**Guidelines for Project Report**

After completion of the project work, every student will submit a project report which should contain the following:

1. **Cover Page** (as per annexure 1)
2. **Title page** (as per annexure 1)
3. **Declaration by the Student** (as per annexure 2)
4. **Acknowledgement** (The candidate may thank all those who helped in the execution of the project.)
5. **Abstract** (It should be in one page and include the purpose of the study; the methodology used and a summary of the major findings.)
6. **Table of Contents** (as per annexure 3): Detailed description of the project (This should be split in various chapters/sections with each chapter/section describing a project activity in totality). This portion of report should contain all relevant diagrams, tables, flow charts, software programme, print outs, photographs etc., which are properly labeled.

#### Conclusion & Recommendations

1. **Appendices** (if any)
   * Appendices are provided to give supplementary information, which if included in the main text may serve as a distraction and cloud the central theme.
   * Appendices should be numbered using Arabic numerals, e.g. Appendix 1, Appendix 2.
   * Appendices shall carry the title of the work reported and the same title shall be listed in the Contents page also
2. **References** (The listing of references should be typed 2 spaces below the heading “REFERENCES” in alphabetical order in single spacing left – justified. It should be numbered consecutively (in square [ ] brackets, throughout the text and should be collected together in the reference list at the end of the report. The references should be numbered in the order they are used in the text. The name of the author/authors should be immediately followed by the year and other details). Typical examples of the references are given below:

#### REFERENCES

[1]. Ariponnammal, S. and Natarajan, S. (1994) ‘Transport Phonomena of SmSel – X Asx’, Pramana – Journal of Physics Vol.42, No.1, pp.421-425.

In addition, following points should be complied with:-

1. Page numbering
2. Numbering of appendices, figures and tables and their reference in the text.
3. For general layout of report, any standard text book layout may be referred.

#### Report Specifications:

1. Project Report’s Cover Type: **PDF-File only**
2. Number of Copies: **1 per student**
3. Paper Size (orientation): **A4** (portrait)

#### Margins: 1” top / bottom / right and 1.5” left

1. Font Type: **Times New Roman**
2. Font Size: **16 bold for chapter names**, **14 bold for headings** and **12 for normal text**
3. Line Spacing: **1.5 throughout**
4. Page Numbering: For introductory pages like Acknowledgment, Declaration, List of Tables, List of Figures, List of Symbols, List of Abbreviations ---Refer to the relevant Appendices. For the remaining file--- Bottom center of page in the format – **Page 1 of N**

**NOTE:** Project report must not contain any description of the following except only a relevant and short mention – technology or platform or OS or tools used or any language details. It must be more focused on project work carried out and its implementation details without including any source code.

##### Annexure-1 (A typical Specimen of Title Page)

**TITLE OF PROJECT**

<Font Size 24><BOLD><Centralized>

### A Project Work

<Font Size 14><BOLD><Centralized>

*Submitted in the partial fulfillment for the award of the degree of*

<Font Size 14><1.5 line spacing><Italic><Centralized>

# BACHELOR OF ENGINEERING

<Font Size 16>><BOLD><Centralized>

### IN NAME\_OF\_SPECIALIZED\_BRANCH

<Font Size 14><BOLD><Centralized>

### Submitted by:

<Font Size 14><BOLD><Centralized>

### NAME OF THE STUDENT

<Font Size 14>><BOLD><Centralized>

### University Roll Number

<Font Size 14><BOLD><Centralized>

### Under the Supervision of:

<Font Size 14>><BOLD><Centralized>

### SUPERVISORS NAME

<Font Size 14>><BOLD><Centralized>



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

<Font Size 16><BOLD><Centralized>

### CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

**PUNJAB**

<Font Size 14><BOLD><Centralized>

#### MONTH & YEAR

<Font Size 12><BOLD><Centralized>

##### Annexure-2

**DECLARATION**

<Font size 16><Bold><Centralized>

I, **‘Candidate Names’**, student of **‘Bachelor of Engineering in Branch Name’**, **session:** , Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Topic Name’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

