devices are used to gather information from or perform actuation on physical entities |
| **Resource** | • Resources are software components which can be either on-device or network-resources  • On-device resources are hosted on the device and provide sensing or actuation (eg: operating system)  • Network-resources include software components that are available on the network (eg: database) |
| **Service** | • Services provide an interface for interacting with the physical entity  • Services access resources to perform operations on physical entities |

)

e.g.

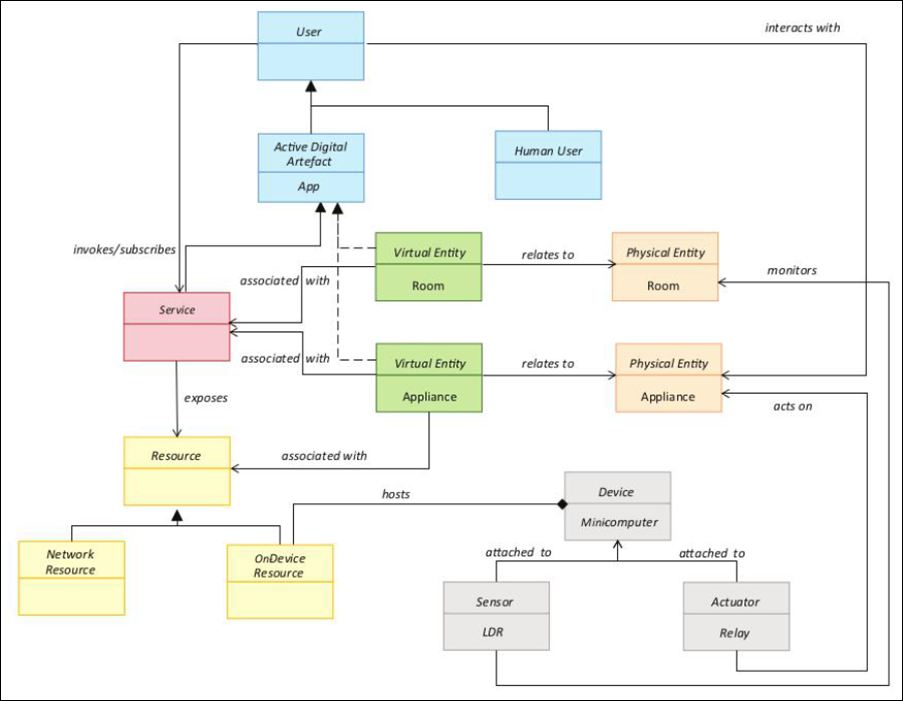


Fig 5.6 Domain model for Home automation

(Describe the domain model specification and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

5.9 Communication model

(Based on the requirements choose the Communication model and describe the suitability for the project

The applicable communication model for the project

1. Request-response model

2. Publish-subscribe model

3. Push-pull model

4. Exclusive pair model)

e.g.

**Request-Response communication model**

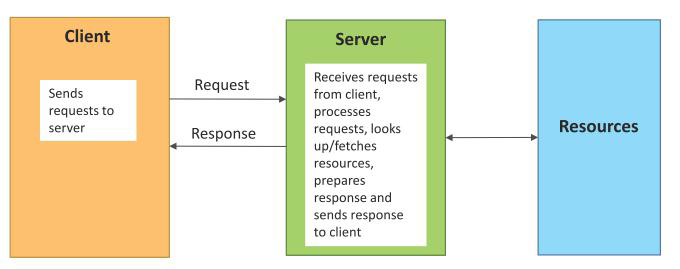


Fig 5.7 Communication model suitable for Home automation

(Describe the Communication model and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

5.10 IoT deployment level

(Based on the requirements choose the IoT application deployment level and describe the suitability for the project

The applicable IoT deployment level for the project

1. IoT deployment Level 1

2. IoT deployment Level 2

3. IoT deployment Level 3

4. IoT deployment Level 4

5. IoT deployment Level 5

6. IoT deployment Level 6)

e.g.

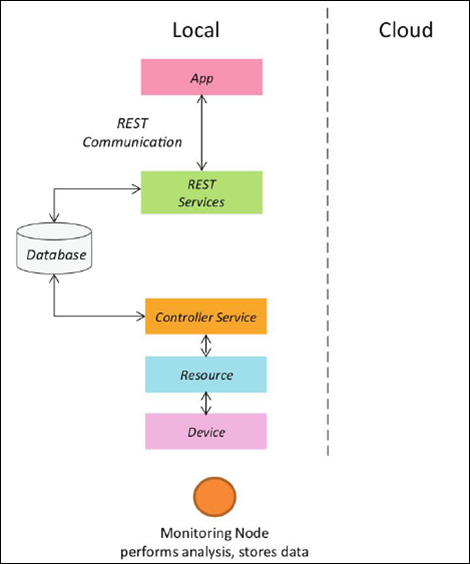


Fig 5.8 IoT deployment level suitable for Home automation

(Describe the IoT deployment level and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

5.11 Functional view

(The functional view defines the functions of the IoT systems grouped into various functional groups. Decompose the project and describe it through the functional view

* Each functional group provides functionalities for interacting with concepts in the domain model and information related to the concepts.
* The functional groups in a functional view include: Device, Communication, Services, Management, Security, and Application.)

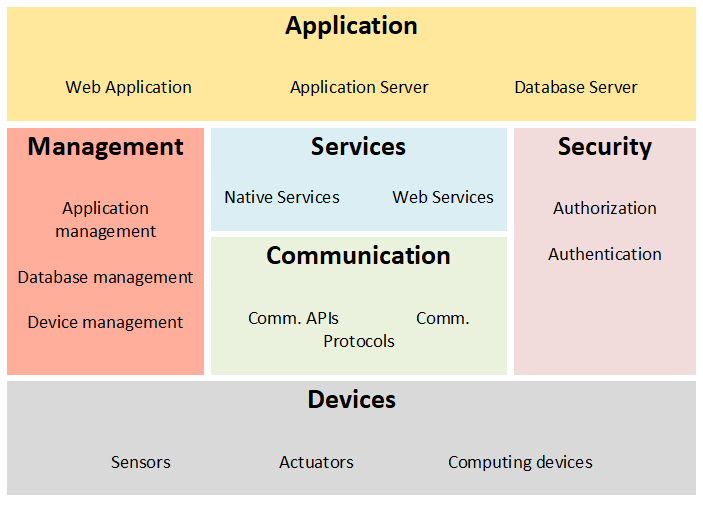


Fig 5.9 Functional view for Home automation

(Describe the functional view and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

5.12 Mapping deployment level with functional blocks

(The mapping between the IoT level and the functional groups as suitable for the project. Describe the mapping)

e.g.

