

ASSESSMENT TOOL

**REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF**

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

Bidipta Saikia

Roll Number: 190102006

Manikangkan Das

Roll Number: 190102020

UNDER THE GUIDANCE OF

Mirzanur Rahman

Guides Affiliation



DEPARTMENT OF INFORMATION TECHNOLOGY

GAUHATI UNIVERSITY

GUWAHATI, INDIA

JUNE –2023



GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

DECLARATION

We, “**Bidipta Saikia**”, Roll No “**190102006**”, and “**Manikangkan Das**”, Roll No “**190102020**”, B.Tech. students of the department of Information Technology, Gauhati University hereby declare that we have compiled this report reflecting all our works during the semester long full time project as part of our BTech curriculum.

We declare that we have included the descriptions etc. of our project work, and nothing has been copied/replicated from other’s work. The facts, figures, analysis, results, claims etc. depicted in our thesis are all related to our full time project work.

We also declare that the same report or any substantial portion of this report has not been submitted anywhere else as part of any requirements for any degree/diploma etc.

Bidipta Saikia
Branch: IT
Date: 13/06/2023

Manikangkan Das
Branch: IT
Date: 13/06/2023



GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date: 13/06/2023

CERTIFICATE

This is to certify that "**Bidipta Saikia**" and "**Manikangkan Das**" bearing Roll No: "**190102006**" and "**190102020**" respectively, have carried out the project work "**Assessment Tool**" under my supervision and have compiled this report reflecting the candidates' work in the semester long project. The candidates did this project full time during the whole semester under my supervision, and the analysis, results, claims etc. are all related to their studies and work during the semester.

I recommend submission of this project report as a part for partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Gauhati University.

Mirzanur Rahman
Assistant Professor, IT
Department



GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

EXTERNAL EXAMINERS CERTIFICATE

This is to certify that “**Bidipta Saikia**”, bearing Roll No “**190102006**” and “**Manikangkan Das**”, bearing Roll No “**190102020**” have delivered their project presentation on “13/06/2023” and I have examined their report entitled “**Assessment Tool**” and recommend this project report as a part for partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Gauhati University.

(External Examiner)



GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date: 13/06/2023

TO WHOM IT MAY CONCERN

This is to certify that “**Bidipta Saikia**” and “**Manikangkan Das**”, bearing Roll No “**190102006**” and “**190102020**” respectively, B.Tech. students of the department of Information Technology, Gauhati University, have submitted the softcopy of their project for undergoing screening through anti-plagiarism software and the similar report found to be xx% (in words).

Mirzanur Rahman
Assistant Professor, IT Department



GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date: 13/06/2023

ACKNOWLEDGEMENT

We would like to express our deep respect and heartfelt gratitude to our advisor and Project Guide, **Mirzanur Rahman**, for his unwavering support and invaluable assistance throughout the project, which led to its successful completion.

We also extend our sincere appreciation and gratitude to **Dr. Shikhar Kumar Sarma**, Head of the Department of Information Technology at Gauhati University, for his generous assistance and cooperation in providing the necessary resources and facilities for our project. His guidance and motivation have been a constant source of inspiration for us.

We would like to convey our thanks to the other faculty members of the Department for their cooperation and generous support in completing the project. Additionally, we are grateful to our friends for their assistance in the successful execution of our project.

Bidipta Saikia
(190102006)

Manikangkan Das
(190102020)

ABSTRACT

Gradr.io is a Software as a Service (SAAS) platform that aims to revolutionize the pre-employment testing landscape for technical, logic, and soft skills. In today's competitive job market, finding the right talent with the necessary skills is a challenge for organizations. Gradr.io offers a comprehensive solution that empowers employers to assess candidates effectively and make informed hiring decisions.

Gradr.io provides a robust testing infrastructure, ensuring accurate evaluation of candidates' technical, logic, and soft skills. Employers can create customized assessments tailored to their specific job requirements, enabling them to evaluate candidates on skills directly relevant to the roles they are being considered for. The platform covers a wide range of technical skills across various domains, facilitating a holistic assessment of candidates' capabilities and suitability for the job.

In the highly competitive landscape of pre-employment testing platforms, Gradr.io stands out by leveraging advanced testing capabilities and customized assessments. It empowers employers to evaluate candidates thoroughly and identify the most qualified individuals. By utilizing Gradr.io's SAAS platform, organizations can optimize their recruitment strategies, improve the efficiency of candidate evaluation, and select the best-fit talent for technical roles.

Gradr.io is reshaping the recruitment landscape by providing a powerful tool that facilitates the identification of candidates with the necessary skills to drive organizational success. With comprehensive skill coverage, tailored assessments, and cutting-edge features, Gradr.io streamlines and enhances the hiring process. Organizations can leverage Gradr.io to stay ahead in the competitive job market and build high-performing teams that drive their success.

List of Contents

I. Abstract.....	(I)
II. List of Contents.....	(II)
III. List of Figures.....	(III)
Chapter 1: Introduction.....	1-7
● 1.1. Intro.....	1
● 1.2. Objective.....	2
● 1.3. Technologies and Tools.....	3
● 1.4. Development platform used.....	4
● 1.5. Features.....	4
● 1.6. Review of Literature.....	6
Chapter 2: System Architecture.....	8-16
● 2.1. Methodology.....	8
● 2.2. Model View Controller Architecture.....	9
● 2.3. System Architecture.....	10
● 2.4. High-level Diagrams.....	13
Chapter 3: Implementation.....	17-18
● 3.1.1. Admin Panel Implementation.....	17
● 3.1.2. Candidate Test App Implementation.....	17
Chapter 4: Results and Discussion.....	19-
● 4.1. Candidate Page.....	19
○ 4.1.1. Pre-flow.....	19
■ 4.1.1.1. Basic details containing Name and LinkedIn profile.....	19
■ 4.1.1.2. Greeting to the platform.....	20
■ 4.1.1.3. Instructions page.....	21
■ 4.1.1.4. Camera access.....	22
■ 4.1.1.5. Start test.....	23
○ 4.1.2. Candidate flow.....	23-27
■ 4.1.2.1. Question answering start.....	23
■ 4.1.2.2. Candidate can answer question using select dropdown modal.....	24

■ 4.1.2.3.Candidate can end the session	25
■ 4.1.2.4. Candidate can report if something unusual happens.....	26
■ 4.1.2.5. Candidate has to go through multiple topic sections.....	27
■ 4.1.2.6. Candidate will be disqualified from the assessment if they attempt to cheat.....	27
○ 4.1.3. Post-flow.....	28-30
■ 4.1.3.1. Result showcase of each section.....	28
■ 4.1.3.2. End assessment page.....	30
● 4.2. Admin Site.....	30-34
○ 4.2.1. Login to admin panel.....	31
○ 4.2.2. Admin Dashboard.....	31
○ 4.2.3. Create a new assessment.....	32
■ 4.2.3.1. Give a test name.....	32
■ 4.2.3.2. Set start time and end time.....	32
■ 4.2.3.3. Pick topics.....	33
■ 4.2.3.4. Add the number of applicants.....	33
○ 4.2.4. Invite candidates.....	34
■ 4.2.4.1. Send invitation.....	34
○ 4.2.5. Analytics page.....	34-38
Chapter 5: Conclusion and Future Works.....	39-40
● 5.1. Conclusion.....	39
● 5.2. Future Works.....	39
References.....	41

List of Figures

• Fig 1: Model View Controller Architecture.....	10
• Fig 2: System Architecture.....	12
• Fig 3: High level diagram.....	14
• Fig 4: State diagram for candidate app.....	15
• Fig 5: Landing candidate page.....	19
• Fig 6: Candidate basic info filled page.....	20
• Fig 7: Candidate greetings page.....	21
• Fig 8: Instruction page.....	22
• Fig 9: Camera access page.....	22
• Fig 10: Test page.....	24
• Fig 11: Test page with select answer.....	25
• Fig 12: Test page with end session modal.....	26
• Fig 13: Test page with feedback modal.....	27
• Fig 14: Test section info page.....	27
• Fig 15: Test marks show page.....	28
• Fig 16: Test submit page.....	29
• Fig 17: Test done page.....	30
• Fig 18: Admin panel login page.....	31
• Fig 19: Admin dashboard.....	31
• Fig 20: Assessment creating page 1.....	32
• Fig 21: Assessment creating page 2.....	32
• Fig 22: Assessment creating page 3.....	33
• Fig 23: Candidate invitation page.....	34
• Fig 24: Analytics page.....	36-38

Chapter One

INTRODUCTION

1.1 Introduction

Gradr.io: Pre-Employment Testing for Technical, Logic, and Soft Skills

In the modern era of recruitment, finding the right talent with the necessary technical, logical, and soft skills can be a daunting task for organizations. To streamline this process, Gradr.io has emerged as a cutting-edge Software as a Service (SAAS) platform. Gradr.io offers a comprehensive solution that allows employers to assess candidates based on specific technical skills required for the role they are looking to fill.

Key Features and Functionality: Gradr.io stands out from the crowd of existing tools by providing a range of advanced features and functionalities. The platform offers:

- **Pre-Employment Testing:** Gradr.io provides a robust testing infrastructure, enabling organizations to evaluate candidates' technical, logic, and soft skills before making hiring decisions.
- **Tailored Assessments:** Employers can customize assessments according to their specific job requirements, ensuring candidates are tested on the skills directly relevant to the role they are being considered for.
- **Comprehensive Skill Coverage:** Gradr.io covers a wide range of technical skills, ensuring that employers can evaluate candidates thoroughly across various domains.

Prominent Competitors: Gradr.io operates in a competitive landscape, with several notable players attempting to solve the same problem for organizations. Some of the prominent competitors include:

- Testgorilla: Recently raising an impressive \$70 million USD in funding, Testgorilla offers a comprehensive platform for pre-employment testing.
- Testdome: Testdome provides a versatile testing platform, allowing organizations to evaluate candidates' skills in various technical domains.
- Canditech: Canditech offers innovative solutions for pre-employment testing, focusing on identifying candidates' potential through comprehensive assessments.
- imocha.io: With its advanced testing capabilities, imocha.io enables organizations to assess candidates' skills effectively, ensuring the right fit for each role.

Gradr.io is the pre-employment testing landscape with its SAAS platform. By providing tailored assessments and comprehensive skill coverage, the platform empowers employers to make informed hiring decisions. In a competitive market where finding the right talent is crucial, Gradr.io stands as a powerful tool to streamline the recruitment process and identify candidates with the necessary technical, logic, and soft skills.

1.2 Objectives:

- Streamline the Candidate Screening Process:
 - The primary objective of implementing Gradr.io is to streamline the candidate screening process in organizations, universities, and companies by providing a comprehensive solution for assessing technical, logic, and soft skills.
- Enhance the Automated Coding Rounds:
 - Addressing the limitations of existing tools, Gradr.io aims to improve the automated coding rounds by incorporating advanced techniques that overcome the challenges posed by language models like ChatGPT. This ensures a more accurate evaluation of candidates' coding skills.
- Assess Practical Concepts Beyond Coding:
 - Gradr.io focuses on evaluating software professionals based not only on their coding proficiency but also on their practical knowledge of concepts such as API standards (e.g., REST, OAuth), GIT, security practices (e.g., OWASP, cryptography), performance optimization, system design, CI/CD, and more.
- Customized Assessments for Job Requirements:
 - One of the key objectives of Gradr.io is to provide organizations with the ability to customize assessments according to specific job requirements. This ensures that candidates are tested on the skills directly relevant to the role they are being considered for.
- Comprehensive Skill Coverage:
 - Gradr.io aims to cover a wide range of technical skills, ensuring that the assessment process is comprehensive and thorough. By evaluating candidates across multiple domains, it enables organizations to identify individuals with a well-rounded skill set.
- Efficient Screening of a High Volume of Candidates:
 - With the goal of screening hundreds of candidates every month, Gradr.io focuses on providing an efficient and scalable screening process. By automating various assessment rounds, it helps organizations save time and effort while ensuring accurate evaluations.
- Evaluation of Practical Application of Knowledge:
 - Beyond mere theoretical knowledge, Gradr.io emphasizes assessing candidates on their ability to apply practical concepts. This ensures that candidates possess the necessary skills to excel in real-world scenarios.
- Confidence in Hiring Decisions:
 - Gradr.io aims to instill confidence in the hiring decisions made by organizations. By providing a reliable and comprehensive assessment platform, it helps employers identify the most suitable candidates for their technical roles.
- Bridge the Gap Between Technical and Soft Skills:
 - In addition to technical expertise, Gradr.io recognizes the importance of evaluating candidates' soft skills. The platform aims to bridge the gap between technical and soft skills assessment, enabling organizations to find candidates who possess a well-rounded skill set.
- Continuous Improvement and Adaptability:

- Gradr.io is committed to continuously improving its platform based on feedback and evolving industry requirements. The objective is to ensure that the assessment process remains up-to-date, aligning with the latest industry trends and technological advancements.

These objectives collectively drive Gradr.io's mission to revolutionize the candidate screening process and provide organizations with a reliable and efficient platform for assessing technical, logic, and soft skills.

1.3 Technologies and Tools:

- **React.js:** React.js is a popular JavaScript library for building user interfaces. It provides a component-based approach, making it easier to develop reusable UI elements. React.js is known for its efficient rendering and virtual DOM, resulting in faster and smoother user experiences.
- **TypeScript:** TypeScript is a typed superset of JavaScript that adds static type checking and additional language features to JavaScript. It helps catch errors during development and improves code maintainability by providing enhanced IDE support and code refactoring capabilities.
- **Tailwind CSS:** Tailwind CSS is a utility-first CSS framework that provides a set of predefined classes to style UI elements. It allows developers to rapidly build user interfaces by composing classes instead of writing custom CSS. Tailwind CSS offers flexibility and customization options, making it suitable for creating responsive and visually appealing designs.
- **Strapi:** Strapi is an open-source headless CMS (Content Management System) that provides a flexible and customizable way to manage content. It allows developers to create and manage APIs easily, making it an ideal choice for building scalable and content-driven applications.
- **Figma:** Figma is a cloud-based design and prototyping tool that enables collaborative interface design. It allows designers and developers to work together in real-time, creating interactive and visually appealing user interfaces for web and mobile applications.
- **Node.js:** Node.js is a JavaScript runtime built on Chrome's V8 engine. It allows developers to execute JavaScript code on the server-side, enabling server-side scripting and building scalable network applications. Node.js is known for its event-driven, non-blocking I/O model, which makes it efficient and well-suited for building real-time applications.
- **Nest.js:** Nest.js is a progressive Node.js framework for building efficient, scalable, and maintainable server-side applications. It utilizes TypeScript and follows the modular and dependency injection principles, providing a solid foundation for developing complex backend systems. Nest.js combines the power of Node.js with architectural patterns inspired by Angular, making it a suitable choice for building robust and scalable server-side components in the project. With its strong focus on modularity, testability, and extensibility, Nest.js enhances code organization and promotes the development of highly maintainable and scalable applications.

