INTERNSHIP: INTERIM PROJECT REPORT

Dear Intern

Interim project report is an inherent component of your internship. We are enclosing a reference table of content for the interim project report.

The key objective of this report is for you to capture how far you have got in completing the internship work against milestones expected to be achieved within a specific duration and seek the mentor's feedback. Depending on the internship project and your progress (IT/Non-IT, Technical/Business Domain), you may choose to include or exclude or rename sections or leave some sections blank from the table of content mentioned below. You can also add additional sections. You can refer the project presentation to view the milestones related to your internship project. Please populate milestone# (1 / 2 / 3) and the milestone description in the interim project report based on the milestone for which you are submitting the interim project report.

You can refer the project presentation to view the milestones related to your internship project.

Internship Project Title	TCS ION RIO - 45: Create a CAPTCHA Service to Secure a Simple Web-based Application
Name of the Company	TCS ION
Name of the Industry Mentor	Niteen Gokhale
Name of the Institute	Sri Ramakrishna College of Arts and Science

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
15.07.2025	19.07.2025	4 Hours	-	Java/JSP

Milestone #	1	Milestone:	Develop a CAPTCHA service with, API to deliver a CAPTCHA image and API to validate user-entered CAPTCHA text.

TABLE OF CONTENT

- Acknowledgements
- Objective
- Introduction / Description of Internship
- Internship Activities
- Approach / Methodology
- Assumptions
- Exceptions / Exclusions
- Charts, Table, Diagrams
- Algorithms
- Challenges & Opportunities
- Risk Vs Reward
- Reflections on the Internship
- Recommendations
- Outcome / Conclusion
- Enhancement Scope
- Link to code and executable file
- Research questions and responses

Acknowledgements

I would like to express my sincere gratitude to TCS ION for providing this internship opportunity through the RIO-45 platform. I would also like to thank my faculty mentors and peers for their valuable support, and my family for their constant encouragement throughout this journey.

Objective

The objective of this internship is to develop a secure CAPTCHA service to protect a simple web-based application from automated bot access. The focus for Milestone 1 was to:

- Understand the role and necessity of CAPTCHA.
- Classify and evaluate different CAPTCHA schemes.

Develop APIs to:

- Generate and deliver a CAPTCHA image.
- Validate user-submitted CAPTCHA input.

Introduction / Description of Internship

This internship under the TCS ION RIO-45 program focuses on implementing security features in web applications using Java-based technology. Milestone 1 introduced the foundations of CAPTCHA:

- Studying its importance in user authentication.
- Classifying its types based on usability and security.
- Creating a working CAPTCHA service using Java Servlets and JSP that can be embedded into web apps.

Internship Activities

- Research → Explored various CAPTCHA systems: Text-based, Image-based, Audio, Math-based, Google reCAPTCHA.
- Evaluation → Compared CAPTCHA schemes based on security (bot resistance) and usability (user experience).
- Design → Chose Text-based CAPTCHA for its balance between ease of development and effectiveness in blocking bots.
- Development → Created Java Servlets: `CaptchaGenerator` – Generates CAPTCHA image with distortion. 'CaptchaServlet` – Validates user input.
- Frontend → Designed a basic frontend `index.jsp` page: Displays CAPTCHA image Accepts user input Sends data for validation.
- Testing → Deployed the app on Apache Tomcat, verified: CAPTCHA generation Input validation Session handling.

Approach / Methodology

Understanding Concepts – Studied the purpose and evolution of CAPTCHA technologies.

Scheme Evaluation – Assessed various CAPTCHA types for accuracy and user-friendliness.

Development Process:

- Used Java Servlets for backend logic.
- Generated CAPTCHA image dynamically with random characters and distortion.
- Stored the CAPTCHA string in HttpSession.
- Validated user input via LoginServlet.
- Testing – Performed functional testing using Apache Tomcat.

Assumptions

- CAPTCHA interaction is through desktop browser only.
- Only alphanumeric characters are used.
- One CAPTCHA per user session.
- No real user login integration is performed yet.

Exceptions / Exclusions

CAPTCHA refresh feature not yet implemented.

Third-party CAPTCHA libraries (e.g., reCAPTCHA) not used.

No support for accessibility options like audio CAPTCHA in this phase.

Charts, Tables, Diagrams

CAPTCHA Type Evaluation Table:

CAPTCHA Type	Accuracy (Bot Resistance)	Ease of Use
Text-based	Medium	Medium
Image-based	High	Medium
Math-based	Low	High
Google reCAPTCHA	Very High	Very High

Algorithms

CAPTCHA Generation Algorithm:

1. Generate a random 5-character alphanumeric string.
2. Store it in the session: session.setAttribute("captcha", generatedCode).
3. Create BufferedImage and draw the text using Graphics2D.
4. Add noise lines, background color, distortion.
5. Write the image to HTTP response as PNG.

CAPTCHA Validation Algorithm:

1. Retrieve user input from form.
2. Retrieve session value (stored CAPTCHA).

3. Compare input with session-stored value.

4. If matched → redirect to success.jsp (or success message).

5. Else → redirect to error.jsp (or show error).

Challenges & Opportunities

Challenges:

Drawing distorted CAPTCHA images using Java AWT.

Ensuring correct session management.

Local deployment and testing on Tomcat.

Opportunities:

Practical learning of servlet-based backend.

Gained confidence in dynamic image generation and session handling.

Real-world relevance in securing web forms.

Risk Vs Reward

Risks	Rewards
CAPTCHA too difficult may frustrate users	Secure from automated bot submissions
CAPTCHA too easy may be bypassed	Practical exposure to web security implementation
Distorted image might be unreadable on all screens	Developed a working CAPTCHA that can be reused in future applications

Reflections on the Internship

This milestone gave real-world experience in secure web development using Java. It helped bridge classroom knowledge with implementation challenges. I gained hands-on skills in working with servlets, HTTP sessions, and response handling. The task strengthened my understanding of how CAPTCHA protects user data and server resources.

Recommendations

Add CAPTCHA refresh/reload button in next milestone.

Support mobile view and responsive design.

Explore advanced CAPTCHA types like drag & drop, puzzle-based.

Integrate with actual login logic for end-to-end protection.

Outcome / Conclusion

Successfully completed a basic CAPTCHA implementation using Java Servlets.

Created two functional APIs:

CAPTCHA generation (CaptchaServlet)

CAPTCHA validation (LoginServlet)

Developed a simple user interface via index.jsp.

Tested functionality locally using Apache Tomcat server.

Enhancement Scope

Implement CAPTCHA refresh button.

Store CAPTCHA history or log failed attempts.

Add audio CAPTCHA for accessibility.

Extend to mobile-friendly UI.

Use OCR-resistant fonts or patterns.

Link to Code and Executable File

https://github.com/R-Raghav30/TCS-RIO-45-CAPTCHA