System Requirements

<P05>:< InstaShop>

<team member names & ids>

Student ID	Name
23100091	MUHAMMAD BILAL SHAHID
23100254	SALMAN MASOOD
23100233	MUHAMMAD UMAIR MOHSIN
23100268	FARVA TALIB
23100016	MUHAMMAD AFFAN ASHRAF

TABLE OF CONTENTS

1. Introduction

<Give an overview of the project here. The overview must highlight the overall objectives of the project and its potential users.>

InstaShop will be a web-based portal which aims to revolutionize influencer marketing. For those who may be unaware, influencer marketing is a recent development in the digital world which involves a brand collaborating with an online influencer to market one of its products or services. These influencers are usually found on social media platforms (Instagram and Snapchat) and have a decent number of followers to whom they market a particular brand's products or services.

There are two main parties involved in this process, the client and the influencer. For better understanding of the motivation behind this project, we must first walk through the process that is generally followed in influencer marketing from the perspective of both the client and the influencer.

For the client, the first step is perhaps the most tedious which is related to finding the right influencer for your brand. This is usually achieved by manually searching social media platforms or relying on word of mouth from friends/family regarding a particular influencer. The second step is contacting the shortlisted profiles. This is either done through direct messages or emails, both of which are again tedious tasks with no guarantees of a timely response. The final step (assuming the contract has been fulfilled) is the issue of payment. This is perhaps the most troublesome aspect because there is no guarantee that the influencer will produce content that is upto the mark and not run off with any advance payments made.

For the influencer (unless they have a huge following) it is usually hard to find clients for collaborations or for sponsored content. Since these influencers are very active on social media, their inbox and comment sections are almost always flooded which means they tend to miss out on potential business opportunities just because they weren't able to see the direct message. Some profiles do have designated emails for business inquiries but their response times are in most cases not ideal because they just don't check their email that often. Finally, the issue of payment also exists. There is no guarantee that the client will pay the influencer the full amount in a timely manner even if the work has been done upto the client's standards and deadline.

As visible, both of these parties are in dire need of a platform that could automate most of these tasks for them as well as provide payment guarantees. This is where **InstaShop** comes in. It aims to streamline all the steps involved in this process by providing an easy to use web

application thus saving time and effort for both parties involved. For the client, it makes it easier to search relevant influencers courtesy of our database and filtering method, connect with shortlisted influencers (via email or live chat) and have your payment secured (via escrow).
For the influencer, you essentially get access to a marketplace where you can find potential clients, not miss out on potential business opportunities just because your inbox was too cluttered and have a guarantee that the client will pay you for your work.

2. System Actors

<List down the actor names and give a 2-3 lines description of the role of each actor>

Actor Name	Description
Admin	The admin will be overlooking the whole application and performing functions such as ensuring efficient flow of money, approving the list of influencers and clients on board, and blocking any actor if needed.
Client	The client will log in and see the list of influencers according to their needs. Clients will be able to apply filters for a narrower search, see their contact details, and pay and rate them after task completion.
Influencer	The influencer will log in and see his/her messages to see if any client has contacted them. He/She will then be able to either accept or reject the client’s offer. The influencer will be able to rate the experience with the client after payment.

3. Functional Requirements

<Write system requirements from users' (actors) perspective. Actor names have been highlighted in the sample requirements below. You may group requirements according to actors or modules>

Requirements	
Sr#	**Requirement**
1	As an Admin, I should be able to login to my account if my account is already created.
2	As an Admin, I should be able to control the payment method and its flow between the application users.
3	As an Admin, provide a list of influencers based on category.
4	As a customer, I should be able to login to my account if it is already created.
5	As a customer, I should be able to sign up to create a new account.
6	As a customer, I should narrow my search for influencers using filters.
7	As a customer, I should be able to see a list of influencers based on the entered filter.
8	As a customer, I should be able to view contact details of a particular influencer.
9	As a customer, I should be able to view my profile which contains a list of all my projects (successful and ongoing) and details associated with them.
10	As a customer, I should be able to rate influencers after task completion.
11	As a customer, I should be able to make payment to the influencer.
12	As an influencer, I should be able to login to the account if already created.
13	As an influencer, I should be able to sign up to create a new account.
14	As an influencer, I should be able to check my inbox containing the messages from customers.
15	As an influencer, I should be able to accept a request for marketing and send confirmation to customers.
16	As an influencer, I should be able to reject a request.
17	As an influencer, I should be able to rate the customer.