- These technologies were carefully chosen to build a robust and modern tech stack for the project, ensuring efficient development, seamless collaboration, and optimal performance of the application.
- **VSCODE:** Visual Studio Code (VS Code) is a lightweight and powerful source code editor developed by Microsoft. It provides a range of features and extensions that enhance productivity and facilitate efficient coding.

1.4 Development Platform used:

- MACBOOK PRO M1:
 - RAM: 16 GB
 - ROM: 256 GB
- MACKBOOK PRO M2:
 - RAM: 32 GB
 - ROM: 512 GB

1.5 Features

1.5.1 How is Gradr.io the same?

- **Easily setup screening assessments:** Gradr.io shares a similarity with other tools in that it allows for the easy setup of screening assessments. Employers can create and customize assessments according to their specific requirements, ensuring that candidates are evaluated on the skills relevant to the desired role.
- **Anti-cheating mechanism (although a lot better):** Similar to other platforms, Gradr.io incorporates anti-cheating mechanisms to maintain the integrity of the assessment process. While the exact details of Gradr.io's anti-cheating measures may differ, the objective is to prevent dishonest practices and ensure fair evaluations.
- **Bias-free recruitment:** Gradr.io, like other tools, aims to facilitate bias-free recruitment by providing standardized assessments. By utilizing objective evaluation criteria, it helps mitigate unconscious biases and ensures that candidates are evaluated solely based on their skills and qualifications.
- **Great candidate experience:** Gradr.io prioritizes delivering a great candidate experience throughout the assessment process. Similar to other platforms, it focuses on providing a user-friendly interface, clear instructions, and a seamless testing experience, allowing candidates to perform their best and showcase their abilities.
- **Great experience for the hiring team. Clean and intuitive interface:** Gradr.io, like its competitors, offers a clean and intuitive interface for the hiring team. This enables recruiters and hiring managers to navigate the platform effortlessly, manage assessments effectively, and access candidate results conveniently, enhancing their overall experience and efficiency in the recruitment process.
- While Gradr.io shares these similarities with other tools, it may differentiate itself through specific features, customization options, or unique user experiences. These factors

contribute to making Gradr.io a standout choice in the landscape of pre-employment testing platforms.

1.5.2 How is Gradr.io different?

- **Largest question pool with the help of AI:** Gradr.io sets itself apart by leveraging AI technology to create the largest question pool available. This extensive pool ensures a diverse range of assessment questions, reducing the risk of repetition and providing a comprehensive evaluation of candidates' skills.
- **Inherently cheating-proof mechanism:** Gradr.io incorporates an inherently cheating-proof mechanism, surpassing existing solutions. With advanced techniques and intelligent algorithms, it detects and prevents cheating attempts during assessments, ensuring the integrity and accuracy of the evaluation process.
- **Global and local comparisons in skills:** Gradr.io goes beyond evaluating candidates in isolation. It enables global and local skill comparisons by providing benchmarking data and industry insights. Employers gain a holistic view of candidates' abilities, comparing them not only against their peers but also against a larger talent pool.
- **GRE-like scoring. Better questioning mechanism:** Gradr.io introduces a GRE-like scoring system, elevating the assessment process. This scoring method assesses candidates' skills comprehensively, allowing employers to make more informed decisions. Moreover, Gradr.io incorporates a refined questioning mechanism that enhances the precision and relevance of assessment items.
- **Candidate and Role-specific tests:** Recognizing the unique requirements of different roles, Gradr.io offers candidate and role-specific tests. Employers can tailor assessments to focus on the specific skills and competencies relevant to each job position, ensuring that candidates are evaluated based on the exact qualifications needed for success in that role.
- These distinctive features set Gradr.io apart from other pre-employment testing platforms, making it a powerful and comprehensive tool for organizations seeking efficient, accurate, and tailored assessments of candidate skills.

1.5.3 Inherently cheating-proof mechanism

- **Webcam Verification:** Gradr.io utilizes webcam verification to ensure that candidates are not receiving assistance from anyone else during the assessment. It helps maintain the integrity of the evaluation process.
- **IP Check:** By performing IP checks, Gradr.io verifies the uniqueness and authenticity of candidates' connections, preventing multiple attempts or unauthorized access.
- **Copy-Paste Detection:** Although not always necessary, Gradr.io incorporates copy-paste detection to identify any potential instances of plagiarism or cheating.
- **Fullscreen Check:** Gradr.io conducts fullscreen checks to ensure that candidates are not accessing external resources or applications during the assessment, promoting a fair evaluation environment.
- **Mouse Pointer Analysis:** Gradr.io analyzes the mouse pointer movements to detect any suspicious patterns or irregularities that may indicate unauthorized assistance.
- **Differentiating Factor:** Unique Questioning Mechanism

- **Unlimited Questions in 10 Minutes:** Gradr.io sets itself apart by allowing candidates to answer as many questions as they can within a specified time frame (e.g., 10 minutes). This approach encourages candidates to demonstrate their knowledge and skills across a broad range of topics.
- **Incorporating Time Constraint for Scoring:** Gradr.io introduces a unique scoring system based on the time constraint. If candidates resort to using external resources, such as Googling or relying on ChatGPT, they will lose time, ultimately resulting in a lower score. This mechanism promotes genuine understanding and discourages cheating.

1.5.4 Global and Local Skill Comparisons

Gradr.io goes beyond assessing candidates solely based on their individual skills. It provides a unique feature of global and local skill comparisons, enabling employers to gain valuable insights into candidates' relative proficiency in specific areas.

- **Global Talent Pool Comparison:** Gradr.io compares candidates' skills to a global pool of talent. This comparison allows employers to identify candidates who stand out as top performers, showcasing their exceptional abilities in the assessment.
- **Individual Skill Proficiency Comparison:** Gradr.io provides insights into candidates' relative strengths and weaknesses in specific skills. It helps employers determine candidates who excel in certain areas, such as being proficient in Python but relatively weaker in HTML. This information allows for better alignment of candidates' skills with the requirements of the role.
- **Benchmarking Against Worldwide Skill Levels:** Gradr.io's unique feature allows employers to benchmark candidates' skills against worldwide skill levels. For example, it can identify a candidate who is average in GIT skills but ranks among the top 99% globally in security concepts. This information gives employers a comprehensive understanding of candidates' skill levels and their comparative standing on a global scale.

The answers obtained from the assessment on Gradr.io can provide insights into the top-performing candidates, their strengths in specific skills, and their comparative skill levels within a global talent pool. This feature empowers employers to make informed decisions and identify candidates who possess exceptional abilities and stand out in their respective areas of expertise.

1.6 Review of Literature

The development of the Gradr.io platform builds upon existing research and literature surrounding pre-employment testing, technical skills assessment, and software development methodologies. A review of relevant literature reveals several key insights and contributions from previous studies:

Pre-employment Testing: Research in the field of pre-employment testing highlights the importance of assessing candidates' technical, logic, and soft skills to make informed hiring decisions. Various studies emphasize the effectiveness of standardized tests and tailored assessments in predicting job performance and identifying candidates who are the right fit for specific roles within organizations.

Technical Skills Assessment: The evaluation of technical skills plays a critical role in determining a candidate's proficiency and suitability for a given job. Literature emphasizes the need for comprehensive skill coverage and customizable assessments that align with the specific technical requirements of different roles. The integration of advanced testing infrastructure and reliable evaluation methodologies is crucial to accurately assess candidates' technical competencies.

Software Development Methodologies: Software development methodologies provide structured frameworks for the systematic development of software products. The Waterfall Model, in particular, has been widely adopted in projects with well-defined requirements. Previous research highlights the benefits of the Waterfall Model's sequential approach, such as its ability to facilitate thorough documentation, effective communication, and rigorous testing.

The development of the Gradr.io platform incorporates insights from the existing literature to address the challenges faced by organizations in the recruitment process. By leveraging previous research on pre-employment testing, technical skills assessment, and software development methodologies, this project aims to contribute to the body of knowledge and provide a robust solution for employers seeking to identify candidates with the necessary technical, logic, and soft skills.

By building upon the existing literature and utilizing the Waterfall Model as the guiding methodology, this project endeavors to enhance the recruitment process by providing a comprehensive SaaS platform that empowers organizations to make informed hiring decisions based on objective assessments of candidates' technical competencies and soft skills.

Chapter Two

SYSTEM ARCHITECTURE

2.1 Methodology

The Waterfall Model serves as the chosen software development methodology for the development of the Gradr.io platform. This traditional and sequential approach provides a structured framework for the systematic development of the software, ensuring a well-defined process from inception to deployment.

- **Requirements Gathering:** The first phase of the Waterfall Model involves gathering and documenting the project requirements in detail. This includes identifying the key features and functionalities of the Gradr.io platform, understanding the specific technical, logic, and soft skills to be assessed, and defining the scope of the project. Requirements are collected through extensive discussions with stakeholders, domain experts, and potential users of the platform.
- **System Design:** Once the requirements are established, the system design phase begins. This involves translating the requirements into a comprehensive system design, including the architecture, data models, user interfaces, and interactions. The design phase focuses on creating a blueprint for the Gradr.io platform, ensuring that all aspects of the system are well-defined and aligned with the project objectives.
- **Implementation:** After the system design is finalized, the implementation phase begins. This phase involves coding and developing the Gradr.io platform according to the design specifications. Software developers and engineers collaborate to write the code, integrate necessary components, and ensure the functionality of the platform. This phase follows a sequential approach, with each component being implemented one after the other.
- **Testing and Quality Assurance:** Once the implementation is complete, the platform undergoes rigorous testing and quality assurance to identify and rectify any defects or issues. Testing includes functional testing, performance testing, and user acceptance testing to ensure that the Gradr.io platform meets the specified requirements and performs as expected. Any identified issues are resolved before proceeding to the next phase.
- **Deployment and Operations:** Once the testing phase is successfully completed, the Gradr.io platform is deployed to a production environment. This involves configuring servers, setting up databases, and ensuring the platform's readiness for usage. Ongoing operations and maintenance activities are also defined in this phase to ensure the smooth functioning of the platform after deployment.

The Waterfall Model provides a structured and well-defined approach to the development of the Gradr.io platform, ensuring that each phase is completed before moving on to the next. This methodology promotes thorough documentation, effective communication, and risk management throughout the development process. It allows for clear project planning, resource allocation, and milestone tracking, enabling the project team to adhere to timelines and deliver a high-quality platform.

By adopting the Waterfall Model, the development of the Gradr.io platform benefits from a systematic approach, ensuring that requirements are accurately captured, design is comprehensive, implementation is robust, and testing is thorough. This methodology provides a strong foundation for the successful development and deployment of the platform, meeting the needs and expectations of employers seeking to assess candidates' technical, logic, and soft skills accurately.

2.2 System Architecture

The system architecture of the Gradr.io assessment platform is designed to ensure seamless functionality, efficient data flow, and secure interactions between different components. It comprises several key elements that work together to deliver a robust and reliable experience for users.

2.2.1 MVC Pattern + React TypeScript for the Project

For the project, we have implemented the MVC (Model-View-Controller) pattern along with React and TypeScript to handle the view layer of the application. This combination offers a robust and scalable approach to organizing and managing the user interface.

Here's how MVC pattern, React, and TypeScript contribute to the project:

- **Model:** The Model represents the data and business logic of the application. It encapsulates the data structures and functions required to manage and manipulate the application's state. By following the MVC pattern, we ensure that the data and logic are separated from the view, promoting better code organization and maintainability.
- **View:** React, along with TypeScript, handles the view layer of the application. React is a popular JavaScript library for building user interfaces, while TypeScript adds static typing and other advanced features to enhance the development experience. With React and TypeScript, we can create reusable and modular UI components that provide an interactive and responsive user experience.
- **Controller:** The Controller acts as the intermediary between the Model and the View. It receives input from the user or other external sources, processes the data, and updates the Model accordingly. In our project, the Controller component orchestrates the flow of data and events between the Model and the View, ensuring smooth communication and synchronization.