<Font size 14><Regular><Justified>

#### Date:

**Place:**

<Font size 12><Bold><Left alligneed>

### (Candidate Name) Candidate UID: xxxxxxxxxxx

<Font size 14><Bold><Right Alligned>

ii

***Annexure-3* (A typical specimen of table of contents)**

# Table of Contents

|  |  |  |
| --- | --- | --- |
|  | Title Page  Declaration of the Student Abstract Acknowledgement  List of Figures  List of Tables (optional) Timeline / Gantt Chart | i |
| ii |
| iii |
| iv |
| v |
| vi |
| vii |
| **1.** | **INTRODUCTION\*** | **1** |
|  | * 1. Problem Definition   2. Project Overview/Specifications\* (page-1 and 3)   3. Hardware Specification   4. Software Specification 1.3.1   1.3.2  … | 1  2  3  4  4 |
| 1. **LITERATURE SURVEY**    1. Existing System    2. Proposed System    3. Feasibility Study\* (page-4) 2. **PROBLEM FORMULATION** | | **5** |
| 5  6  7 |
| 9  16  18  19  22 |
| 1. **OBJECTIVES** 2. **METHODOLOGY** 3. **CONCLUSIONS AND DISCUSSION** 4. **REFERENCES** | |

## List of Tables

##### Table Title page

***3.1*** *Quantities of Materials Required in the Designs with Different* ***10***

*Grades of Concrete*

.

vii

## List of Figures

##### Figure Title page

* 1. *Joint in a steel moment resisting frame (a) geometry, and (b) in-plane 11*

*lateral distortional shear force on it. Results of analytical study (a)*

* 1. *Idealised trilinear model used in this study of or RC Frame 11*

*buildings with masonry infilled walls; (b) Mean DRF*

*spectra of Uttarkashi earthquake strong motions records derived for bare and masonry infilled RC frame buildings characteristics with k= 2, =2,and 0.2. The spectra*

*correspond to ductility values of 1,2,3,5,8,10,12 and 15. Dark and dashed lines correspond to bare and infilled frame buildings respectively.*

viii

## List of Symbols

*Symbol Description*

***Ast Asc Asv b d d’***

***fc,ave fsc***

##### fy Sv xu

***~~x~~***

*τ* ***c***

*Area of steel reinforcement bars on tension face*

*Area Of steel reinforcement bars on compression face Area of two legs of the closed stirrups*

*Breadth of rectangular beam section Effective depth of rectangular beam section Effective cover on compression face Average compressive stress in concrete Stress in steel on the compression side*

*Characteristic strength of steel reinforcement bars Spacing of the stirrups*

*Depth of neutral axis from compression face*

*Depth of centroid of the compression block in concrete Shear strength offered by concrete*

ix

# INTRODUCTION

#### 1.1

**1.1.1**

1. **LITERATURE REVIEW**

Kim et al. [25] proposed VUDDY, which is a scalable approach for detection of vulnerable code clones. This approach can detect vulnerabilities efficiently and accurately in large software. They able to achieve extreme level of scalability by using function-level granularity and a length-filtering techniques that decreases number of signature comparisons. Most interesting feature of this technique is that it can even detect variants of known vulnerabilities. To achieve extreme level of scalability, they used function-level granularity and length-filtering techniques to reduce number of signature comparisons.

**2.1 Literature Review Summary**

Table 2.1: Literature review summary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Year and citation** | **Article Title** | **Purpose of study** | **Tools / Software Used** | **Comparison of technique** | **Source (Journal / Conference)** | **Findings** | **Data set (if used)** | **Evaluation parameters** |
| 2010 |  |  |  |  |  |  |  |  |

# PROBLEM FORMULATION

During software development, clones can occur in software intentionally or unintentionally. Developers tend to clone fragments of software during development to save efforts and expedite the development process. ........................................