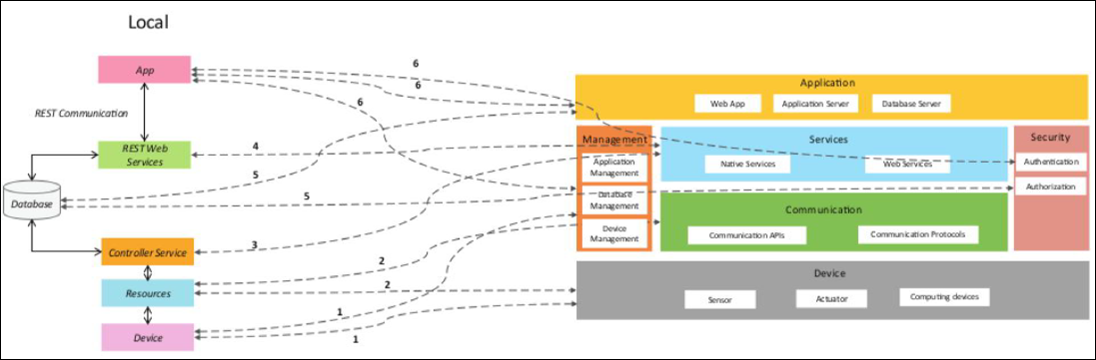


Fig 5.10 mapping IoT deployment level with functional view

(Describe the mapping between the IoT level and the functional view and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

5.13 Operational view

(define communication options, service hosting options, storage options, device options, various options related to the IoT system deployment and operation are defined, such as:

* Service hosting options
* Storage options
* Device options
* Application hosting options)

e.g.

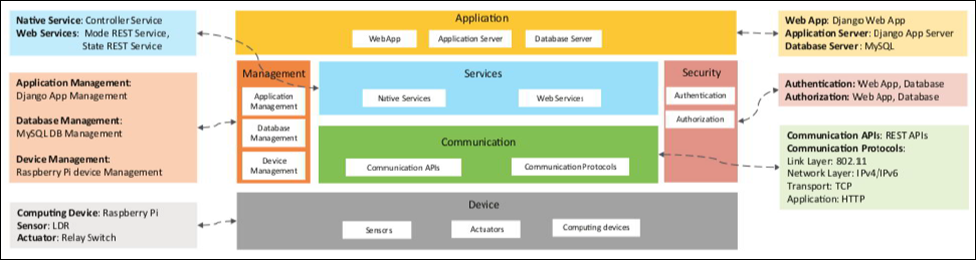


Fig 5.11 Operational view

(Describe the operational view and describe the suitability for the project)

**Use open source tools viz. Draw.io, Dia, or any other to draw block diagrams.**

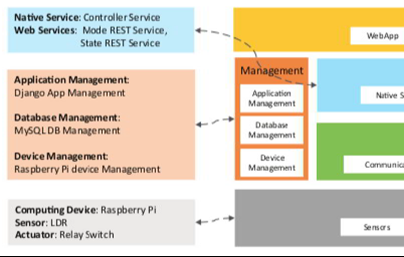




Fig 5.12 another Example of operational view

5.14 Other Design aspects

1. Process specification
2. Information model specification
3. Service specification
4. Etc.

Chapter 6

Hardware, Software and Simulation

* 1. Hardware

(include

* The **circuit diagram and description** of each sub-project or functional unit.
* The **integration and description** of sub-project or functional unit to realize complete project.
* The description of hardware tools required and their **configuration process** for development of hardware.
* **Ensure proper citation and referencing.**

**NOTE: DO NOT insert images and description of hardware components**

e.g.

* **Hardware development tools** for legacy MCUs.
* **Debugger and Programmer tools**. Debugging and programming tools and accessories.
* **Reference Designs**. Application-specific designs showcasing complete system solutions
* **Explorer Kits**. Low-cost exploratory kits - to get you started.
* **Starter Kits**. Modular kits for software development and feature evaluation of IoT devices.
* **Radio Boards**. Wireless starter kit plug-in boards featuring specific SoC or module wireless devices.
* **Expansion boards**. Starter kit and ecosystem accessory boards
* **Evaluation Kits**. Product and feature evaluation kits
* **Development Kits**. Full- featured application development kits
* **Pro Kits**. Modular kits for software development and feature evaluation of IoT devices.
* **Thunderboards**. Low-cost software development and prototyping kit for IoT devices)
  1. Software development tools

(include the software programs and platforms tools that help streamline the software development lifecycle by automating tasks, improving collaboration, and ensuring code quality.

* Software used to create, manage, test, and deploy applications, including IDEs, version control systems, project management software, and deployment tools.
* Describe the **configuration procedure** of each tool as per the project requirement.
* **Ensure proper citation and referencing.**

**NOTE: DO NOT insert images, logo and description of software**

e.g.

Key Categories of Software Development Tools

**Integrated Development Environments (IDEs) / Code Editors:**

These tools provide a comprehensive environment for writing, editing, and debugging code.

Examples: Visual Studio Code, IntelliJ IDEA, Eclipse

**Version Control Systems (VCS):**

Used for tracking changes in code and facilitating collaboration among developers.

Examples: Git, GitHub, GitLab, Bitbucket

**Project Management Tools:**

Help teams organize tasks, manage workflows, and track project progress.

Examples: Jira, ClickUp, Trello

**Continuous Integration/Continuous Deployment (CI/CD) Tools:**

Automate the process of building, testing, and deploying software.

Examples: Jenkins, GitHub Actions

**Containerization Tools:**

Enable developers to package applications and their dependencies into portable containers.

Examples: Docker

**Cloud Platforms:**

Provide cloud services and infrastructure for developing, deploying, and managing applications.

Examples: Microsoft Azure, Amazon Elastic Compute Cloud

**Collaboration Tools:**

Facilitate communication and discussion within development teams.

Examples: Slack

**API Testing Tools:**

Used for designing and testing Application Programming Interfaces (APIs).

Examples: Postman

**Testing Frameworks:**

Tools for automating the testing of software applications to ensure quality and identify errors.

Examples: Selenium)

* 1. Software code

(include the code,

* Describe the function of blocks of code,
* Comment each line of code
* **Ensure proper citation and referencing.**)

e.g.

The software for temperature unit is as follows

The temperature sensor and standard libraries are included and variables are initialised

#include <stdio.h>

#include <dht.h>

float temp;

int tempPin = 0;

float tempCelsius = 25.0;

float tempFahrenheit = 77.0;

Two function to convert Celsius to Fahrenheit and vice versa are

// Function to convert Celsius to Fahrenheit

float celsiusToFahrenheit(float celsius) {

return (celsius \* 9.0 / 5.0) + 32.0;

}

// Function to convert Fahrenheit to Celsius

float fahrenheitToCelsius(float fahrenheit) {

return (fahrenheit - 32.0) \* 5.0 / 9.0;

}

The program initialise the variables and call the conversion functions to process the input values and display the results.