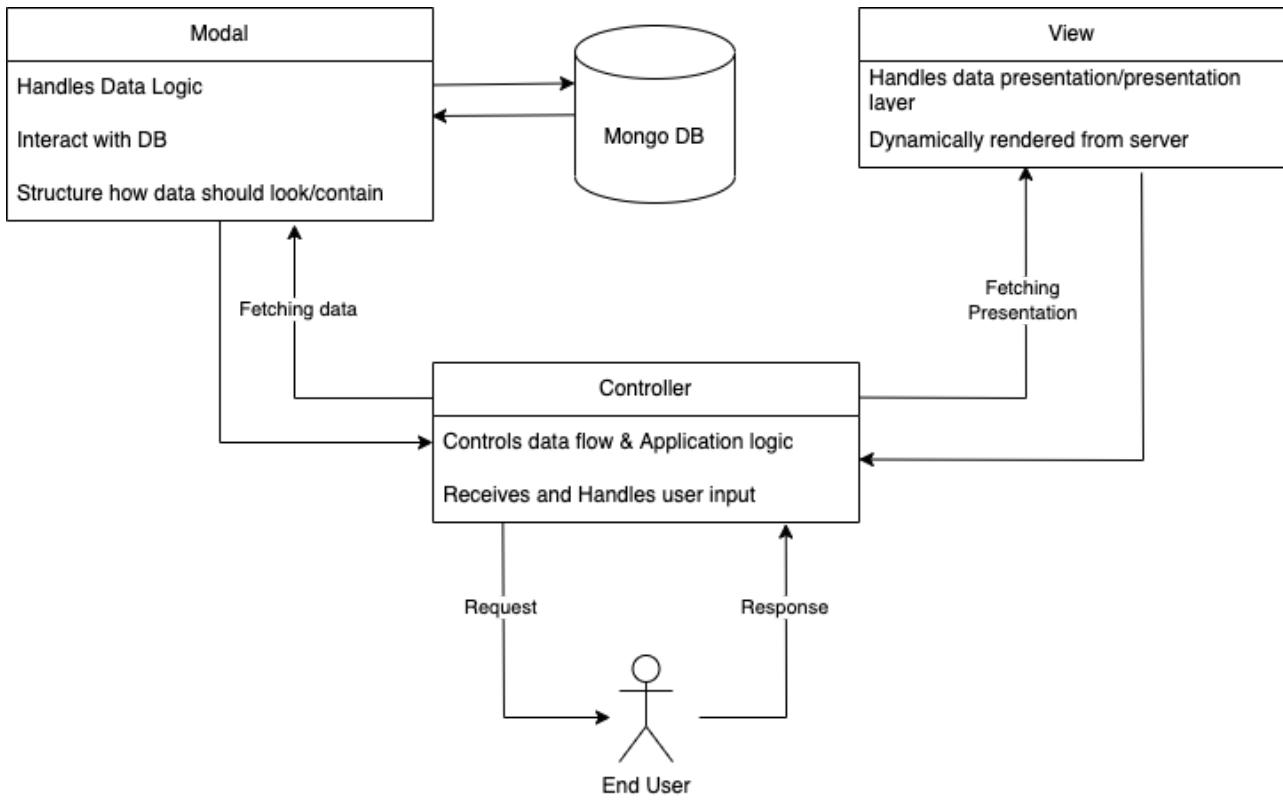


Fig 1: Model View Controller Architecture

By leveraging the MVC pattern, React, and TypeScript in our project, we achieve the following benefits:

- Separation of concerns: The MVC pattern separates the data, logic, and presentation aspects of the application, promoting a modular and maintainable codebase.
 - Reusability: React's component-based architecture allows us to create reusable UI components, reducing code duplication and increasing development efficiency.
 - Type safety: TypeScript provides static typing, allowing us to catch potential errors during the development phase and improve code robustness.
 - Scalability: The combination of MVC, React, and TypeScript enables us to handle complex UI requirements and scale the application as needed.

Overall, the use of the MVC pattern along with React and TypeScript empowers us to build a well-structured, responsive, and scalable user interface for the project, ensuring a high-quality user experience.

2.2.2 System Architecture

In our project, we have utilized a combination of Netlify, AWS (Amazon Web Services), Git, and Azure DevOps to enable continuous integration and deployment processes. This powerful toolset

ensures efficient development workflows and seamless delivery of updates to the production environment.

Here's how each component contributes to our continuous integration and deployment strategy:

- **Netlify:** Netlify is a popular hosting and deployment platform that simplifies the process of building, deploying, and managing web applications. With Netlify, we can easily configure and automate the deployment of our application to the web. It provides features such as automatic build triggers, continuous deployment, and custom domain management, streamlining the deployment process.
- **AWS:** Amazon Web Services offers a wide range of cloud-based services that we leverage for various aspects of our project. AWS provides scalable and reliable infrastructure services, including storage, compute power, and database management. We can utilize AWS services such as Amazon S3 for file storage, AWS Lambda for serverless computing, and Amazon RDS for database management, enhancing the performance and scalability of our application.
- **Git:** Git is a distributed version control system that enables efficient collaboration among developers and facilitates code management. With Git, we can track changes to our codebase, create branches for different features or bug fixes, and merge changes seamlessly. Git's branching and merging capabilities allow us to work on multiple features simultaneously while maintaining code integrity.
- **Azure DevOps:** Azure DevOps is a comprehensive set of development tools and services that support the entire development lifecycle. It provides features for project planning, source control, continuous integration, and continuous deployment. With Azure DevOps, we can automate the build and deployment processes, configure pipelines for different environments, and ensure smooth integration with our code repository.

By integrating Netlify, AWS, Git, and Azure DevOps, we achieve the following benefits:

- **Streamlined development workflows:** The combination of these tools enables us to establish efficient development workflows, ensuring seamless collaboration, version control, and automated deployment processes.
- **Continuous integration:** With Azure DevOps, we can set up continuous integration pipelines that automatically build and test our application whenever changes are pushed to the code repository. This helps identify issues early and maintain code quality.
- **Continuous deployment:** Netlify and AWS integrate seamlessly with Azure DevOps pipelines, allowing us to automate the deployment process. This ensures that changes are deployed to the production environment quickly and reliably.
- **Scalability and reliability:** AWS provides scalable infrastructure services that can accommodate increasing traffic and demand. With the combination of Netlify, AWS, and Azure DevOps, we can ensure the availability and performance of our application.

Overall, the integration of Netlify, AWS, Git, and Azure DevOps empowers us to establish robust continuous integration and deployment processes. This allows us to deliver updates to our application efficiently, maintain code quality, and ensure a seamless experience for our users.

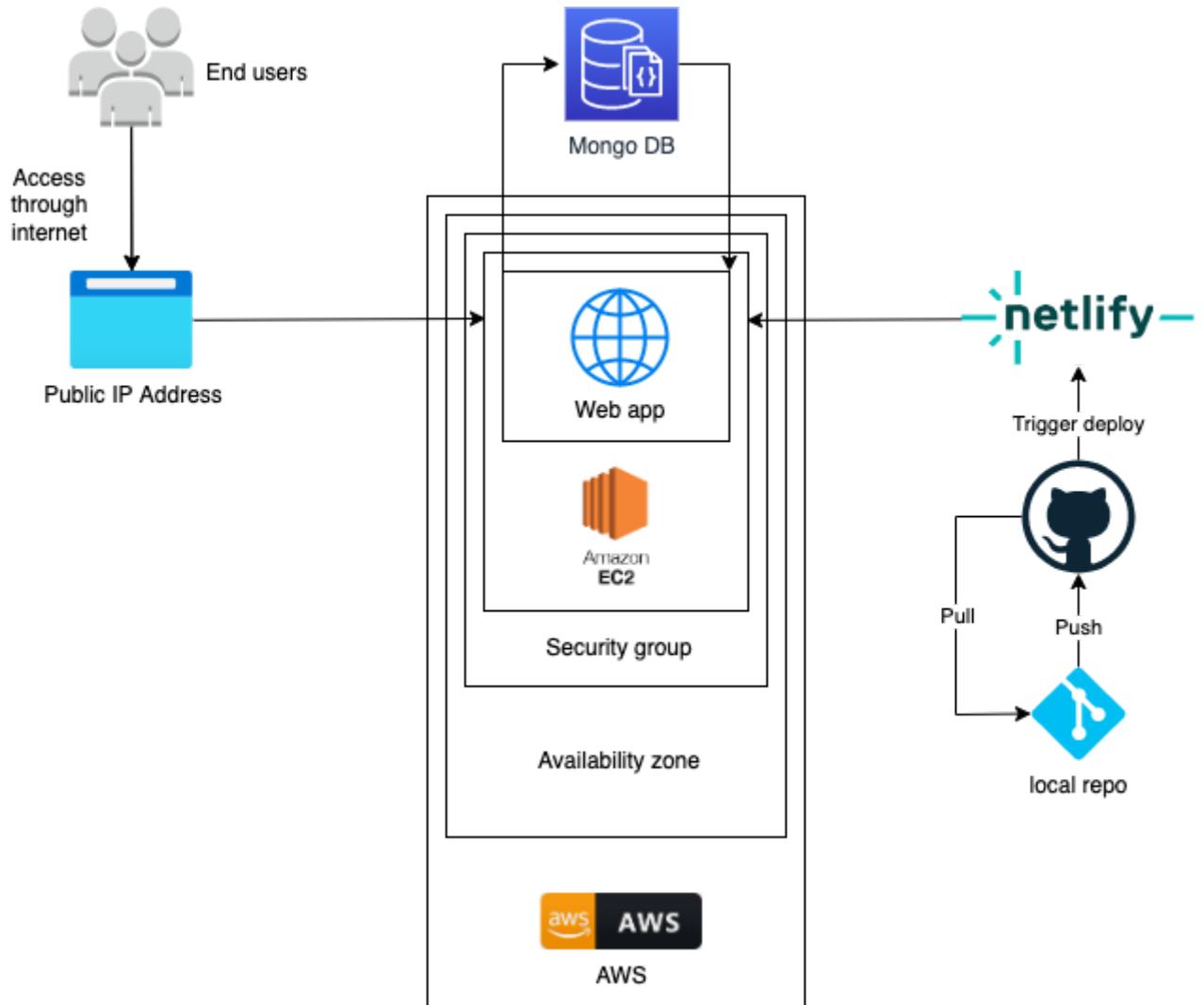


Fig 2: System Architecture

Front-end Components:

- Applicant App: This component provides the interface for candidates to take assessments. It includes screens for information display, webcam access, test questions, and transitions between assessment sections.
- Org Admin Portal/App: This component serves as the management interface for organization administrators. It allows them to log in, create assessments, view assessment results, and access candidate score details.
- Super Admin Interface: This component provides the necessary tools for super admins to manually create organizations, admin accounts, and set passwords.

Back-end Components:

- Authentication and Authorization: This component handles user authentication and authorization processes, ensuring secure access to different system functionalities based on user roles (Super Admin, Org Admin, Applicant).

- Database: The data model is implemented using a database to store essential entities such as Tenants, Products, Organizations, Users, and Audit records. The database ensures data integrity and persistence.
- API Layer: This component serves as the communication interface between the front-end components and the database. It handles requests from the front-end, retrieves and manipulates data from the database, and sends responses back to the client-side applications.
- Business Logic: This component contains the core logic of the system, including the creation of assessments, calculation of scores, and processing of user actions. It ensures consistent and accurate behavior across the platform.
- Integration: The system architecture allows for integration with external services, such as email services for sending invitations and notifications to candidates and admins.

Data Flow:

- User interactions and data inputs flow from the front-end components (Applicant App, Org Admin Portal/App) to the API layer, where requests are processed.
- The API layer communicates with the business logic component to perform the necessary operations, such as creating assessments, managing user accounts, and generating result summaries.
- The business logic component interacts with the database to retrieve or update data as required.
- Result data is stored in the database and made available for viewing by Org Admins through the Org Admin Portal/App.
- Notifications and invitations are sent to candidates and admins via integrated email services.

The system architecture follows a client-server model, with the front-end components serving as the clients and the back-end components providing the server-side functionality. This architecture ensures scalability, flexibility, and modularity, allowing for easy maintenance and future enhancements.

By adopting this system architecture, the Gradr.io platform ensures a well-structured, secure, and efficient environment for conducting assessments, facilitating seamless collaboration between different user roles, and delivering a streamlined user experience.

2.2.3 Data Flow Diagram

Below, we present diagrams illustrating the comprehensive data flow and functionality of our assessment tool. These diagrams include:

2.2.3.1 High-Level Architecture of the App

The high-level architecture diagram presents an overview of the app's architecture, highlighting the key components and their interactions. It demonstrates how information flows within the system and illustrates the various processes involved in the effective functioning of the assessment tool.

