

# **1 SRS Document (Software Requirements Specification):**

## **Introduction:**

### **Purpose:**

The document holds the project plan for Scholar.

### **Intended Audience:**

It's for the educational institute intending to move their student evaluation process online.

### **Intended Use:**

The system allows educational institutes to carry out the examination and report online. Helping the institute to eradicate traditional methods of taking the examination and bringing a revolution in modern approach in then educational sector.

### **Scope:**

Outcomes from the admin:

The admin has the highest control amongst the acting bodies in this application.

- a) Register teacher/student/staff
- b) Remove teacher/student/staff
- c) Update teacher/student/staff
- d) Post notice
- e) Generate a report

Outcomes from the staff:

The staff is the support system. Questions and queries asked by other bodies are resolved by this body.

- a) Register teacher/student
- b) Remove the teacher/student
- c) Update teacher/student/staff
- d) Post notice
- e) Generate a report

Outcomes from the teacher:

The teacher is the evaluation conducting body.

- a) Remove the student
- b) Generate a report
- c) Conduct exam/quiz/evaluation/problem set
- d) Set answers

Outcomes from the student:

The student is the body that attends the examination or the evaluation.

- a) Give exam or quiz
- b) View report

**Overall Description:**

### **Assumptions and Dependencies:**

- i) Students/teachers cannot register themselves.
- ii) Admin or the staff register the student/teacher.
- iii) Staff is registered by admin.

### **System Features and Requirements:**

#### **Functional Requirements:**

- i) The student should be able to take the examination.
- ii) The teacher/examiner should be able to design exam test papers.
- iii) The system should provide a platform to take exams.
- iv) The teacher/examiner should be able to generate a report about the examination.
- v) Notice board feature.
- vi) The student should be able to view the result.

#### **Non-Functional Requirements:**

Design and Implementation Constraints:

The academy has imposed a constraint to make the online system using Python (Django) for the backend and JavaScript for the front end.

External Interfaces Required:

Software requirement:

- i) Operating System: Windows 7 or above

#### Hardware Specification:

- i) RAM: 4GB
- ii) Storage: 5GB
- iii) Processor: Intel core i5(5<sup>TH</sup> generation or above)

#### Other Non-Functional Requirements:

- i) The system should be compatible with windows 7 or above.
- ii) The interface must be user-friendly.
- iii) Notifications function in the portal itself.
- iv) The system must be updated frequently.
- v) The system must have a clean-looking user interface.

#### **Goals of Implementation:**

- i) Developers will update the program if there is a new operating system or device in the future.
- ii) Developers will add new features according to the requirement on that date.
- iii) Developers will update the system in case of any bugs.