// Initilaise the serial monitor

void setup() {

Serial.begin(9600);

}

// Continuously monitor the temperature sensor and out the value

void loop() {

temp = analogRead(tempPin);

// read analog volt from sensor and save to variable temp

temp = temp \* 0.48828125;

// convert the analog volt to its temperature equivalent

Serial.print("TEMPERATURE = ");

Serial.print(temp); // display temperature value

Serial.print("\*C");

Serial.println();

delay(1000); // update sensor reading each one second

Serial .print("Temperature Conversions:\n");

// Celsius conversions

Serial.println("%.2f Celsius is %.2f Fahrenheit.\n", tempCelsius, celsiusToFahrenheit(tempCelsius));

// Fahrenheit conversions

Serial.println("%.2f Fahrenheit is %.2f Celsius.\n", tempFahrenheit, fahrenheitToCelsius(tempFahrenheit));

}

Similarly describe codes of other units and integration of the units.

* 1. Simulation

Simulation allows for the virtual testing of electronic circuits, microcontrollers, and complex systems, enabling developers to predict behavior, test designs, and optimize performance without physical hardware

Simulators can be used depending on project

* **Circuit Simulators**: Tools like LTSpice and TINA-TI provide detailed analysis of analog and digital circuits, allowing users to test component behavior and circuit performance before building prototypes.
* **Microcontroller (MCU) Simulators**: Software like Proteus VSM, Oshonsoft, and WOKWI simulate the operation of microcontrollers such as Arduino, PIC, and AVR, enabling the testing of microcontroller code.
* **Full-System Simulators**: Intel Simics is a comprehensive platform for simulating complex hardware systems, facilitating development and debugging of software for these systems.
* **Hardware-in-the-Loop** (HIL) Simulators: These systems (often supported by platforms like MATLAB/Simulink) test a controller's real-time response to virtual stimuli, acting as a bridge between software control and physical hardware.
* **3D Modeling and Simulation Software**: Platforms such as Ansys and Fusion 360 combine design capabilities with powerful simulation tools to analyze physics-based problems like structural integrity, fluid dynamics, and thermal behavior.

**Popular Software Examples**

* **LTSpice**: A free circuit simulator from Analog Devices, widely used for analog and power electronics, offering accurate simulations and a vast component library.
* **TinkerCAD Circuits**: A free, web-based platform from Autodesk that is excellent for learning and simulating basic electronics and Arduino projects.
* **Proteus VSM**: A versatile simulator for microcontrollers, supporting various architectures like PIC, AVR, and Arduino.
* **Ansys**: A comprehensive suite for advanced engineering simulation, covering fields like structural analysis, computational fluid dynamics (CFD), and more.
* **Intel® Simics**®: A full-system simulator that allows developers to access and test complex electronic systems without needing the physical hardware.
* **MATLAB/Simulink**: Provides tools for modeling and simulation, with capabilities for Hardware-in-the-Loop (HIL) testing and integrating with physical systems.

**Some Free Circuit Design Software**

1. [CircuitMaker](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#1-circuit-maker)
2. [Open Circuit Design Software](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#2-open-circuit-design-software)
3. [KiCad EDA](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#3-ki-cad-eda)
4. [ADS Circuit Design Software](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#4-ads-circuit-design-software)
5. [nagaEDA Circuit Design Software](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#5-naga-eda-circuit-design-software)
6. [OpenSce Circuit Design Software](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#6-open-sce-circuit-design-software)
7. [QSapecNG Circuit Design Software](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#7-q-sapec-ng-circuit-design-software)
8. [SimulIde](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#8-simul-ide)
9. [gerbv](https://www.electronicsforu.com/special/cool-stuff-misc/open-source-circuit-design-software#9-gerbv)

Chapter 7

Evaluation and Results

7.1 Test points

* Identifying test points
* Highlight the test points in each functional unit and determine various scenarios. (trouble shooting)
* Different test points should be marked for hardware and software to check functionality.
* Determine the list of measurement (values, type of signals, etc.) at each test point.
* Each design test value should be compared with expected value, simulation and hardware for every functional unit.)
* **Use open source tools viz. Draw.io, Dia, or any other to draw circuits diagrams.**

e.g.

The test points are marked as V1, V2, V3, V4 and V5 for checking the required voltage levels at each of the point in functional unit automatic switching.

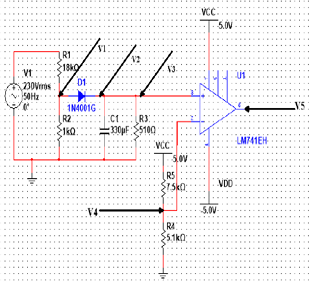


Fig 7.1 Power supply unit indicating the test points

Figure 7.1 Testing points for the automatic switching. …..

Example of test scenarios for a project related to solar panels

o First scenario or test plan should test the output with various considerations. Test the output of the solar panel at different levels and calculate the potentiometer adjustment theoretically.

o Second scenario or test plan 2 in this stage calculates the voltages for the interfaces connected.

Test point 3 or scenario 3…

7.2 Test plan

* Develop test plan Develop the test plan for each functional unit.
* The test plan should be in the form

<Subject> <verb> <Object> <conditions> <values> <range> <constraints>

* Develop the test cases for measuring various system characteristics and performance such as accuracy, linearity, latency, power aware etc.
  + Black box testing (positive, negative, boundary)
  + White box testing (Control flow, Data flow, Branch, Path)
  + Hardware testing
  + Unit Testing
  + Integrated testing
  + Software testing
  + System testing
  + Validation (dynamic, testing user requirements)

e.g.

TP1: The voltage must be 3.25V when the input is varied between 4 to 6V.

TP2: Measure output voltage of temperature sensor for range of temperature

TP3: Measure the output voltage of the IR sensor for detecting object and distance.

TP4: for various input value at ADC pin of microcontroller display the value in serial terminal.

Tp5: for each value of ADC pin of microcontroller compute the Temperature and display the temperature in serial terminal.

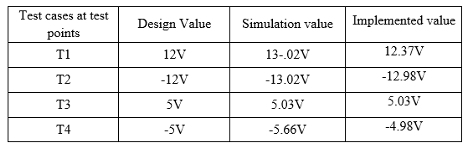
TP6: Send initialization commands LCD and test it turn ON

TP7: Send char in serial port to WiFi and check output.

7.3 Test Result

* Tabulating test results.
* (For all the test case scenarios listed with respect to design, simulation and implementation aspects.
* Measure the sensed values for a range, represent data in tabular form, visualize using graphs, and describe the interpretations
* Measurement of various system characteristics and performance such as resolution, efficiency, errors, accuracy, linearity, latency, power aware etc. also
* should be tabulated with detailed description.)

Table 7.1 Observations of the Temperature unit



Observations:

Table 7.2 include Observations for each unit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Computed value | Simulated value | Analog value | Digital value |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* Reflect on observations.

e.g.

Referring to the table 7.1 it observed that the simulation values are very close to the actual values which reflect the proper functioning of real prototype to meet the objectives. The error is less than 10% and the accuracy of the output is within 90%

**Visualise the results with appropriate graphs and describe the insights.**

e.g.