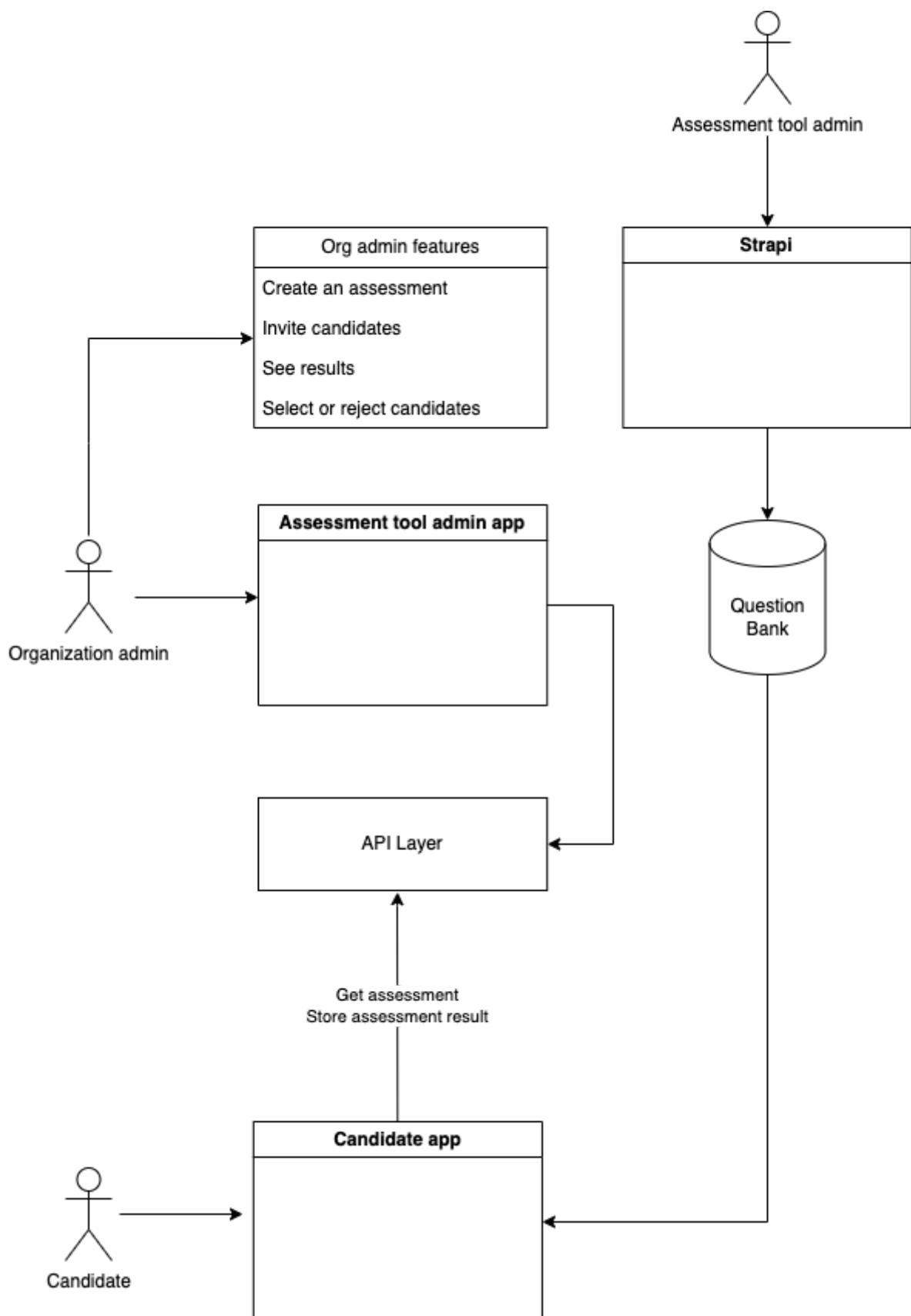
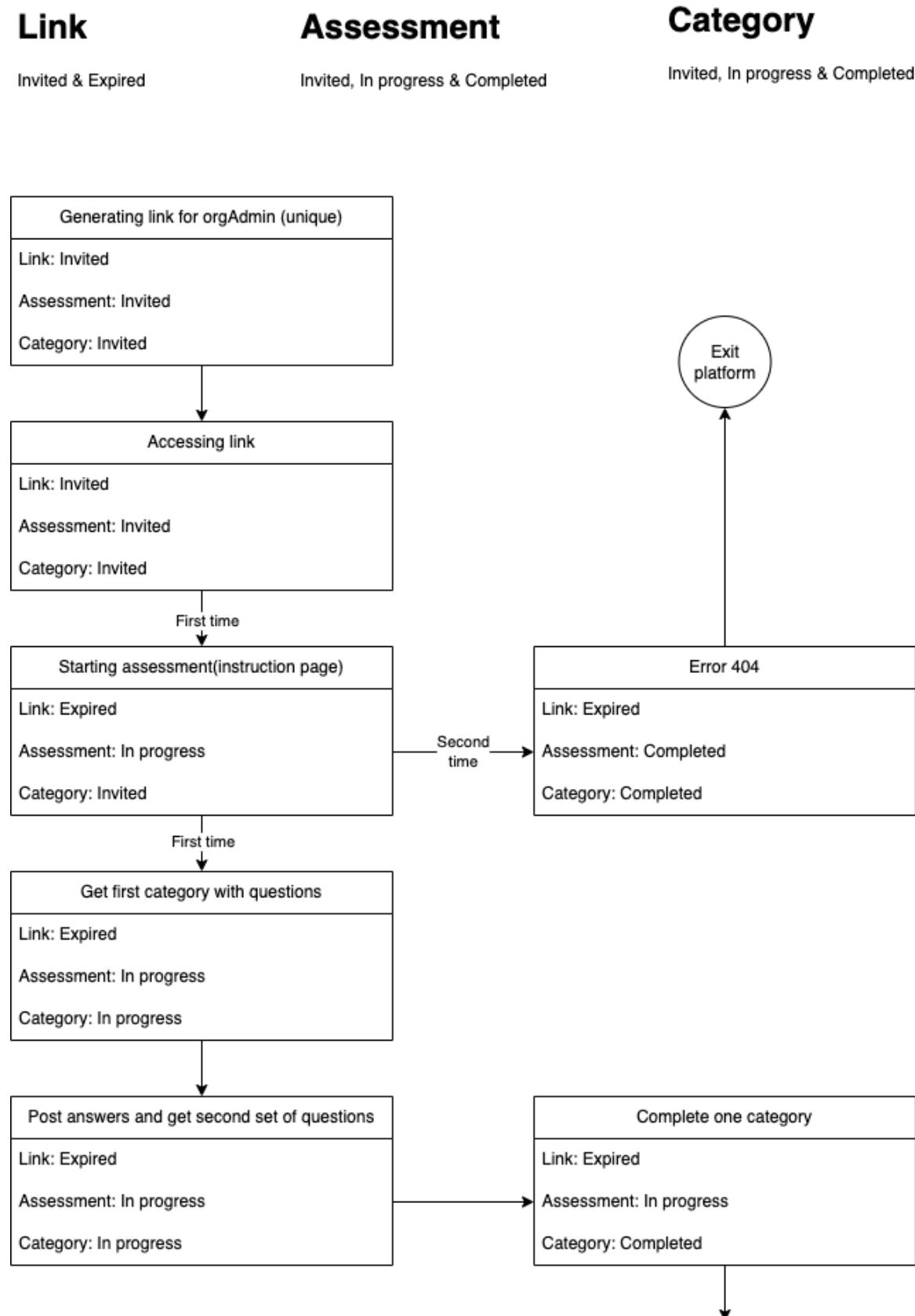


Fig 3 - High level diagram

2.2.3.2 State Diagram for Candidate App

The state diagram showcases the different states and transitions that the candidate app goes through during the assessment process. It provides a visual representation of how candidates interact with the app and navigate through different stages of the assessment.



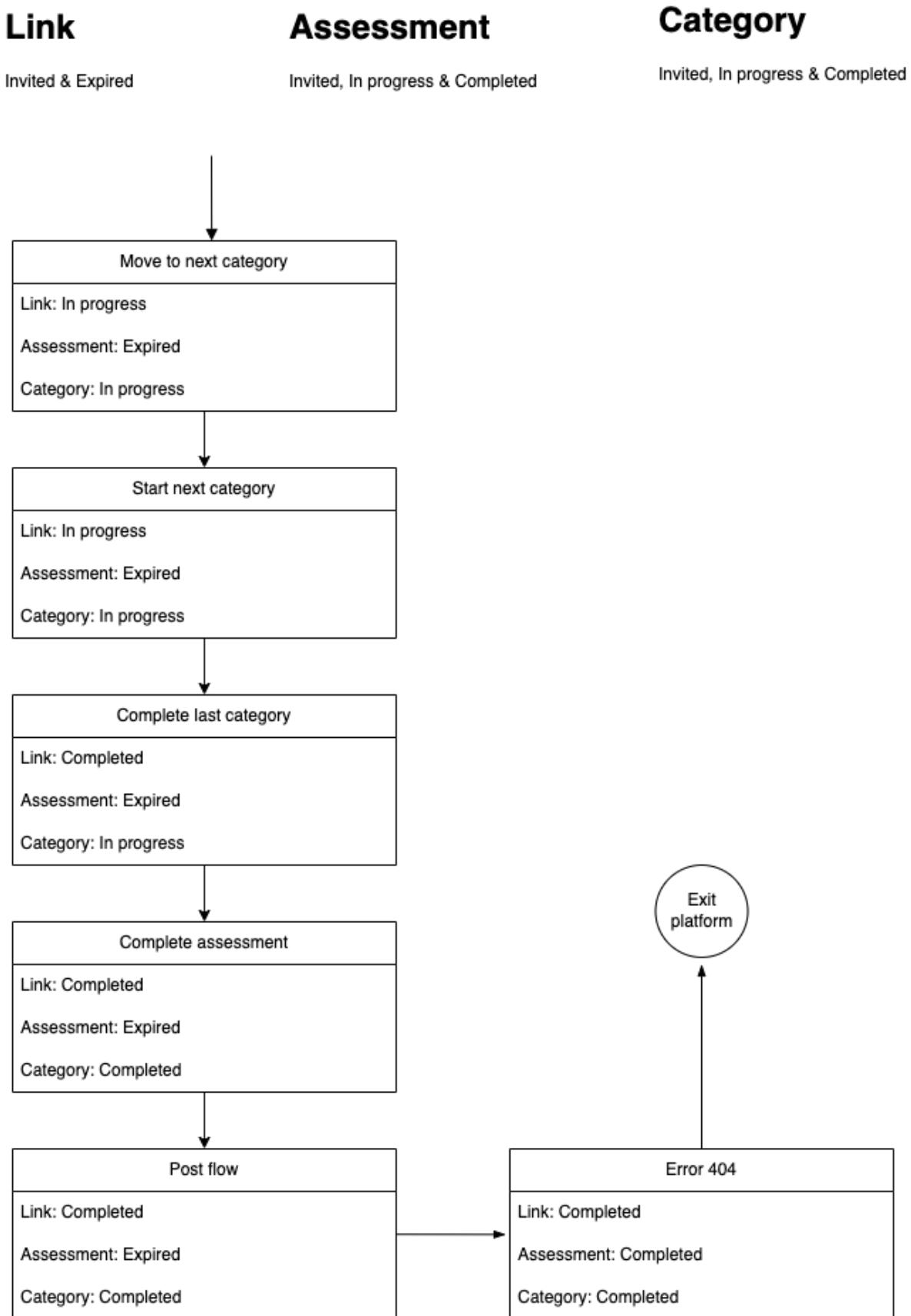


Fig 4: State diagram for candidate app

Chapter Three

Implementation

3.1 Candidate App Implementation

The candidate app is a crucial component of Gradr.io, providing candidates with a user-friendly interface to participate in assessments. The development of the candidate app involves creating different sections, such as the pre-flow, candidate flow, and post-flow.

3.1.1 Pre-flow

The pre-flow section of the candidate app includes greetings and instructions provided to the candidate before starting the assessment. It welcomes candidates, provides necessary guidelines, and sets expectations for the assessment process.

3.1.2 Candidate Flow

The candidate flow section allows candidates to engage in the assessment process. They can answer questions from multiple sections, navigate through different topics, and report any technical issues or crashes encountered during the assessment. The candidate flow section also provides the option to end the session when the assessment is completed.

3.1.3 Post-flow

After completing the assessment, candidates enter the post-flow section. Here, they receive feedback on their performance and can view their assessment results. The post-flow section may also include the option to provide additional comments or feedback on the assessment experience.

3.2 Admin Panel Implementation

The admin panel serves as a control center for administrators to manage assessments effectively. It provides functionalities for creating tests, inviting candidates, accessing analytics, and reviewing individual candidate reports.

3.2.1 Test Creation

Administrators can create customized tests using the admin panel. They can define the sections, questions, and grading criteria based on the specific technical skills and job requirements.

3.2.2 Candidate Invitation

The admin panel allows administrators to invite candidates to participate in assessments. Administrators can send personalized invitations with assessment details and instructions to selected candidates via email or other communication channels.

3.2.3 Analytics and Reporting

The admin panel provides comprehensive analytics and reporting capabilities. Administrators can access real-time data on candidate performance, overall assessment results, and individual reports. This information enables data-driven decision-making in the hiring process.

Chapter Four

RESULTS AND DISCUSSION

4.1 Candidate Page

4.1.1 Pre flow

Description: The Candidate Page serves as the initial point of interaction for candidates accessing the Gradr.io assessment tool. It consists of three flows, and the first flow is the "Pre-Flow First Screen." Candidates are required to provide their name and optional profile information to proceed to the next page.

Access Restriction: It's important to note that the Candidate Page can only be accessed through a unique email invitation sent by the HR department or an organization member. This access restriction ensures that the page remains exclusive to each candidate and maintains the integrity of the assessment process.

By implementing this access restriction, Gradr.io ensures that only authorized individuals can participate in the assessment, maintaining confidentiality and security throughout the evaluation process.

4.1.1.1 Basic details containing Name and LinkedIn profile

The screenshot shows a landing candidate page with a light gray background. In the center, there is a large, faint watermark-like graphic of a network of lines forming a star or web pattern. At the top left, there is a small, partially visible logo consisting of a lowercase 't' followed by a period. Below the logo, there are two input fields stacked vertically. The top field has the placeholder text "Enter your name here*". The bottom field has the placeholder text "Enter your linkedin profile url". At the bottom left of the page, there is a dark blue rounded rectangular button with the word "Next" written in white capital letters.

Fig 5: Landing candidate page

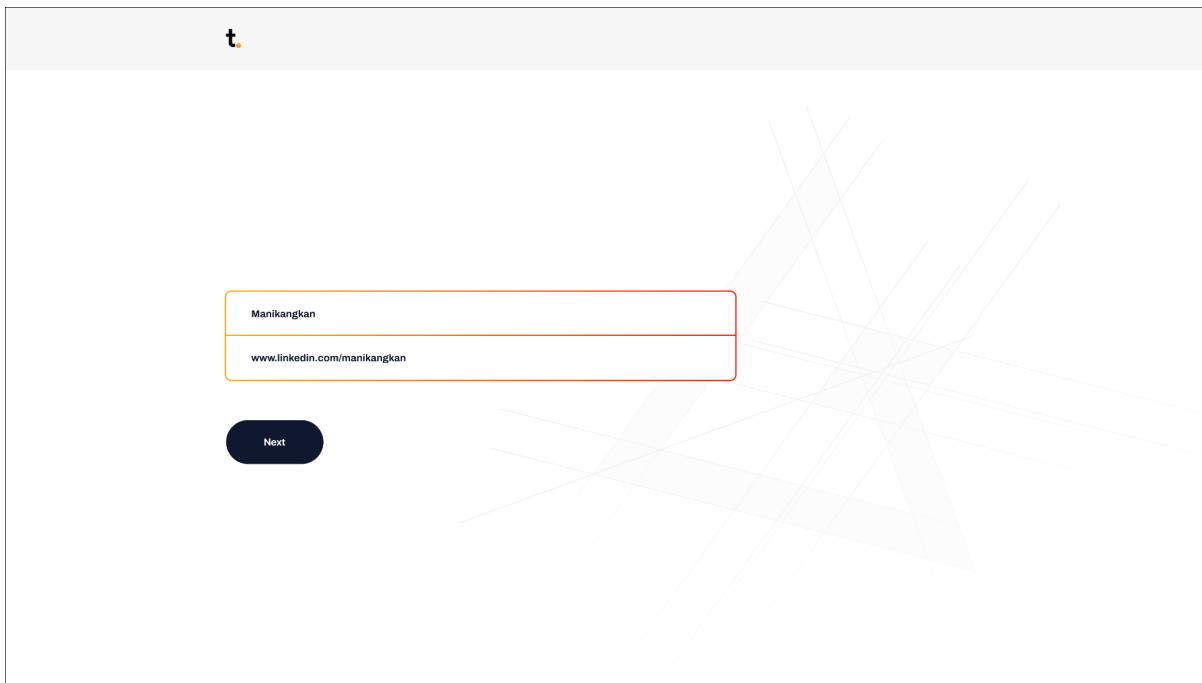


Fig 6: Candidate basic info filled page

4.1.1.2 Greeting Page

Description: The Greeting Page is the second stage of the Candidate Page flow in Gradr.io. After candidates have provided their name and optional social professional profile information on the previous page, they are redirected to the Greeting Page. This page serves as a warm welcome and provides a personalized experience for the candidate.