4. Non-functional Requirements / Quality Attributes

<Requirements must be testable>
<Security requirements fall in the category of "Non-functional requirements"; however, you need to list them separately in the section **Security Requirements** later in this document.>

Although functionality and design are important for a successful user experience, a user notices a website's performance immediately. Studies show that 47% of people don't visit websites with load times of more than two seconds. A single second can cost an e-commerce website millions of dollars per year because it can reduce conversion rates by 7%. A website's speed has an impact on its SEO rating, therefore one that loads quickly has advantages. Thus, it is clear that performance is important while trying to assure a good user experience. (https://www.browserstack.com/guide/why-website-speed-is-important)

Sr#	Requirements
1	At no point during operation should the system use more memory than 1 GB.
2	No more than three failures should occur in the system per 24-hour period. In the event of a failure, the system should resume normal operations within five minutes.
3	It shouldn't take more than 5 seconds for a user to log.
4	Authentication should take no more than 2 or 3 seconds.
5	Searching for influencers should not take more than 3 seconds.
6	Filtering influencers by area of expertise shouldn't take longer than 2 seconds.
7	Within two or three seconds, messaging applications should deliver texts.
8	System should be able to handle at least 10,000 users at once without crashing. (This is our initial goal which can be modified once our app gets more users.)
9	The execution of smooth payment should not take more than a minute.
10	It should not take more than a second to rate clients and influencers.
11	It should not take more than a second for an influencer to send confirmation or rejection to the clients.
12	Our app should be available to its users 24/7.

5. Security Requirements

< Go through OWASP top 10 security risks (https://owasp.org/www-project-top-ten/).

(a) Select **3 security risks** that you think are top threats for your system. While doing this, carefully consider the information/functionality that is most vulnerable from security perspective in the context of your project.

(b) For each security risk (identified above), identify **potential losses** (e.g., financial loss, total business loss, litigation etc.) if you do not take necessary measures to address the above security risks.

(c) Identify the **controls** (e.g., input validation, audit logs, multi-factor authentication, user roles etc.) that should be implemented in your system in order to address the above security risks.

Sr #	Security Risks	Potential Losses	Controls
1	Broken Access Control	Metadata Manipulation , Business loss	Implement access control mechanisms , Log access control failures alert admins , Stateful session identifiers be invalidated on the server after logout , Rate Limit API
2	Injection	Financial Business loss , Data Breached , Litigation	Use Safe APIs , Use positive server-side input validation , Using LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.
3	Identification and Authentication failures	Audit logs ,	Implement multi-factor authentication , implement weak password check , Use a server-side, secure, built-in session manager to generates a new-random session ID with high entropy after

			login , Session identifier should not be in the URL, Align password length , Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks

<Explore and select **one static and one dynamic security scanning tools** for your project. The tools should be selected considering languages and technologies being used in your project.>

Sr#	**Security Tool Name**	**Brief description** (why the tool is suitable for your project)
1	Intruder (Dynamic)	It is a cloud based scanner and it will help us find the weaknesses in our online systems to prevent hackers. It saves you time by proactively scanning new threats and makes it easy to interpreting the threat easier
2	SonarQube(Static)	Compatible with our programming languages , open source community , selection of thousands of automated Static Code Analysis rules, which are flagged during coding, saving developers' time

Alternative options:

1) Dynamic(Gitlab) -> DevOps platform, delivered as a single application, fundamentally changing the way Development, Security, and Ops teams collaborate and build software. From idea to production, GitLab helps teams improve cycle time from weeks to minutes, reduce development process costs and decrease time to market while increasing developer productivity.
2) Codacy (static) -> Codacy provides either a cloud or self-hosted platform which checks for code standardization, quality and security errors, notify you of errors from within your current workflows and platforms , open source community , good user interface .

6. Who Did What?

Name of the Team Member	Tasks done
Muhammad Bilal Shahid	Introduction
Affan Ashraf	Actors
Salman Masood	Functional Requirements
Farva Talib	Non-Functional Requirements
Muhammad Umair Mohsin	Security Requirements

7. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

Section Title	Reviewer Name(s)
Introduction	Muhammad Umair Mohsin
Actors	Salman Masood
Functional Requirements	Muhammad Bilal Shahid
Non-functional requirements	Muhammad Bilal Shahid
Security Requirements	Farva Talib