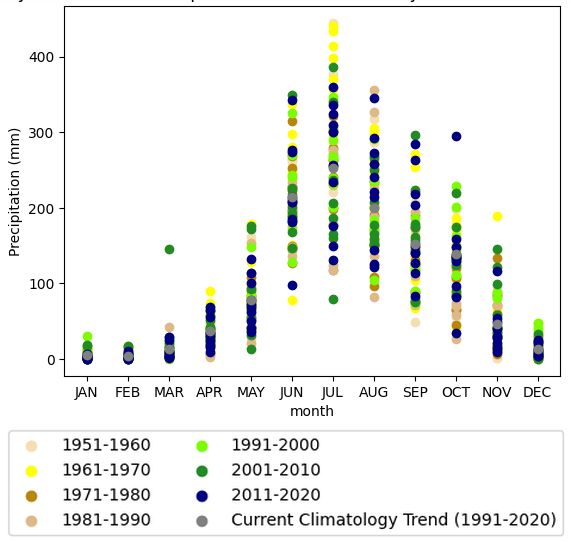


Fig 7.2 Variability and trends of precipitation across seasonal cycle of Karnataka region

Fig. 7.2 shows the variability and trends of precipitation across seasonal cycle of Karnataka region for the period 1951-2020 to observe the change in precipitation of each month over the years. Precipitation is usually high in the monsoon month of July and low in the non-monsoon month of December, being highest during period 1951-1960 and lowest in the period 2001-2010. Comparing the average trend values of the latest decade 1991-2020 with earlier decades, the precipitation values are lower in recent decades compared to earlier years indicating climatology trend of precipitation decrease over the years.

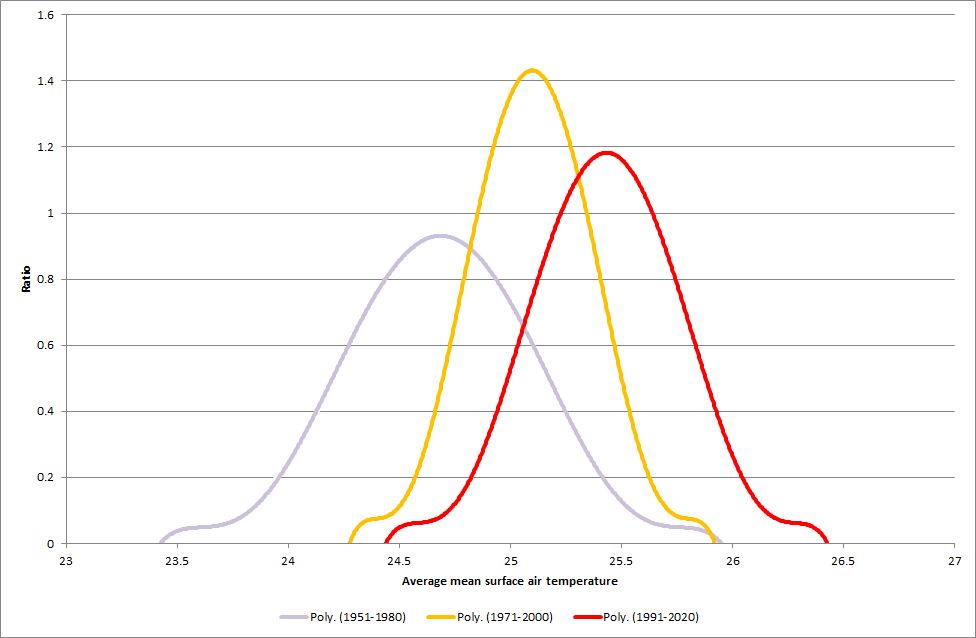


Fig. 7.3: Distribution of Average mean surface air temperature of Karnataka region.

Fig. 7.3 shows change in distribution of average mean surface air temperature of Karnataka region for three periods 1951- 1980, 1971-2000 and 1991-2020. It is observed the peaks have shifted 24.50C, 25.150C and 25.60C respectively indicating increase 1.10C in temperature value compared to the previous period. Moreover, the shift does not only signify increase in the peak but also the range of temperatures in later periods. The distribution suggests that the higher temperature event is more often than before.

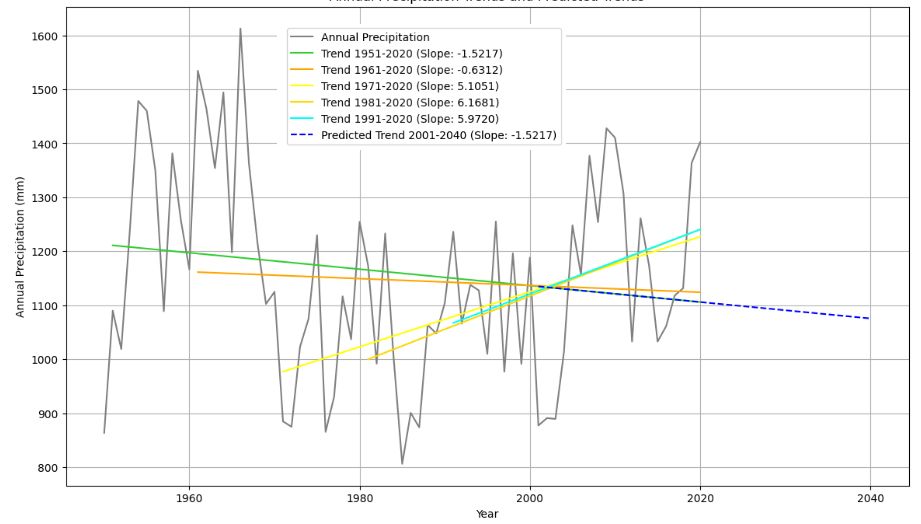


Fig. 7.4: Trend line of Precipitation per decade of Karnataka region.

Fig. 7.4 shows the Trend line of the Precipitation per decade of Karnataka region. The annual average mean precipitation values have gradually fluctuated over the period of 1951-2020. Although the recent years 1971-2020, 1981-2020 and 1991-2020 shows an increasing precipitation, there is a decreasing trend compared to the previous period of 1951-2020 and 1961-2020. The trend line 2001 to 2040, is directed downward predicting a decrease in the precipitation.

**Other examples of graph**

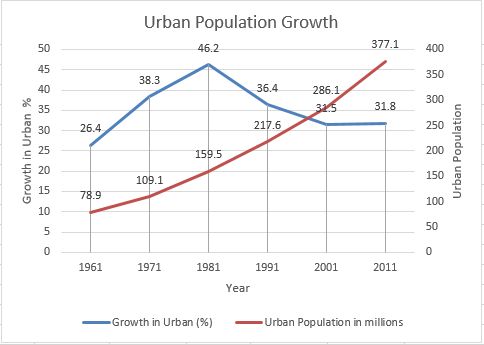


Fig. 7.5 Fig. 1: Urban population and rate of growth of urban population [1].

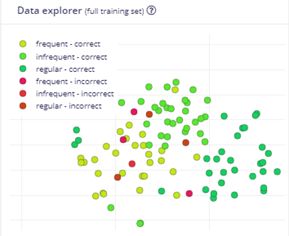
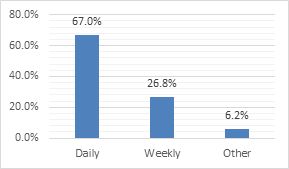
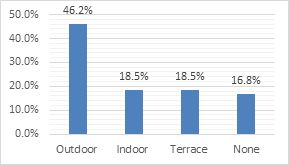


Fig. 7.6: Data explorer view of correct and incorrect predictions.