Page Content: Upon reaching the Greeting Page, candidates are greeted with a friendly message, such as "Welcome, [Candidate Name]!" The page may also display a brief introduction or overview of the assessment process to provide candidates with a clear understanding of what to expect.

Purpose: The Greeting Page aims to create a positive and engaging environment for candidates. By addressing them by their name and providing a warm welcome, Gradr.io seeks to establish a friendly rapport and enhance the candidate's overall experience. Additionally, the page sets the tone for the assessment process, helping candidates feel more comfortable and confident.

Overall, the Greeting Page plays a vital role in ensuring candidates have a smooth and personalized onboarding experience before proceeding to the main assessment tasks.

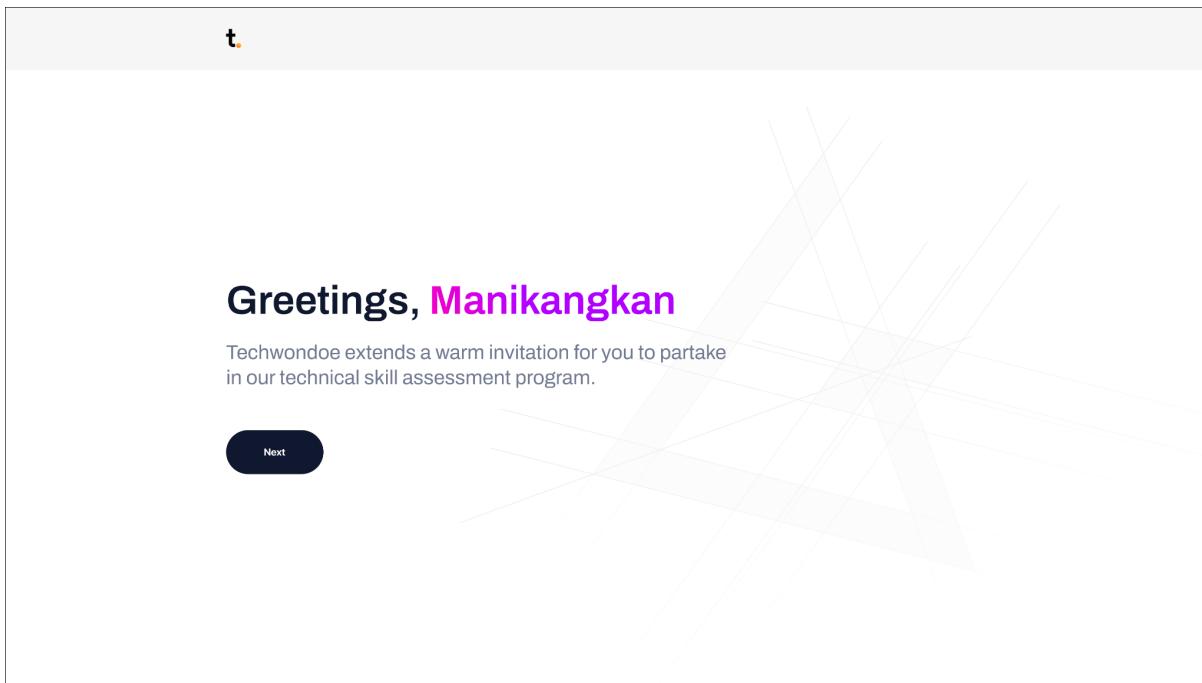


Fig 7: Candidate greetings page

4.1.1.3 Page Name: Instruction Page

Description: The Instruction Page is the third stage of the Candidate Page flow in Gradr.io. After being greeted on the Greeting Page, candidates are directed to the Instruction Page. This page provides candidates with detailed information about the test process and prepares them for the upcoming assessment tasks.

Page Content: Upon reaching the Instruction Page, candidates are presented with a comprehensive set of instructions. These instructions include:

- Test Overview: A brief introduction explaining the purpose and structure of the assessment.
- Task Description: Clear instructions on the specific tasks the candidates need to perform during the assessment.
- Time Limit: Information regarding the time allocated for completing the assessment tasks.
- Technical Requirements: Any technical specifications or requirements, such as enabling the webcam or microphone.
- Guidelines: Guidelines on how to approach the assessment, including tips for success and any specific rules or expectations.

Additionally, the page may include visual aids such as diagrams or images to illustrate the instructions effectively.

Purpose: The Instruction Page serves as a crucial component in ensuring that candidates have a clear understanding of the test process and what is expected of them. By providing detailed instructions, Gradr.io aims to minimize any confusion or uncertainty candidates may have and create a level playing field for all participants. Moreover, the page helps candidates familiarize themselves with the assessment format and guidelines, enabling them to perform at their best.

After reading and understanding the instructions, candidates can proceed to access their webcam, as prompted by the system, to start the test. By accessing the webcam, candidates allow Gradr.io to monitor their behavior during the assessment, ensuring integrity and authenticity throughout the process.

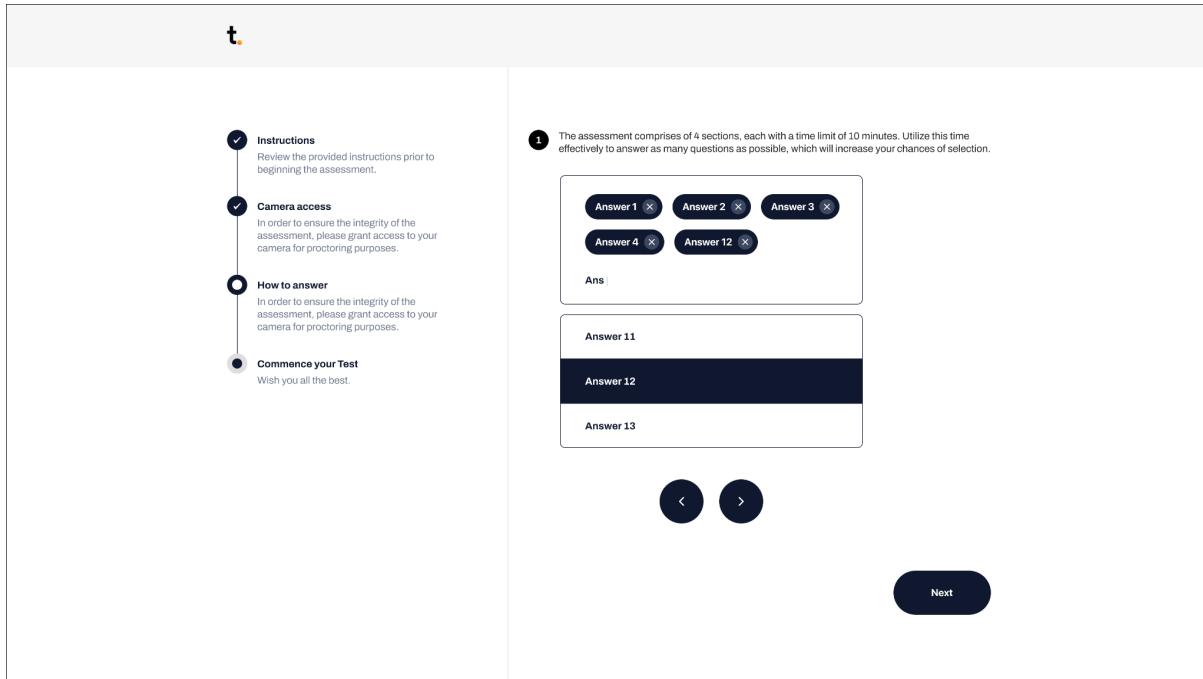


Fig 8: Instruction page

4.1.1.3 Camera access

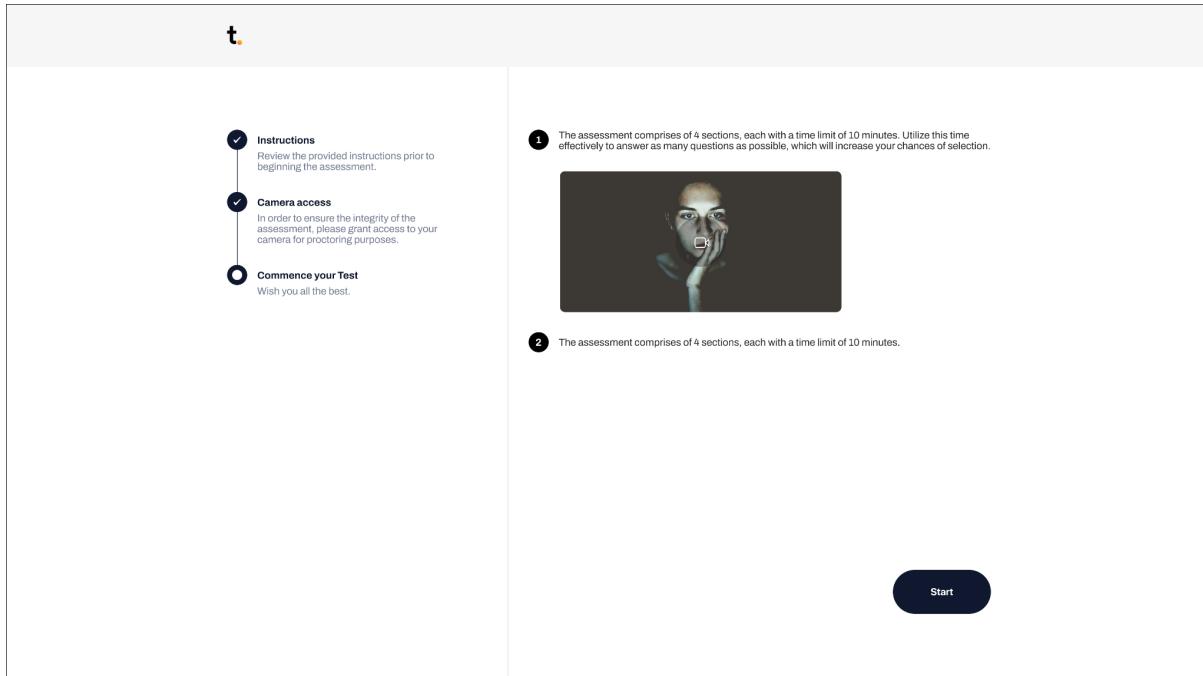


Fig 9: Camera access page

4.1.2 Candidate flow

4.1.2.1 Question answering start

Description: The Test Page is the main stage where candidates engage in the assessment process within Gradr.io. After going through the Instruction Page, candidates proceed to the Test Page, where they encounter the assessment questions and tools to provide their answers.

Page Layout: The Test Page is designed with a user-friendly interface to enhance the candidate's experience. It consists of the following elements:

- **Question Panel (Left Side):** This panel displays the assessment questions, one at a time, allowing candidates to read and understand the content.
- **Answer Select Dropdown:** Located below each question in the Question Panel, this dropdown provides options for candidates to select their answers from predefined choices. The dropdown may vary based on the question type (e.g., multiple-choice, single-choice, etc.).
- **Timer (Right Top):** Positioned in the top-right corner of the page, the Timer displays the remaining time for the assessment. It helps candidates manage their time effectively and stay aware of the time constraints.
- **Footer Section:**
 - **End Session Button:** Placed at the bottom of the page, the End Session button allows candidates to conclude the assessment before the allocated time expires. Clicking this button prompts a confirmation message before ending the session.
 - **Skip and Next Buttons:** Positioned in the footer section, these buttons enable candidates to navigate between questions. The Skip button allows candidates to skip a question and move to the next one without providing an answer. The Next button takes candidates to the next question after they have selected their answer from the dropdown.

Purpose: The Test Page serves as the core component of the assessment process. It provides candidates with a clear view of the questions, facilitates their selection of answers through the dropdown, and ensures effective time management through the timer display. The user-friendly layout and intuitive navigation options enable candidates to focus on the questions and efficiently progress through the assessment.

The End Session button offers candidates the flexibility to conclude the assessment at any time, while the Skip and Next buttons enable them to move forward, allowing for non-linear navigation based on their preferences or comfort with specific questions.

By providing a well-structured and intuitive interface, Gradr.io aims to optimize the candidate's testing experience and facilitate accurate evaluation of their knowledge and skills.

