## უსაფრთხოების ზომები

უსაფრთხოების მიმართულებით პროექტის პრეზენტაციამდე გატარდა შემდგომი ღონისძიებები: უწყვეტი სტატიკური ანალიზი კოდის პროდაქშენ გარემოში გაშვებამდე, კოდირების OWASP საუკეთესო პრაქტიკის შესაბამისობა, ძლიერი მულტიფაქტორული ავთენთიფიკაციის და წვდომის კონტროლის მექანიზმების დანერგვა, მომხმარებლის პერსონალური და სენსიტიური ინფორმაციის დაცვა, უსაფრთხო კონფიგურაციები და დიზაინი: არასაჭირო პორტების, პროტოკოლების, სერვისების და ბრძანებების ინაქტივაცია აპლიკაციაში, სესიების უსაფრთხოდ მართვა, მათ შორის უსაფრთხო შიფრაცია, უსაფრთხო ლოგირება, API-ების და მონაცემთა ბაზების უსაფრთხოება, უსაფრთხოების რეგულაციების შესაბამისობა, სრული და ინკრემენტული სარეზერვო ასლების კოპირება, მოწყვლადობის სკანირება, უახლოეს მომავალში უსაფრთხოების ზომების მიღება ხელსაწყოების ოპტიმალური კომბინაციით, როგორიცაა DDos Protection Tool, WAF, EDR, NGFL, PAM, SIEM, email spam filter და სხვა.

**Information Security Pinpoints:**

1. MFA – for 1-minute active code on a mail (Least Privilege Principle for specific roles). What would be used for identification, secure authentication and credential recovery
2. Secure codding practices [OWASP Developer Guide | OWASP Foundation](https://owasp.org/www-project-developer-guide/) – OWASP Tools and OWASP Application Security Verification Standard compliance - Code Vulnerability hardening, Input Validation, Output Encoding, masking sensitive info, secure API and so on
3. Static Analysis Tools - [Source Code Analysis Tools | OWASP Foundation](https://owasp.org/www-community/Source_Code_Analysis_Tools)
4. Password Policy- attached document in word format
5. User consent on Privacy Policy before user registration – attached document in word format
6. Session Management: Session encryption AES 256, Inactive up to 30 minute Session blocking to re-authenticate user with code (1min), generation of ungues able unique sessions each time, each session termination, encryption key secure safe
7. Secure logging and what are information to log: authentication information especially failure log on attempts, user activity
8. Secure API and Database – API Authentication; Database Security by middleware technology implementation for requests not to contact directly database, by segmentation of network or encryption of a database (Password and Sensitive information PII Hashing or encrypting in a database).
9. Regulations to follow-

* [მომხმარებლის ძლიერი ავთენტიფიკაციის წესის დამტკიცების შესახებ | სსიპ ”საქართველოს საკანონმდებლო მაცნე” (matsne.gov.ge)](https://matsne.gov.ge/ka/document/view/4983845?publication=0);
* [პერსონალურ მონაცემთა დაცვის შესახებ | სსიპ ”საქართველოს საკანონმდებლო მაცნე” (matsne.gov.ge)](https://matsne.gov.ge/ka/document/view/1561437?publication=33);
* [პერსონალურ მონაცემთა დაცვის სათანადო გარანტიების მქონე ქვეყნების ნუსხის დამტკიცების თაობაზე | სსიპ ”საქართველოს საკანონმდებლო მაცნე” (matsne.gov.ge)](https://matsne.gov.ge/ka/document/view/2502808?fbclid=IwZXh0bgNhZW0CMTAAAR2yYE4pvgKeDDjM9sy9IUOAlY8EySwsx43e_6vRXh5YzM-XHFKcpyCKFvI_aem_AXIH7GgTH65mQfnPwtBmsynzo3Gf9LRWh_860cpPETDLytfB0xdsd63w5gh3JLXauCPNNkYqjaxXXyh9UKvye5Cl&publication=0);
* [ინფორმაციული უსაფრთხოების შესახებ | სსიპ ”საქართველოს საკანონმდებლო მაცნე” (matsne.gov.ge)](https://www.matsne.gov.ge/ka/document/view/1679424?publication=7);

1. Vulnerability Scanning - [Vulnerability Scanner - Web Application Security | Acunetix](https://www.acunetix.com/vulnerability-scanner/)
2. Restriction on unnecessary request inputs on an application fields
3. filter unnecessary ports, protocols, services, input in the beck end and else ware if database and application server coexist
4. Shortly be discussed in the presentation, about DDos Protection Tool, WAF, EDR, NGFL, PAM, SIEM, email spam filter (Email spam filter if we use email for notifications and MFA) for a future use; DDoS protection tool and WAF- Cloudflare
5. Full backup once in a week and incremental backup once in every 24 hours