(a) Type of garden owned. (b) Watering frequency of plants.

Fig. 7.7: Garden owned and frequency of watering plants

**Provided insights for each figure, graph, table**

7.4 Insights

**Discuss insights. Give reasons for the problem**

e.g.

* What was the reason for the component getting burnt?
* What are the conditions under which some components do not work?
* Discuss the evaluation in terms of working principle, signal conditioning, latency, linearity, resolution, efficiency, error, accuracy etc.

e.g.

* The regulator provides current 1.3A which is close to the maximum rating and gets heated. To avoid exceeding ratings the current may be driven using regulator of higher rating, using heat sink to avoid burning, or multiple regulators to distribute currents to different devices.
* The microcontroller requires 200ms to return from sleep state but the delay does not affect the measurement as temperature variation is slower.
* As per datasheet the minimum voltage the 10 bit ADC of the microcontroller can measure is 4.89mV and lower values cannot be measured with precision. A high precision ADC would be required for measuring smaller values.
* The maximum output voltage of solar panels is 18V that can damage the input to microcontroller and requires interfacing circuits to shift level below 5V to avoid damage to microcontroller.
* The measurement of distance by ultrasonic sensor is 34cm whereas the actual value is 31cm which is within 10% accuracy. Ultrasonic sensors are affected by oil, humidity and heat, thus affecting the readings and their instability.
* The current sourced by microcontroller digital pin is 40mA and DC current between Vcc and Ground is 200mA. It cannot be used to drive the relay that requires 100mA current and so relay drivers ULN2003 are required. It cannot drive the DC motor and L298N is used
* The accuracy between the design, simulation and implementation values is within 10% Although the error is 2% in measurement of time it is not cumulative and remains constant for all values.

**Discuss the aspects to improve each functional units in terms of reliability, power aware, interrupt driven, precise timing, meeting standards, safety etc.**

e.g.

* The switch allows system configuration in OFF mode and normal operation in ON mode.
* Interrupt driven programming is used. The microcontroller is programmed to sleep state. When an event occurs an interrupt is generated to wake the system. The measurement and processing is completed and once again the system is put to sleep mode. The power in wake mode is 6.5mW and 2uW in sleep mode, thereby saving power. The event occurs every 500ms but It takes 100ms to wake, so no value is lost.
* The standard frequency 2.4GHz is the used for communication
* The freewheeling diode is connected across the relay coil to avoid damage to circuit.
* The system has a power switch and reset switch to provide safety and power saving when not in use.
* One possible solution is to change the ultrasonic sensor with the waterproof ultrasonic sensor.

Chapter 8

Social, Legal, Ethical, Sustainability and Safety aspects

The actions that society finds acceptable versus the actions which society do not accept.

Unfortunately, not all situations are morally well defined.

* Who is responsible for assuring the safe, legal, and ethical use of this project?
* What are the consequences of dishonesty in system/project use, individual and professional?
* How does ethical analysis apply to activities that are against the law?
* **Ensure proper citation and referencing.**

8.1 Social Aspects

**Social aspects** address how a technology or product affects society, including human interactions, communities, and culture.

* **Ensure proper citation and referencing.**

e.g. impact of AI

**Positive impacts**: Enhanced communication through social media and messaging, improved healthcare with advancements like telemedicine, and broader access to information and education via online platforms.

**Negative impacts**: The digital divide, which exacerbates inequality by leaving those without access to technology behind. Social media can also contribute to social isolation, mental health issues, and cyberbullying.

**Case study**: AI: The social impact of Artificial Intelligence (AI) includes changes to the workforce due to automation, the potential for algorithmic bias to reinforce societal inequalities, and challenges to human dignity if AI replaces roles that require empathy, such as caregiving or counseling.

8.2 Legal Aspects

**Legal aspects** concern regulations and compliance, with a strong focus on data privacy in the digital age.

* **Ensure proper citation and referencing.**

e.g. impact of AI

**Data privacy laws**: Landmark legislation such as the EU's General Data Protection Regulation (GDPR) and India's Digital Personal Data Protection Act (DPDPA) of 2023 establish frameworks for handling personal data. Key principles include lawful and fair processing, purpose limitation, data minimization, and accountability.

**Rights and obligations**: Under such laws, individuals have rights to access, correct, and erase their personal data. Data fiduciaries (those who process data) are obligated to obtain consent, implement security safeguards, and manage grievances.

**Challenges**: Regulatory compliance is complex for multinational companies. For technologies like autonomous vehicles, determining liability in the event of an accident presents a new legal challenge.

8.3 Ethical Aspects

**Ethical aspects** involve moral principles guiding the development and use of technology, considering fairness, accountability, and potential harm.

Ethical action on the part of engineer can be partially simplified, as the simple mandate that an engineer's greatest responsibility is to the public good.

* What effects have your project had on the quality of life in the workplace, and in society in general?
* Are your projects addictive?
* Do they depersonalize the individual to the point that ethical issues no longer seem relevant?
* How do electronics engineering professionals determine ethical standards to deal with the effects that their work has in these areas?
* **Ensure proper citation and referencing.**

e.g. impact of AI

**AI ethics**: This field addresses issues like algorithmic bias and fairness, which can arise when AI is trained on skewed data, leading to biased outcomes in areas like hiring or criminal justice. Other key concerns include transparency (the "black box" problem), data privacy, and accountability.

**Robotics**: Ethical questions also surround robot rights, the use of AI in lethal autonomous weapons, and how to program morally sound decision-making into autonomous systems.

**Generative AI**: Ethical debates also encompass copyright and intellectual property, as generative AI models are trained on existing human-created content.

8.4 Sustainability Aspects

**Sustainability aspects** relate to the environmental impact of a product or process, especially throughout its lifecycle.

Sustainable design is a kind of design meant to yield products that are made only of renewable resources. Furthermore, products made, though sustainable design is intended not to seriously impact the environment either when they are being created or when they are being used. These products are also often designed to allow the users to feel more connected or to relate more closely to the natural environment. This design is based on economic, ecological, and social principles regarding the importance of sustainability.

Discuss the relevant sustainability aspects of your project by answering some of the principles (but not restricted to these principles only) of sustainable design as follow:

* + Efficient use of raw materials (e.g. lightweight and regenerative materials and reduction of waste)
  + Resource efficient design (e.g. by optimization of energy and water consumption or using less virgin materials)
  + Durable design (e.g. stable and high durability of the product)
  + High disposability (e.g. use of easily degradable materials)
  + Efficient logistics (e.g. minimized packing and efficient transports)
* Consumer health & safety (e.g. informing the consumer via product labels and voluntary information
* **Ensure proper citation and referencing.**)

e.g. impact of AI

**Supply chain sustainability**: Businesses are increasingly focused on creating sustainable supply chains by adopting ethical sourcing, using renewable energy, and managing waste. Measures include reducing emissions from transportation, designing products for recyclability, and establishing transparency with suppliers.