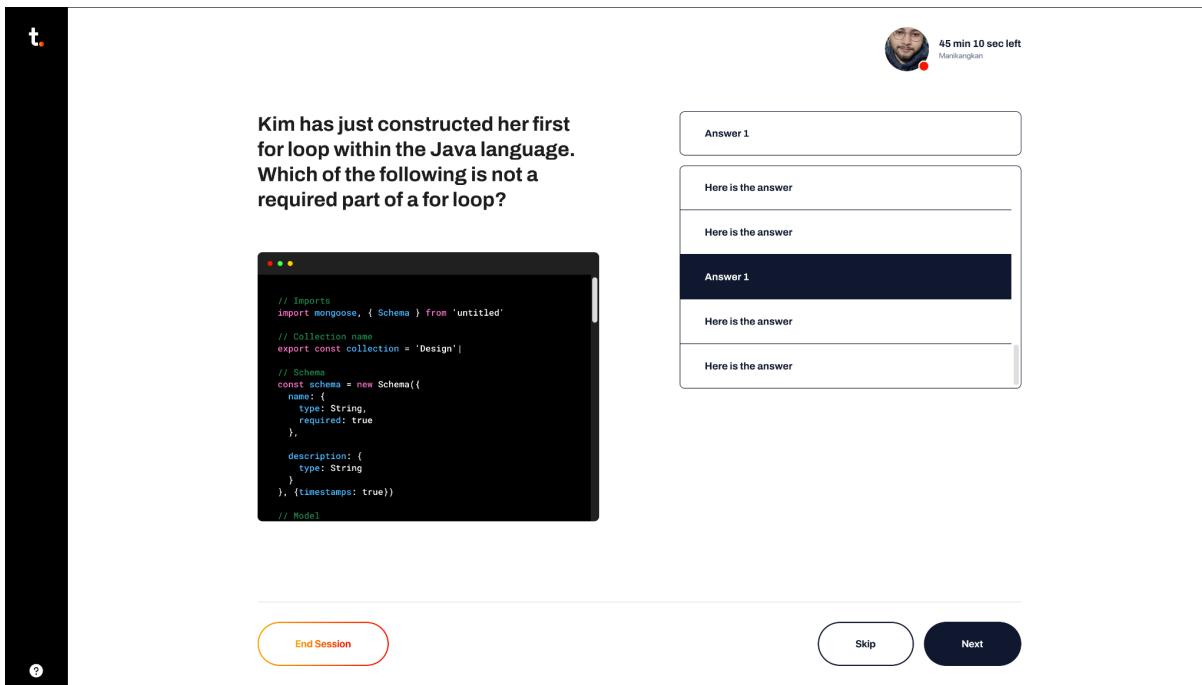


Fig 10: Test page

4.1.2.2 Answer Select Dropdown:

Instead of using traditional multiple-choice radio buttons, Gradr.io utilizes an Answer Select Dropdown to facilitate the candidate's response to each question. This approach is specifically designed to align with the unique requirements of the assessment process in Gradr.io, where candidates need to have knowledge of the answer in order to provide a response.

The Answer Select Dropdown is located below each question in the Question Panel on the left side of the Test Page. It presents a list of predefined choices related to the question, allowing candidates to select the appropriate answer from the dropdown options.

By employing this approach, Gradr.io ensures that candidates actively engage with the assessment content and are not limited to guessing or random selection. It promotes a deeper understanding of the subject matter and rewards candidates who possess the necessary knowledge and skills.

The Answer Select Dropdown provides a clear and concise selection mechanism, enabling candidates to choose their response efficiently. It enhances the assessment's accuracy and validity by allowing candidates to demonstrate their understanding and comprehension of the content.

Gradr.io's unique implementation of the Answer Select Dropdown sets it apart from other platforms, as it focuses on assessing the candidate's knowledge rather than relying solely on multiple-choice radio buttons. This approach fosters a more meaningful evaluation and provides valuable insights into the candidate's abilities and expertise.

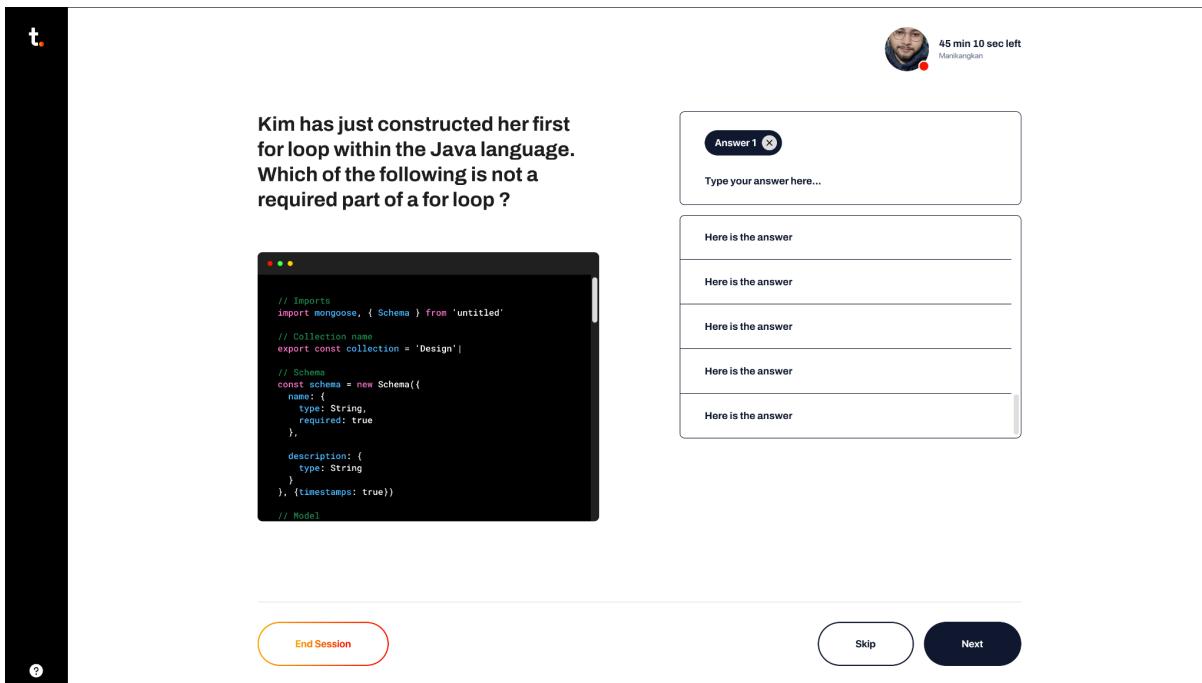


Fig 11: Test page with select answer

4.1.2.3 End Session Modal:

The End Session Modal in Gradr.io provides candidates with the flexibility to end their test session at any time. This modal is accessible through the End Session button located in the footer of the Test Page. When the candidate clicks on the End Session button, a modal window appears, presenting them with a confirmation message. The modal prompts the candidate to confirm their decision to end the test session.

By offering this feature, Gradr.io ensures that candidates have control over their testing experience and can conclude their assessment when needed. It allows candidates to manage their time effectively and provides a convenient option for exiting the test interface without any complications.

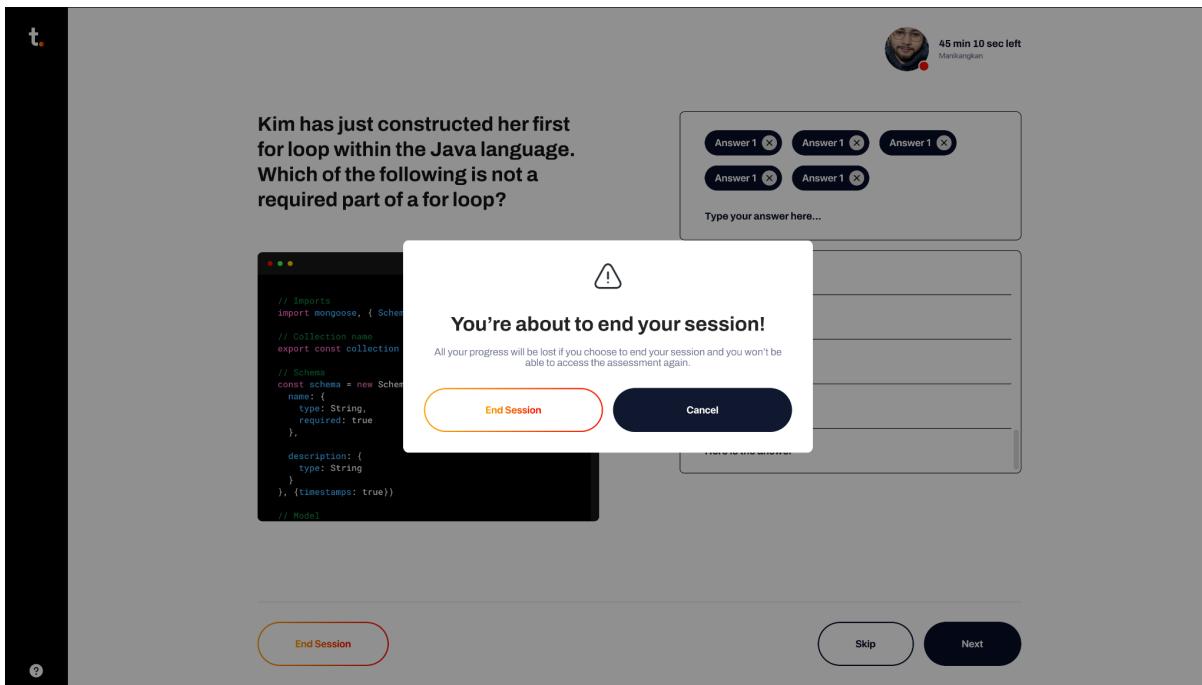


Fig 12: Test page with end session modal

4.1.2.4 Feedback Modal:

The Feedback Modal in Gradr.io enables candidates to provide valuable feedback about any technical issues, crashes, or bugs they may encounter while using the platform. This modal is designed to gather feedback and improve the overall user experience.

Candidates can access the Feedback Modal by selecting the Feedback option in the navigation menu or through a dedicated button within the platform. Once opened, the modal presents a user-friendly form where candidates can describe the issue they encountered, provide details about the specific problem, and offer suggestions for improvement.

Gradr.io values the feedback provided by candidates, as it helps identify and address any technical issues promptly. This proactive approach ensures a smooth and reliable testing experience for all users. The Feedback Modal serves as a channel for candidates to contribute to the ongoing enhancement of the platform, making it more robust and user-friendly for future test takers.

By incorporating the End Session Modal and Feedback Modal, Gradr.io prioritizes user satisfaction and continuously strives to deliver an efficient and reliable assessment tool.

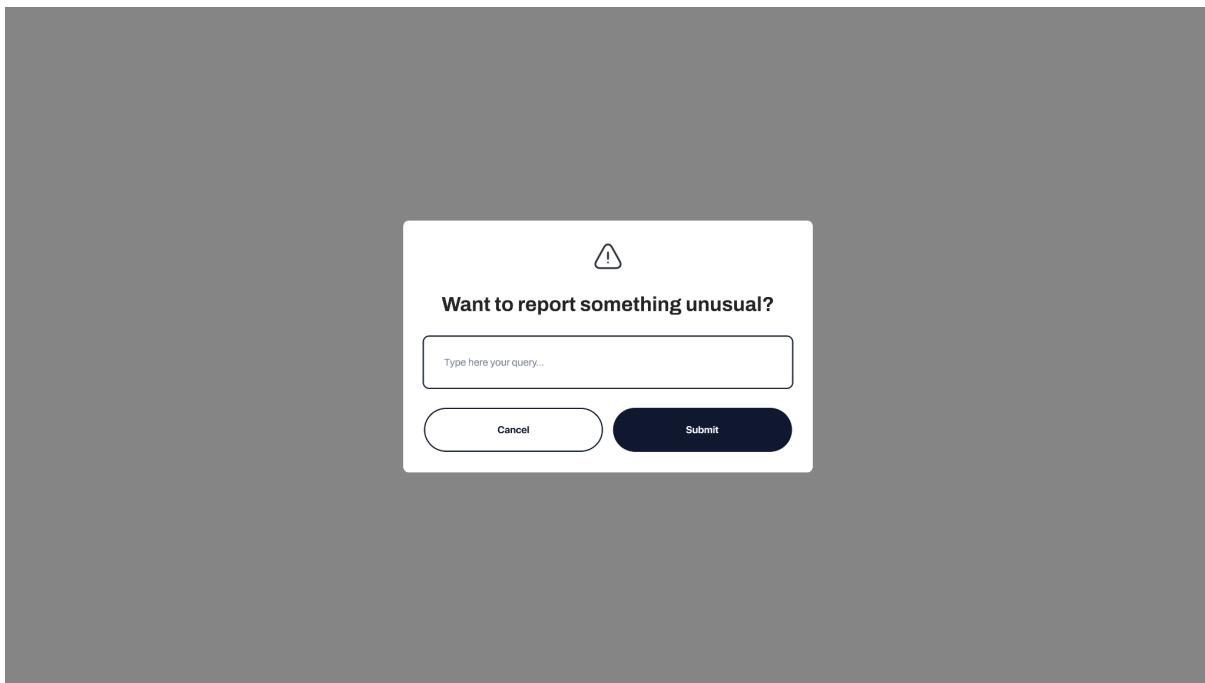


Fig 13: Test page with feedback modal

4.1.2.5 Candidate has to go through multiple topic sections

Next section	GIT
Section time	25 min

Kindly ensure to not close this window or navigate away from the page during the assessment process.

Next

Fig 14: Test section info page

4.1.3 Post-flow

4.1.3.1 Assessment Summary Screen:

Once a candidate completes the entire assessment in Gradrio, they are presented with an Assessment Summary Screen. This screen provides a comprehensive overview of the candidate's performance throughout the assessment, offering valuable insights into their progress and achievements.

The Assessment Summary Screen displays section-wise time and attempted question statistics. Candidates can view the amount of time they spent on each section, allowing them to evaluate their time management skills and identify areas where they may need improvement. This information enables candidates to reflect on their performance and make necessary adjustments in future assessments.

Additionally, the Assessment Summary Screen presents the number of attempted questions in each section. Candidates can see how many questions they answered in each section, giving them a sense of their level of engagement and completion. This data helps candidates assess their coverage of the assessment content and gauge their overall effort and thoroughness.

Gradrio's Assessment Summary Screen serves as a useful tool for candidates to review their performance holistically and gain valuable insights into their strengths and areas for improvement. By providing section-wise time and attempted question statistics, this screen offers candidates a clear picture of their performance and aids in their self-assessment and learning process.

The screenshot shows a mobile application interface for an assessment summary. At the top, there is a small orange logo followed by the letter 't.'. Below this is a light gray header bar. The main content area features a table with a white background and black borders. The table has four columns: 'Section', 'Time', 'Attempted', and 'Status'. There are six rows in the table, each representing a different section of the assessment. The sections and their details are as follows:

Section	Time	Attempted	Status
HTTP	20 min	43	Attempted
UI & UX	15 min	65	Attempted
Frontend	4 min	32	Attempted
Backend	11 min	43	Attempted
Network	45 min	23	Attempted

In the bottom right corner of the main content area, there is a dark blue circular button with the word 'Next' written in white.