From the literature review, it is observed that studies highlight the need of efficient and scalable approach for detecting code clones having software vulnerability. The existing techniques are not able to detect all types of vulnerable code clones. Different approaches suffer from high false negative rate and not scalable to large software systems due to high time complexity. So firstly, there is a need........................................ Second same subject systems should be used to compare the approaches which detect

# RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of an approach for vulnerable code clone detection. The proposed aim will be achieved by dividing the work into following objectives:

* 1. To understand and explore various types of software vulnerabilities existing in open source software.
  2. To study and analyse various clone detection techniques that are suitable for vulnerable code clone discovery.
  3. To design and develop the technique for vulnerable code clone detection.
  4. To verify and validate the proposed system.

# METHODOLOGY

The following methodology will be followed to achieve the objectives defined for proposed research work:

* 1. Detailed study of software vulnerabilities, their types and impact of code cloning practice on software vulnerabilities will be done.
  2. Installation of clone detection tools and hand on experience on existing approaches of clone detection applicable for software code clone vulnerability will be done. Relative pros and cons will be identified.
  3. Vulnerability database will be created, which corresponds to the C/C++ open source projects that have some vulnerabilities according to National Vulnerability Database (NVD) [36] and Open Sourced Vulnerability Database (OSVDB) [51].
  4. Different clone detections techniques will be analysed to figure it out which clone detection technique is appropriate for particular type of vulnerability.
  5. An approach will be developed for vulnerable code clone detection.
  6. Various parameters will be identified to evaluate the proposed system.
  7. Comparison of new implemented approach with exiting approaches will be done.

1. **RESULTS AND DISCUSSION**

# REFERENCES

1. D. Rattan, R. Bhatia, and M. Singh, “Software clone detection: A systematic review,”

*Information and Software Technology*, vol. 55, no. 7, pp. 1165–1199, Jul. 2013.

1. J. F. Islam, M. Mondal, and C. K. Roy, “Bug Replication in Code Clones: An Empirical Study,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 68–78.
2. M. R. Islam and M. F. Zibran, “A Comparative Study on Vulnerabilities in Categories of Clones and Non-cloned Code,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 8–14.
3. M. R. Islam, M. F. Zibran, and A. Nagpal, “Security Vulnerabilities in Categories of Clones and Non-Cloned Code: An Empirical Study,” in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017, pp. 20–29.
4. C. K. Roy, M. F. Zibran, and R. Koschke, “The vision of software clone management: Past, present, and future (Keynote paper),” in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 18–33.
5. J. Krinke, “A Study of Consistent and Inconsistent Changes to Code Clones,” in *14th Working Conference on Reverse Engineering (WCRE 2007)*, 2007, pp. 170–178.
6. D. Chatterji, J. C. Carver, N. A. Kraft, and J. Harder, “Effects of cloned code on software maintainability: A replicated developer study,” in *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013, pp. 112–121.
7. D. Rattan, R. Bhatia, and M. Singh, “An Empirical Study of Clone Detection in MATLAB/ Simulink Models,” *International Journal of Information and Communication Technology*.
8. D. Rattan, R. Bhatia, and M. Singh, “Detecting High Level Similarities in Source Code and Beyond,” *International Journal of Energy, Information and Communications*, vol. 6, no. 2, pp. 1–16, 2015.
9. D. Rattan, R. Bhatia, and M. Singh, “Detection and Analysis of Clones in UML Class Models,” *International Journal of Software Engineering, IJSE*, vol. 8, no. 2, pp. 66– 99, 2015.
10. D. Rattan, R. Bhatia, and M. Singh, “Model clone detection based on tree comparison,” in *2012 Annual IEEE India Conference (INDICON)*, 2012, pp. 1041– 1046.
11. C. K. Roy and J. R. Cordy, “NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization,” in *2008 16th IEEE International Conference on Program Comprehension*, 2008, pp. 172–181.
12. M. Mondal, C. K. Roy, and K. A. Schneider, “SPCP-Miner: A tool for mining code clones that are important for refactoring or tracking,” in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2015, pp. 484–488.