**Environmental considerations**: The goal is to minimize environmental harm, including protecting natural resources and reducing greenhouse gas emissions. Technologies like AI and advanced analytics are being used to optimize supply chains and promote circular economy principles.

**Social dimension of sustainability**: A sustainable approach also incorporates social responsibility, ensuring fair labor practices and safe working conditions throughout the supply chain.

8.5 Safety Aspects

**Safety aspects** focus on preventing harm and ensuring the security and reliability of products and systems**.**

**(Discuss and describe the safety aspect from the perspective of the project**

* **Ensure proper citation and referencing.**

e.g.

The Internet of Things (IoT) in Health, Safety, and Environment (HSE) refers to the application of connected physical devices, sensors, and network technology to optimize and enhance processes related to **occupational health, workplace safety, and environmental management**. This integration enables real-time data collection, facilitating informed decision-making and promoting safer and more sustainable practices.

IoT safety is the process of protecting IoT devices, networks, and data from threats by using best practices like **strong, unique passwords and multi-factor authentication, regular firmware updates, data encryption, creating isolated networks** for IoT devices, and monitoring for suspicious activity.

IoT also improves **safety by enabling real-time monitoring of environmental hazards, providing data-driven insights for risk prevention, and improving emergency response** through alerts and guided evacuations.)

e.g. impact of AI

**Autonomous vehicles**: Safety is paramount for self-driving cars, requiring robust sensor suites, extensive software testing, and cybersecurity measures to prevent hacking. Redundancy and fail-safe mechanisms are built in to ensure functional safety.

**AI systems**: Safety guidelines are also crucial for AI, including developing systems that are robust and reliable under various conditions and implementing continuous monitoring. The goal is to prevent AI from producing harmful or unintended outputs.

**Cybersecurity**: This is a key component of safety, particularly in technology. It involves protecting systems from cyberattacks, data breaches, and unauthorized access.

Chapter 9

Conclusion

(Summarise the approach used in the project, the results obtained and future work)

* Discuss how the implementation of this project meets the objectives listed in the Introduction.
* Summarize the results and link it with the objectives of the project
* Future recommendation - Reflect on design aspects that can be improved but not implemented in your project.)

e.g.

Solar tracking systems designed in the proposed work fulfills the optimized structure in different scenarios based on sunlight. The designed prototype is IoT based to monitor and control the system and also serves as an efficient and a sustainable solar tracking device…….

In the future, this project can be developed and improved to make it more efficient and usable by......

References

(When you write a research paper or report, including a list of references is an important step. Below are some guidelines for selecting and formatting your references.

Include **references of all sources** used in chapters – 1. Introduction, 2. Literature review, 3. Methodology, 4. Project management, 5. Analysis and Design, 6. Hardware and Software implementation, 7. Evaluation and results, and 8. Social, Legal, Ethical, Sustainability and Safety aspects.

**Reference Selection**

* Make sure all references are directly **relevant** to the topic/project.
* You should cite **recent conference and journal papers**, preferably those published within the **last three to five years**.
* If you include any figures or tables from other sources, they must be properly cited as well.

**Citation and Formatting**

* Every source you cite in the body of the report must appear in your reference list,
* Every source in your reference list must be cited in the text.
* If you cite a source multiple times, you only need to include one entry for it in your reference list.
* Finally, do not include any references in your list that you haven't cited in your paper.

**Use Havard style** – e.g. author name, year, title of paper, conference or publication, vol, issue, page no, ISBN no …

e.g.

[1] United Nations, Sustainable Development Goals, Department of Economic and Social Affairs UN, https://sdgs.un.org/goals

[2] NarasimhaRao, Y., Chandra, P.S., Revathi, V. and Kumar, N.S., 2020. Providing enhanced security in IoT based smart weather system. Indonesian Journal of Electrical Engineering and Computer Science, 18(1), pp.9-15.

[3] Raju, M.P. and Laxmi, A.J., 2020. IOT based online load forecasting using machine learning algorithms. Procedia Computer Science, 171, pp.551-560.

[4] ….

**Do not summarise from websites, wikipedia, blogs, etc**.)

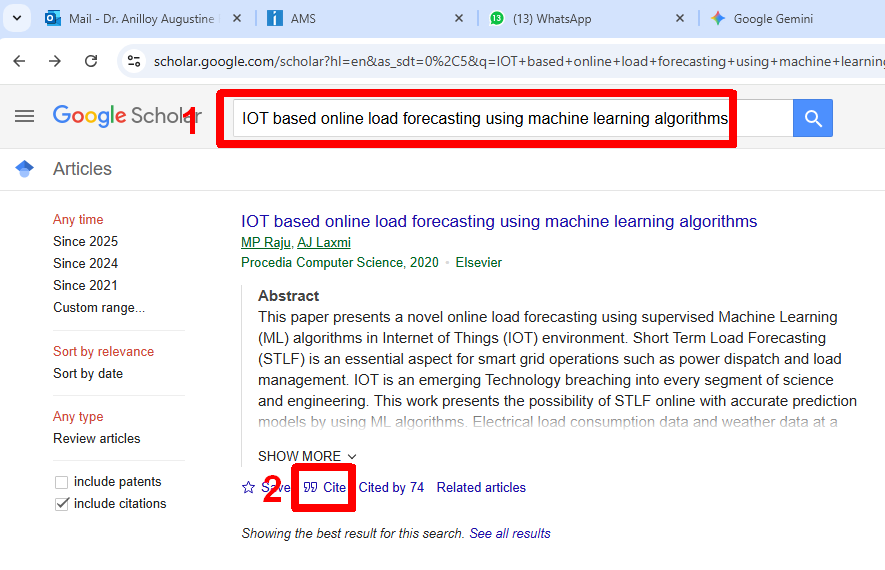
Base Paper

(From References the mainly referred paper)

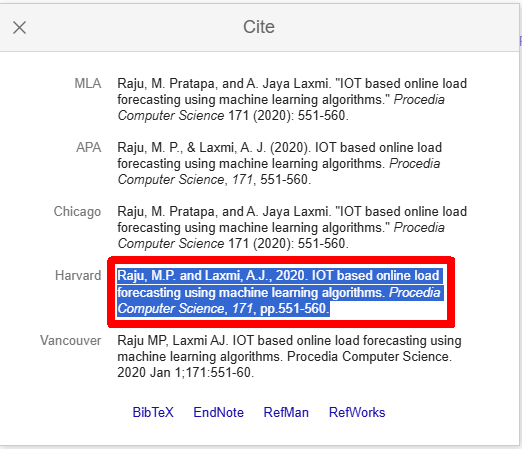
**Use the instructions below to format your references to match the example in the "References" section above.**

Step 1. Visit <https://scholar.google.com/>

Step 2. **Search the title** of the publication. (screenshot below)



Step 3. Click on **“Cite “** link below the searched publication. It will automatically generate the citation in standard formats (screenshot below)



Step 4. Choose the **Harvard** style and copy-paste in reference

Appendix

(Include summary of specifications of components from datasheet, and a few images of the project demonstrating different scenarios)

1. Data sheets –
2. Publications – Acceptance letter, Scopus URL, Certificate.
3. Project Report- Similarity Report
4. Few images of project –
5. Any auxiliary documents –
6. Datasets
7. Etc.