Fig 14: Test marks show page

4.8 Next Section Info Screen:

Upon completing a section of the assessment, candidates are presented with the Next Section Info Screen in Gradr.io. This screen provides candidates with information about the upcoming section of the test, allowing them to prepare and transition smoothly to the next set of questions.

The Next Section Info Screen includes details such as the title or name of the next section, a brief description or instructions related to the section, and any additional information that may be relevant for the candidate. This information helps candidates understand the context and expectations of the upcoming section, enabling them to focus their attention appropriately.

Additionally, the Next Section Info Screen displays the allocated time for the next section. This time information serves as a reminder for candidates, enabling them to manage their time effectively and allocate sufficient effort to complete the upcoming section within the specified time frame.

Gradr.io's Next Section Info Screen aims to provide candidates with a seamless testing experience by offering clear guidance and an overview of what to expect in the subsequent section. This feature allows candidates to mentally prepare themselves, ensuring a smooth transition and optimal performance throughout the assessment.

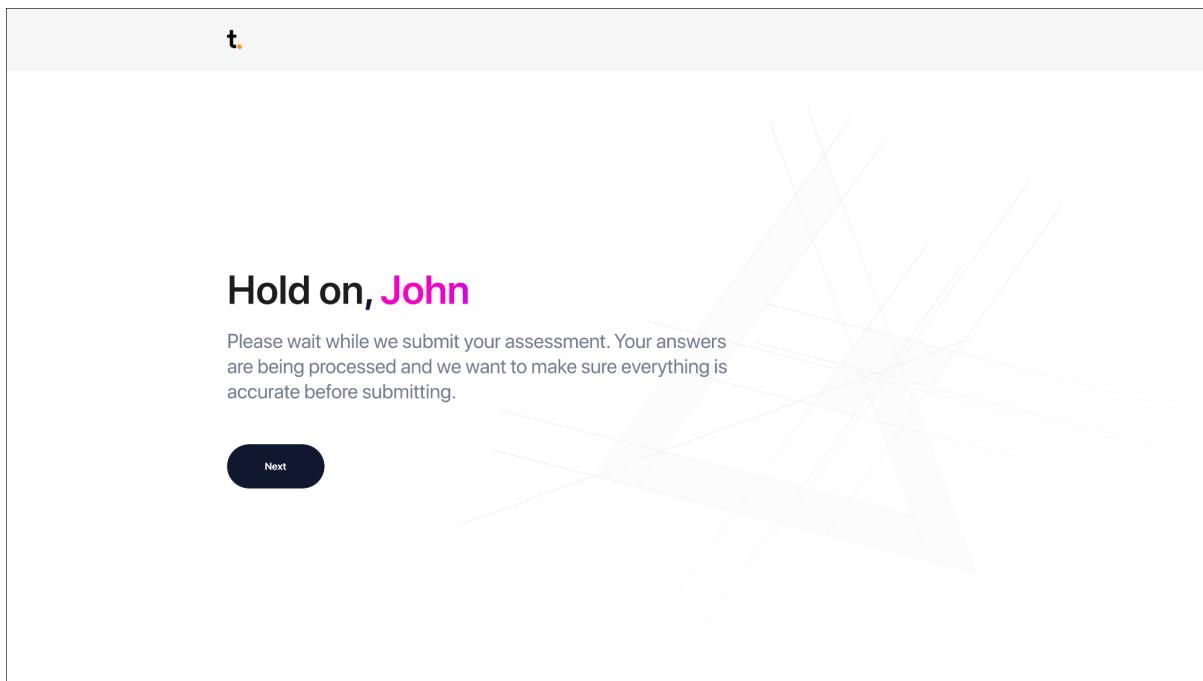


Fig 15: Test submit page

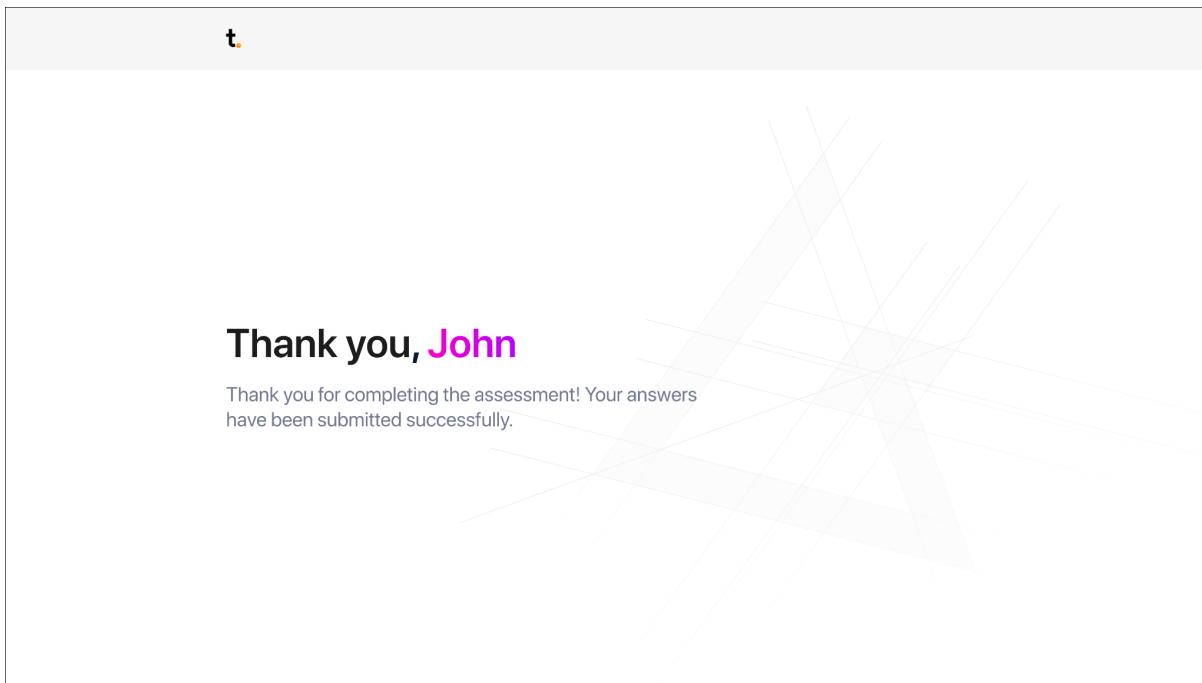


Fig 16: Test done page

4.2 Admin Site

- **Creating an Assessment** To get started on Grader.io's Admin Panel, the first step is to log in. Once logged in, click on the "Create Assessment" button. Fill in the assessment name and select the start and end times for the assessment. You can also choose from a variety of subjects. After selecting the subjects, add candidates by entering their email addresses. Finally, click on "Submit" to create the assessment.
- **Log in to Grader.io's Admin Panel:** Access the admin panel by logging in with your credentials.
- **Click on "Create Assessment":** Once logged in, locate the "Create Assessment" button and click on it.
- **Fill in Assessment Details:** Provide a name for the assessment and select the start and end times during which the assessment will be active for candidates.
- **Choose Assessment Subjects:** Select the subjects or topics for the assessment from the available options. This helps categorize the assessment and align it with the relevant content.
- **Add Candidates:** Enter the email addresses of the candidates you want to invite for the assessment. These candidates will receive unique invitations to participate.
- **Specify Maximum Applications:** Set the maximum number of candidates that can participate in the assessment. This ensures a controlled number of applicants for effective evaluation.
- **Submit the Assessment:** Once all the necessary details are filled in, click on the "Submit" button to create the assessment.

4.2.1. Login to admin panel

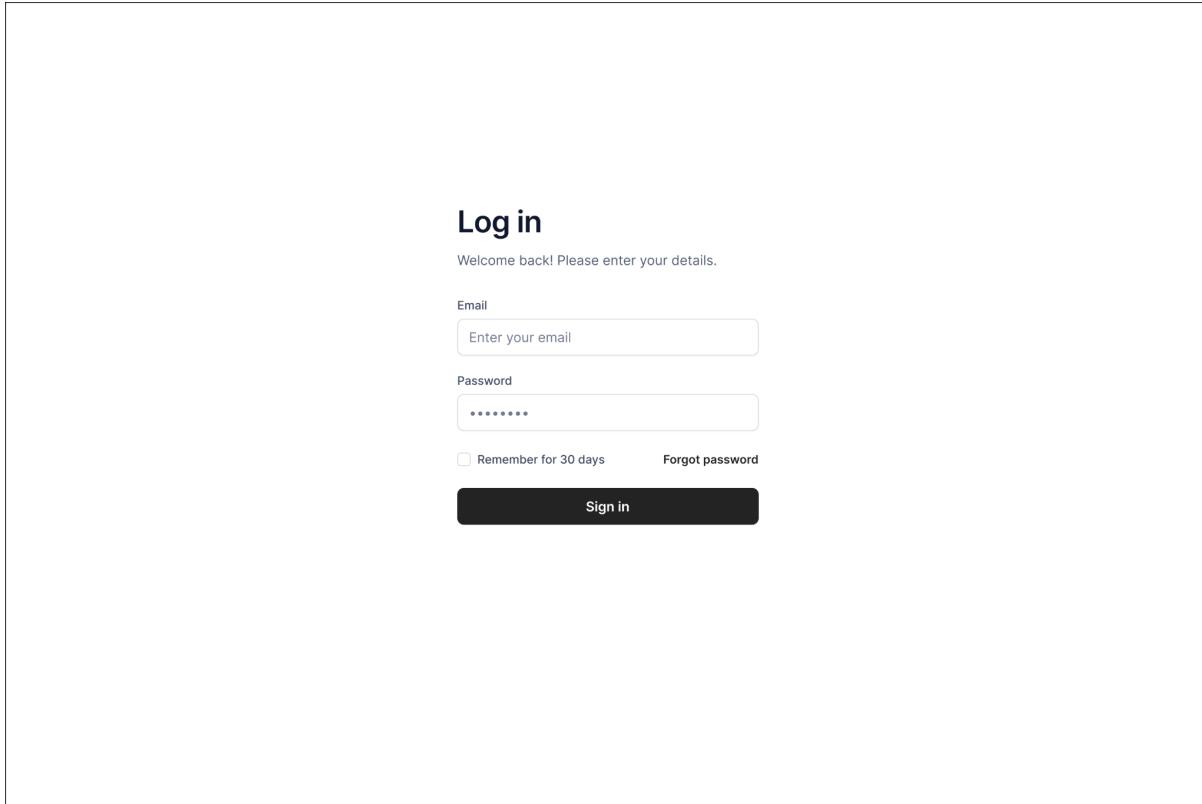


Fig 18: Admin panel login page

4.2.2. Admin Dashboard

The dashboard has a sidebar on the left with "Gradr.io" logo, "Home", and "Analytics" buttons. The main area starts with a greeting "Good morning, Mani" and a search bar. It features six cards representing different roles: "Frontend dev", "Backend dev", "Network security", "UI & UX Designer", "Sales & marketing", and "Human resource". Each card displays four metrics: Total (454), Attempted (43), Shortlisted (20), and Assessment expired (10). The "Network security" card includes edit, share, and delete options. At the bottom, there's a user profile for "Manikangan Das" (mani@techwondoe.com), navigation buttons for "Previous" and "Next", and a page number indicator from 1 to 10.

Fig 19: Admin dashboard

4.2.2. Create new assessment

The screenshot shows the 'Untitled assessment' creation page. On the left is a sidebar with 'Gradr.io' logo, 'Home', and 'Analytics'. The main area has a title 'Untitled assessment' and a sub-section 'Set assessment settings'. It includes fields for 'Name' (Untitled assessment), 'Start time' (with 'Select date' and 'Type start time' buttons), 'End time' (with 'Select date' and 'Type start time' buttons), 'Select sections' (a search bar showing 'Git', 'UI & UX', and 'Frontend'), and 'Max applicants' (set to 99 with a note: 'Max applicants should be less than or equal to 100'). A 'Next' button is in the top right. At the bottom left is a user profile for 'Manikangan Das mani@techwondoe.com'.

Fig 20: Assessment creating page 1

The screenshot shows the 'Frontend Dev' assessment creation page. The layout is identical to Fig 20, with a sidebar and 'Set assessment settings' section. The 'Name' field is filled with 'Frontend & UI/UX'. The 'Start time' and 'End time' fields are present. The 'Select sections' field is empty. The 'Max applicants' field is set to 100 with a note: 'equal to 100'. A date picker modal is open over the 'End time' field, showing 'January 2022' with 'Jan 6, 2022' to 'Jan 13, 2022'. The calendar grid shows dates from Jan 1 to Jan 31, with Jan 6 highlighted. Buttons for 'Cancel' and 'Apply' are at the bottom of the modal. The bottom left shows the user profile for 'Manikangan Das mani@techwondoe.com'.

Fig 21: Assessment creating page 2

The screenshot shows the 'Frontend Dev' assessment settings page. The 'Name' field contains 'Frontend & UI/UX'. The 'Start time' section includes a 'Select date' button and a dropdown menu for 'Type here time' with options from 10.00 AM to 16.00 AM. The 'End time' section has a 'Select date' button. The 'Select sections' section shows a search bar with 'Git', 'UI & UX', and 'Frontend' selected. The 'Max applicants' field is set to '99'. A note below says 'Max applicants should be less than or equal to 99'. On the left sidebar, there are links for 'Home' and 'Analytics'. At the bottom, there is a user profile for 'Manikangan Das' with the email 'manji@techwondoe.com'.

Fig 22: Assessment creating page 3

4.2.4 Adding Candidates to an Assessment

As an admin, you can add candidates to an assessment even after creating it. Locate the assessment and click on the three dots. From the dropdown menu, select "Invite" and enter the candidate's email address. The candidate will receive an email with a unique link to access the assessment during the active period. This allows you to easily add candidates and ensure their participation.