**Checklist**

* **Students to print checklist separately.**
* Confirming the checklist is only for ensuring completeness, correctness and quality of report. Use the checklist to confirm all aspects are fulfilled.
* Confirming the checklist does not assure full marks. In addition to report, marks are awarded for other aspects of the project viz. implementation and presentation.
* **Do not include this checklist in the report.**
  + 1. Confirm **Formatting** – line spacing, paragraph justification, font size.
    2. Confirm **Text** – details, spellings, grammar
    3. Confirm **Figures and Tables** - in-text citation, description
    4. Confirm **Figures and Tables** – updated in List of Figures and Tables
    5. Confirm **Citation** – all sources referred are cited
    6. Confirm **References** –all sources referred are updated in Reference section

**Faculty to mark ‘X’ in columns not confirming**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Description** | **A** | **B** | **C** | **D** | **E** | **F** |
|  | **Declaration** |  |  |  |  |  |  |
|  | **Acknowledgement** |  |  |  |  |  |  |
|  | **Abstract** |  |  |  |  |  |  |
|  | **Table of content** |  |  |  |  |  |  |
|  | – confirm Section, sub-sections from all chapters included. |  |  |  |  |  |  |
|  | – confirm Page no. of Section, sub-sections updated |  |  |  |  |  |  |
|  | – confirm Page no. are aligned to right |  |  |  |  |  |  |
|  | **List of Figures** |  |  |  |  |  |  |
| – confirm Caption of Figures from all chapters included. |  |  |  |  |  |  |
| – confirm Page no. of figures updated |  |  |  |  |  |  |
| – confirm Page no. are aligned to right |  |  |  |  |  |  |
| – confirm Figures are numbered correctly |  |  |  |  |  |  |
|  | **List of Tables** |  |  |  |  |  |  |
| – confirm Caption of Tables from all chapters included. |  |  |  |  |  |  |
| – confirm Page no. of tables updated |  |  |  |  |  |  |
| – confirm Page no. are aligned to right |  |  |  |  |  |  |
| – confirm Tables are numbered correctly |  |  |  |  |  |  |
|  | **Abbreviations** |  |  |  |  |  |  |
| – confirm Abbreviations used in all chapters included |  |  |  |  |  |  |
| – confirm Abbreviations are in alphabetic order. |  |  |  |  |  |  |
| – confirm Abbreviations appear only once in list and are not repeated |  |  |  |  |  |  |

* + 1. Confirm **Formatting** – line spacing, paragraph justification, font size.
    2. Confirm **Text** – details, spellings, grammar
    3. Confirm **Figures and Tables** - in-text citation, description
    4. Confirm **Figures and Tables** – updated in List of Figures and Tables
    5. Confirm **Citation** – all sources referred are cited
    6. Confirm **References** –all sources referred are updated in Reference section

**Faculty to mark ‘X’ in columns not confirming**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Description** | **A** | **B** | **C** | **D** | **E** | **F** |
| 1. | **Introduction** |  |  |  |  |  |  |
| 1.1 **Background** |  |  |  |  |  |  |
| 1.2 **Statistics of project** |  |  |  |  |  |  |
| 1.3 **Prior existing technologies** |  |  |  |  |  |  |
| 1.4 **Proposed approach** |  |  |  |  |  |  |
| 1.5 **Objectives** |  |  |  |  |  |  |
| 1.6 **SDGs** |  |  |  |  |  |  |
| 1.7 **Overview of project report** |  |  |  |  |  |  |
| 2. | **Literature review** |  |  |  |  |  |  |
| 3. | **Methodology** |  |  |  |  |  |  |
| 4. | **Project management** |  |  |  |  |  |  |
| 4.1 **Project timeline** |  |  |  |  |  |  |
| 4.2 **Risk analysis** |  |  |  |  |  |  |
| 4.3 **Project budget** |  |  |  |  |  |  |
| 5. | **Analysis and Design** |  |  |  |  |  |  |
| **5.1 Requirements** |  |  |  |  |  |  |
| **5.2 Block Diagram** |  |  |  |  |  |  |
| **5.3 System Flow Chart** |  |  |  |  |  |  |
| **5.4 Choosing devices** |  |  |  |  |  |  |
| **5.5 Designing units** |  |  |  |  |  |  |
| **5.6 Standards** |  |  |  |  |  |  |
| **5.7 Mapping with IoTWF reference model layers** |  |  |  |  |  |  |
| **5.8 Domain model specification** |  |  |  |  |  |  |
| **5.9 Communication model** |  |  |  |  |  |  |
| **5.10 IoT deployment level** |  |  |  |  |  |  |
| **5.11 Functional view** |  |  |  |  |  |  |
| **5.12 Mapping IoT deployment level with functional view** |  |  |  |  |  |  |
| **5.13 Operational view** |  |  |  |  |  |  |
| **5.14 Other Design** |  |  |  |  |  |  |
| 6. | **Hardware, Software and Simulation** |  |  |  |  |  |  |
| **6.1 Hardware** |  |  |  |  |  |  |
| **6.2 Software development tools** |  |  |  |  |  |  |
| **6.3 Software code** |  |  |  |  |  |  |
| **6.4 Simulation** |  |  |  |  |  |  |

* + 1. Confirm **Formatting** – line spacing, paragraph justification, font size.
    2. Confirm **Text** – details, spellings, grammar
    3. Confirm **Figures and Tables** - in-text citation, description
    4. Confirm **Figures and Tables** – updated in List of Figures and Tables
    5. Confirm **Citation** – all sources referred are cited
    6. Confirm **References** –all sources referred are updated in Reference section

**Faculty to mark ‘X’ in columns not confirming**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Description** | **A** | **B** | **C** | **D** | **E** | **F** |
| 7. | **Evaluation and Results** |  |  |  |  |  |  |
| **7.1 Test points** |  |  |  |  |  |  |
| **7.2 Test plan** |  |  |  |  |  |  |
| **7.3 Test result** |  |  |  |  |  |  |
| **7.4 Insights** |  |  |  |  |  |  |
| 8. | **Social, Legal, Ethical, Sustainability and Safety Aspects** |  |  |  |  |  |  |
| **8.1 Social aspects** |  |  |  |  |  |  |
| **8.2 Legal aspects** |  |  |  |  |  |  |
| **8.3 Ethical aspects** |  |  |  |  |  |  |
| **8.4 Sustainability aspects** |  |  |  |  |  |  |
| **8.5 Safety aspects** |  |  |  |  |  |  |
| 9. | **Conclusion** |  |  |  |  |  |  |
|  | **References** |  |  |  |  |  |  |
|  | **Base Paper** |  |  |  |  |  |  |
|  | **Appendix** |  |  |  |  |  |  |

|  |
| --- |
| **Remarks:** |
|  |

**Tentative marking scheme**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Internship/Dissertation/Capstone Project/Minor Project - Evaluation Components** | **Weightage in %** | **Marks** |
| 1 | Review-1: Problem Statement Proposal & Objectives | 5% | 10 |
| 2 | Review-2:-Literature and Model Design presentation on the proposed work | 10% | 20 |
| 3 | Review-3 :Presentation on progress of the proposed work | 15% | 30 |
| 4 | Review-4 :Presentation on progress of the proposed work | 20% | 40 |
| 5 | Final VIVA :Submission of report & final presentation | 50% | 100 |
|  | **Total** | **100** | **200** |
|  |  |  |  |

NOTE: The above-mentioned evaluation rubrics will be applicable for Internship, Dissertation, Capstone Project, Minor Project, and other project-based courses from 2022-2026 Batch onwards.

**Evaluation Rubrics (Total: 200 Marks)**

## **Review 1 (10 Marks, 5%)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Description** | **Marks** | **Performance Indicators** |
| Title Finalization with Supervisor | Selection and finalization of a clear, concise, and relevant project title with faculty guidance. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Survey on Related Works | Identification and study of related work from reputed sources to justify the problem. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Identification of Generalized Objectives | Well-defined, specific, and measurable objectives derived from problem analysis. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Problem Identification | Clearly defines a relevant, current, and challenging problem aligned with societal or industrial needs. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Innovation | Novelty or improvement in the idea, solution, or methodology of existing methods. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| **Sub Total** |  | **10** |  |

## **Review 2 (20 Marks, 10%)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Description** | **Marks** | **Performance Indicators** |
| Abstract | Concise overview of the project covering problems, objectives, and approach. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Literature Survey (Minimum 10 Papers) | Comprehensive review and critical analysis of at least 10 relevant research papers using IEEE/Scopus Indexed/Web of Science sources. | 3 | Excellent (3)  Very Good (2.4–2.9)  Good (1.8–2.3)  Average (0.6–1.7)  Below Average (0–0.5) |
| Objectives | Specific, measurable, and outcome-focused project goals. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Existing Methods and Drawbacks | Summary and critique of current approaches and their limitations. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Proposed Method & Feasibility Study | Innovative solution explained with justification and feasibility in terms of technology, cost, and resources. | 3 | Excellent (3)  Very Good (2.4–2.9)  Good (1.8–2.3) Average (0.6–1.7) Below Average (0–0.5) |
| Architecture Diagram | Logical and clear representation of system architecture. | 3 | Excellent (3)  Very Good (2.4–2.9)  Good (1.8–2.3) Average (0.6–1.7) Below Average (0–0.5) |
| Modules | Well-defined system modules with clear functionalities. | 2 | Excellent (2)  Very Good (1.5–1.9)  Good (1.0–1.4)  Average (0.5–0.9)  Below Average (0–0.4) |
| Hardware and Software Details | Accurate and relevant list of technologies used. | 1 | Excellent (1)  Very Good (0.8–0.9)  Good (0.6–0.7)  Average (0.3–0.5)  Below Average (0–0.2) |
| Timeline (Gantt Chart) | Realistic and structured project timeline. | 1 | Excellent (1)  Very Good (0.8–0.9)  Good (0.6–0.7)  Average (0.3–0.5)  Below Average (0–0.2) |
| References | Properly cited in IEEE format. | 1 | Excellent (1)  Very Good (0.8–0.9)  Good (0.6–0.7)  Average (0.3–0.5)  Below Average (0–0.2) |
| **Sub Total** |  | **20** |  |

## **Review 3 (30 Marks, 15%)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Description** | **Marks** | **Performance Indicators** |
| Algorithm Details | Well-explained and optimized algorithms with visuals such as flowcharts or pseudo-code. | 5 | Excellent (5)  Very Good (4–4.9)  Good (3–3.9)  Average (2–2.9)  Below Average (0–1.9) |
| Source Code Details (50%) | Code implementation is modular, structured, and well-commented. | 5 | Excellent (5)  Very Good (4–4.9)  Good (3–3.9)  Average (2–2.9)  Below Average (0–1.9) |
| 50% Implementation with Live Demo | Working demo of at least 50% functionality of the system. | 15 | Fully Completed (10–15)  Partial Completion (5–9)  Minimal/No Completion (0–4) |
| 50% Report Softcopy Submission | Properly formatted draft report covering implemented parts. | 5 | Excellent (5)  Very Good (4–4.9)  Good (3–3.9)  Average (2–2.9)  Below Average (0–1.9) |
| **Sub Total** |  | **30** |  |

## **Review 4 (40 Marks, 20%)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Description** | **Marks** | **Performance Indicators** |
| Source Code Details (80%) | Complete implementation with structured, documented code. | 10 | Excellent (9–10)  Very Good (7–8.9)  Good (5–6.9)  Average (3–4.9)  Below Average (0–2.9) |
| 80% Implementation | Demonstrated 80% functionality of the system. | 20 | Fully Completed (10–20)  Partial Completion (0–9) |
| 80% Completed Report (Softcopy) | Completed report in softcopy, formatted per guidelines. | 10 | Fully Completed (10–20)  Partial Completion (0–9) |
| **Sub Total** |  | **40** |  |

## **Final VIVA (100 Marks, 50%)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Description** | **Marks** | **Performance Indicators** |
| Live Demonstration | Fully functional demo covering all features/modules. | 40 | Fully Functional (30–40)  75% Functional (20–29)  Partially Working (10–19)  Non-Functional (0–9) |
| Final Report (Hardcopy & Softcopy) | Professional report meeting PU formatting and guidelines. | 10 | As per Format (8–10) Partially Matched (4–7) Not Matched (0–3) |
| Plagiarism Report Submission | Turnitin similarity index ≤ 20%. | 10 | ≤20% Similarity (10)  21–40% (5)  >40% (0) |
| Publication of Paper | Paper published/accepted/communicated in journals or conferences. | 10 | Published (10)  Accepted (7)  Communicated (5)  Not Done (0) |
| SDG Mapping with Problem Statement | Project mapped to appropriate SDG(s). | 10 | Mapped (10)  Not Mapped (0) |
| GitHub Repository | Public GitHub repo with source code and documentation. | 10 | Excellent (9–10)  Very Good (7–8.9)  Good (5–6.9)  Average (3–4.9) Below Average (0–2.9) |
| Answering Ability | Clarity and accuracy in answering technical questions. | 10 | Excellent (9–10)  Very Good (7–8.9)  Good (5–6.9)  Average (3–4.9)  Below Average (0–2.9) |
| **Sub Total** |  | **100** |  |
|  | **Total** | **200** |  |

**General Rating Scale Based on Weightage for Each Criterion**

- Excellent (4): 85–100% of allocated marks  
- Good (3): 70–84% of allocated marks  
- Satisfactory (2): 50–69% of allocated marks  
- Poor (1): Below less than 50% of allocated marks

- Absent/Not Reported (0): 0% of allocated marks