- a. **Adding Candidates:** After filling in the assessment details, including the start and end times, you can add candidates by entering their email addresses.
- b. **Invitation Process:** Once the assessment is created and the candidates are added, the system will generate unique links for each candidate.
- c. **Email Notification:** An email will be sent to each candidate's provided email address, containing the unique link to access the candidate page.
- d. **Accessing the Candidate Page:** Candidates need to open the email and click on the unique link provided. This will direct them to the candidate page within Grader.io's platform.
- e. **Test Instructions:** On the candidate page, candidates will be greeted with instructions about the test process and what is expected from them.
- f. **Personal Information:** Candidates will be prompted to fill in their basic information, such as their name and optional profile.

- g. **Test Start:** After completing the necessary information, candidates can proceed to start the test by following the given instructions.

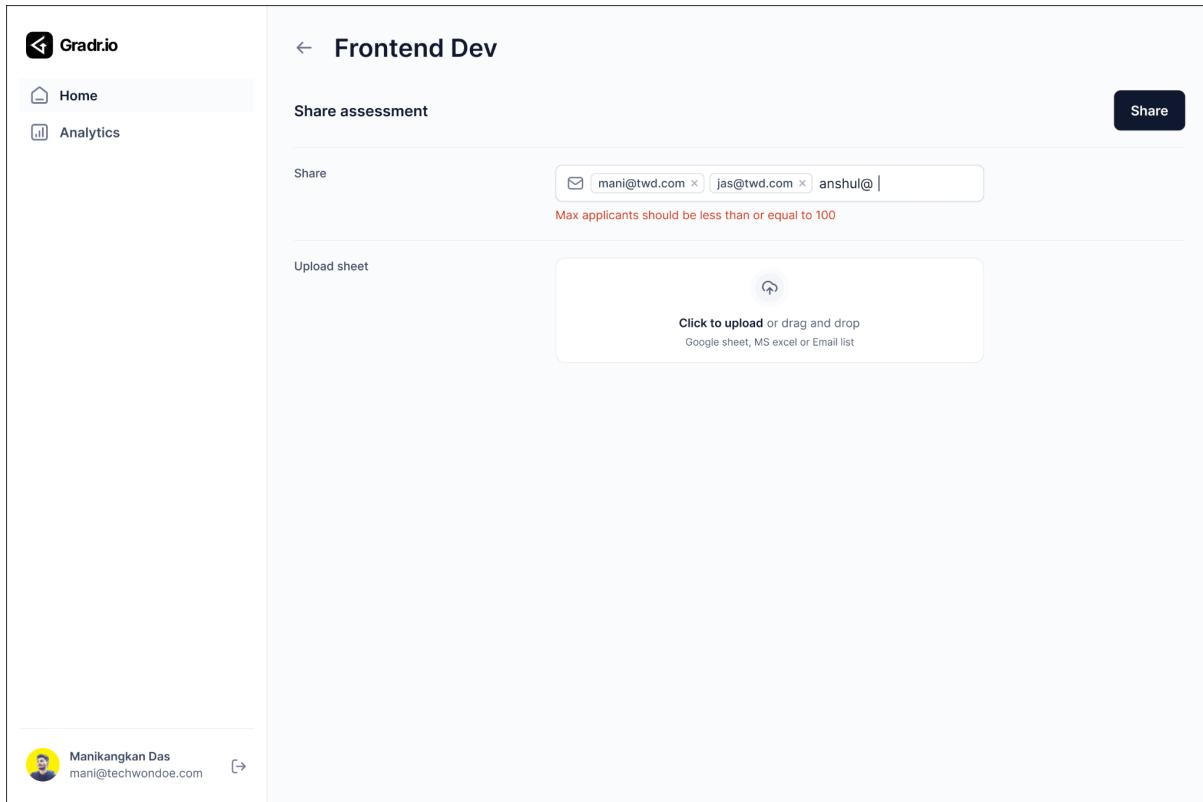


Fig 23 Candidate invitation page

4.2.5 Analytics page

The Analytics page in Grader.io's Admin Panel provides valuable insights into candidate performance and assessment data. Here are the key features and information available on the Analytics page:

- **Candidate Performance:** The Analytics page allows administrators to view the performance of individual candidates. It provides a comprehensive overview of their scores, time taken to complete the assessment, and section-wise performance.
- **Global Candidate Pool:** Administrators can compare a candidate's performance with the global pool of candidates who have taken the same assessment. This comparison helps assess the candidate's relative performance and provides valuable context.
- **Assessment Details:** The Analytics page displays information about each assessment, including the assessment name, start and end times, and overall statistics.
- **Time Information:** Administrators can analyze the average time taken by candidates to attempt each question. This data helps identify if candidates are spending an appropriate amount of time on each question or if they are rushing through the assessment.

- **Browser Information:** Grader.io captures information about the browser used by candidates to access the assessment. This data helps administrators ensure compatibility and identify any issues related to specific browsers.
- **Graphical Representation:** To make the data more visually appealing and easier to interpret, the Analytics page presents the information in various graphical formats. These visual representations may include bar charts, line graphs, pie charts, or other visualizations, depending on the type of data being presented.
- **WebCam snapshot feature:** Grader.io's Analytics page includes a valuable Webcam Snapshot feature. This feature enables administrators to monitor candidate activity during the assessment by capturing screen snapshots through the candidate's webcam.

By leveraging the Analytics page, administrators can gain valuable insights into candidate performance, identify trends, and make data-driven decisions to improve the assessment process and candidate experience.

 MD

Manikangan Das
manikangan.das@techwondoe.com

[Download as PDF](#)



Assessment Name

Frontend Engineering



Time

1 Aug 23, 7:30-8:30 AM

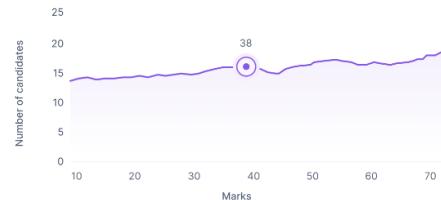


Browser Used

Google Chrome

Global Performance

It refers to the overall performance of a system or entity, such as a website, application, or business.



Overall Performance

It refers to the overall performance of a system or entity, such as a website, application, or business.



Section wise performance

It refers to the overall performance of a system or entity, such as a website, application, or business. The analytics page tooltip provides detailed insights and metrics on this performance, enabling users to monitor and optimize the system's overall effectiveness.

GIT



Amateur



GIT

Total marks

32

Skill level

Amateur

Highest score global

Intermediate

Attempted

60

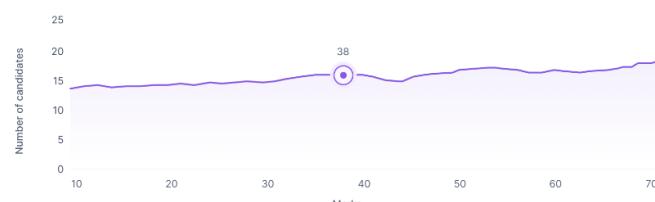
Correct

40

Skip questions

90

Javascript

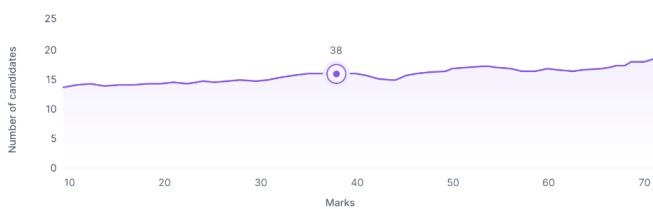


Expert



GIT

GIT



Amateur



GIT

Total marks

23

Skill level
Amateur

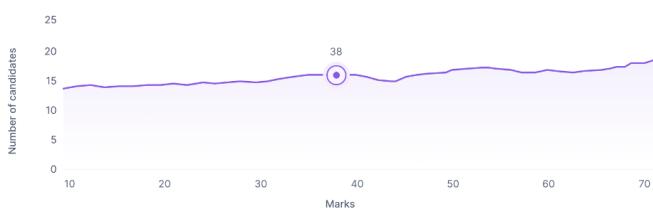
Attempted
60

Correct
40

Highest score global
Intermediate

Skip questions
90

GIT



Intermediate



GIT

Total marks

67

Skill level
Amateur

Attempted
60

Correct
40

Highest score global
Intermediate

Skip questions
90

Miscellaneous performance

It refers to the overall performance of a system or entity, such as a website, application, or business. The analytics page tooltip provides detailed insights and metrics on this performance, enabling users to monitor and optimize the system's overall effectiveness.



Average time taken to attempt a particular question

00.25

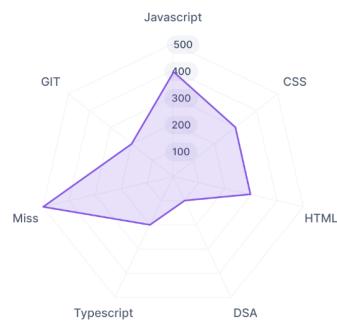


Average time taken to attempt a particular question globally

00.34

Radar chart

It refers to the overall performance of a system or entity, such as a website, application, or business. The analytics page tooltip provides detailed insights and metrics on this performance, enabling users to monitor and optimize the system's overall effectiveness.



Screenshots taken

It refers to the overall performance of a system or entity, such as a website, application, or business. The analytics page tooltip provides detailed insights and metrics on this performance, enabling users to monitor and optimize the system's overall effectiveness.



Resume

It refers to the overall performance of a system or entity, such as a website, application, or business.

[Download as PDF](#)

Fig 24: Analytics page - admin view

Chapter Five

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Gradr.io has established itself as a leading platform in the pre-employment testing landscape with its cutting-edge Software as a Service (SAAS) platform. By providing tailored assessments and comprehensive skill coverage, Gradr.io empowers employers to make informed hiring decisions. The platform stands out from its competitors by offering advanced features such as customizable assessments, an AI-powered question pool, a cheating-proof mechanism, and global/local skill comparisons. With a user-friendly interface and a great candidate experience, Gradr.io streamlines the candidate screening process for organizations.

Key Takeaways:

- Gradr.io offers a comprehensive solution for assessing technical, logic, and soft skills required for job roles.
- The platform provides tailored assessments, allowing employers to customize assessments based on specific job requirements.
- Gradr.io covers a wide range of technical skills, ensuring a thorough evaluation of candidates across various domains.
- The platform incorporates an inherently cheating-proof mechanism, including webcam verification, IP checks, copy-paste detection, fullscreen checks, and mouse pointer analysis.
- Gradr.io introduces unique features such as unlimited questions in a specified time frame, incorporating time constraints for scoring, and global/local skill comparisons.
- Employers can benchmark candidates' skills against a global talent pool, gaining insights into candidates' relative proficiency in specific areas.
- Gradr.io promotes bias-free recruitment by providing standardized assessments based on objective evaluation criteria.
- The platform prioritizes a great candidate experience and offers a clean and intuitive interface for the hiring team.

In conclusion, Gradr.io's SAAS platform addresses the challenges of pre-employment testing by providing organizations with a comprehensive, customizable, and efficient solution. Leveraging advanced technologies and unique features, Gradr.io empowers employers to identify top talent with the necessary technical, logic, and soft skills. With a commitment to continuous improvement and adaptability, Gradr.io aims to remain at the forefront of the candidate screening process, enabling organizations to make confident and informed hiring decisions.

5.2 Future Works

Gradr.io has several exciting avenues for future work, including:

- Incorporating Artificial Intelligence (AI) for Question Generation:
 - Integrating AI technology to generate assessment questions.

- Leveraging natural language processing (NLP) algorithms to generate diverse and contextually relevant questions.
 - Enhancing the question generation process to ensure comprehensive coverage of technical, logic, and soft skills.
 - Utilizing machine learning techniques to continuously improve the question generation algorithm based on user feedback and data analysis.
- Tracking Candidate Behavior and Performance:
 - Implementing advanced tracking mechanisms to monitor candidate behavior and performance during assessments.
 - Analyzing candidate interactions, time spent on questions, and response patterns to gain insights into their problem-solving approach and skill level.
 - Applying data analytics techniques to identify patterns and trends that help evaluate candidate suitability and potential.
- Creation of Dynamic and Engaging Landing Pages:
 - Developing dynamic landing pages that showcase the features and benefits of Gradr.io in a visually appealing and interactive manner.
 - Incorporating animations, videos, and interactive elements to engage and captivate potential users.
 - Optimizing landing page content to effectively communicate the value proposition of Gradr.io and highlight its unique selling points.
 - Conducting user testing and feedback collection to refine and enhance the landing page design and user experience.
- Scalability Testing with Large User Base:
 - Conducting scalability testing to ensure Gradr.io can handle a large number of users and assessments concurrently.
 - Simulating high traffic scenarios to evaluate system performance and identify potential bottlenecks.
 - Optimizing the platform infrastructure and database to handle increased user load and maintain a seamless user experience.
 - Employing load testing tools and techniques to assess the platform.

REFERENCES

1. Quizco frontend - [manikangkan/quizco-frontend: Quizco: Quiz Builder + Assessment Tool \(github.com\)](https://github.com/manikangkan/quizco-frontend)
 2. Quizco backend - [manikangkan/quizco-backend \(github.com\)](https://github.com/manikangkan/quizco-backend)
 3. Friendship quiz - [manikangkan/friendship-quiz: Friendship quiz is a platform where an user can create quiz & play with friends. \(github.com\)](https://github.com/manikangkan/friendship-quiz)
 4. React js - [React](https://reactjs.org/)
 5. Tailwind css - [Tailwind UI - Official Tailwind CSS Components & Templates](https://tailwindcss.com/)
 6. Node.js - [Node.js \(nodejs.org\)](https://nodejs.org/)
 7. Nest.js - [NestJS - A progressive Node.js framework](https://nestjs.com/)
 8. Express.js - [Express - Node.js web application framework \(expressjs.com\)](https://expressjs.com/)
 9. Typescript - [TypeScript: JavaScript With Syntax For Types. \(typescriptlang.org\)](https://typescriptlang.org/)
-