Table 1: Result of the Static Taint Analysis for XSS Vulnerabilities

Report	File	Line	Input	Classification	Motivation	Type
2	maketop.php	2	db: select schoolname from schoolinfo	FALSE	The statement prints a string extracted from the database. Since all writes to the given database field are sanitised properly against XSS attacks, the attacker has no way to store malicious content. This is a false positive.	- | -
3			Identical to entry #2.			
4			Identical to entry #2.			
5			Identical to entry #2.			
10			Identical to entry #2.			
11.a	mailtop.php	3	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
11.b	AddAssignment.php		http: $_POST['selectclass']	TRUE	The statement prints the variable $_POST['selectclass'] without any sanitisation.	3 | reflected XSS
11.c			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
13.a			http: $_POST['semester']	TRUE	The statement prints the variable $_POST['semester'] without any sanitisation.	3 | reflected XSS
13.b	AddAttendance.php	3	http: $_POST['student']	TRUE	The statement prints the variable $_POST['student'] without any sanitisation.	3 | reflected XSS
13.c			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
15.d			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
16.a	AddAnnouncement.php	3	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
16.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
18.a	AddUser.php	3	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
18.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
19.a	AddTerm.php	3	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
19.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
20	ViewAnnouncements.php	4	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
23	AddClass.php	4	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
24	StudentViewCourses.php	5	db: frame, frame --> SELECT title, startdate, enddate, type FROM studentmc WHERE userid = $_SESSION[userid]'	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the columns in the database are limited to 15 characters. To complete the attack, we split the XSS vector in 2 parts, and store each piece in a different column.	4 | stored XSS
26	AddSemester.php	5	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
27	ParentViewCourses.php	6	db: frame, frame --> SELECT studentid, frame, frame FROM studentmc WHERE userid = $_SESSION[userid]'	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the columns in the database are limited to 15 characters. To complete the attack, we split the XSS vector in 2 parts, and store each piece in a different column.	4 | stored XSS
28	ParentViewStudents.php	6	db: frame, frame --> SELECT studentid, frame, frame FROM studentmc WHERE userid = $_SESSION[userid]'	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the columns in the database are limited to 15 characters. To complete the attack, we split the XSS vector in 2 parts, and store each piece in a different column.	4 | stored XSS
29	VisualizeClasses.php	6	db: title --> SELECT title FROM semesters WHERE semesterid = $_SESSION['semester]	TRUE	The statement prints some string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
30	ViewAssignments.php	5	db: coursename --> SELECT coursename FROM courses WHERE courseid = $_POST['selectclass']	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
31			Identical to entry #30.			
32	AddAssignment.php	10	http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
			http: $_POST['selectclass']	TRUE	The statement prints the variable $_POST['selectclass'] without any sanitisation.	3 | reflected XSS
			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
38.a	VisualizeRegistration.php	10	db: frame, frame --> SELECT frame, frame FROM students WHERE userid = $_POST['student']	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
38.b			title --> SELECT title FROM semesters WHERE semesterid = $_SESSION['semester]	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
39	EditParent.php	10	db: frame, frame --> SELECT userid, frame, frame FROM parents WHERE parentid = $id[0]	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
40	EditStudent.php	10	db: frame, mi, frame --> SELECT frame, mi, frame FROM students WHERE studentid = $id[0]	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15, 2 and 15 characters respectively, which makes the attack difficult.	4 | stored XSS
41.a	AddAnnouncement.php	10	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
41.b	EditAnnouncement.php		http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
41.c			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
42	EditTeacher.php	10	db: frame, frame --> SELECT userid, frame, frame FROM teachers WHERE teacherid = $id[0]	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
43	EditUser.php	10	db: username --> SELECT username FROM users WHERE userid = $id[0]	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
44.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
44.b	EditTerm.php	10	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
44.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
45	EditClass.php	10	coursename --> SELECT coursename, teacherid, semesterid, sectionnum, roomnum, periodnum, dutie, startdate, enddate, type FROM courses WHERE courseid = $id[0]	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
46	ViewGrades.php	12	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
47	ViewClassSettings.php	12	db: SELECT apex, rpexc, rpexc, dpexc, coursename FROM courses WHERE courseid = $_POST['selectclass]	FALSE	The statement prints only values extracted from integer field in the database. So, they can not be used as a XSS vector. The statement also print the variable $id=id[0], which is always empty since the query to the database return only 5 values (and not 6). This is probably a programming bug, but makes this entry a false positive.	- | -
48			Identical to entry #46.			
49			Identical to entry #47.			
50	ClassSettings.php	12	db: SELECT apex, rpexc, rpexc, dpexc, coursename FROM courses WHERE courseid = $_POST['selectclass]	FALSE	The statement prints only values extracted from integer field in the database. So, they can not be used as a XSS vector. The statement also print the variable $id=id[0], which is always empty since the query to the database return only 5 values (and not 6). This is probably a programming bug, but makes this entry a false positive.	- | -
51	EditSemester.php	12	db: title --> SELECT title, startdate, midtermdate, enddate, type FROM semesters WHERE semesterid = $id[0]	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
52	ParentViewStudents.php	14	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
53	header.php	15	db: schoolname --> select schoolname from schoolinfo	FALSE	The statement prints a string extracted from the database. Since all writes to the given database field are sanitised properly against XSS attacks, the attacker has no way to store malicious content. This is a false positive.	- | -
54	Login.php	15	db: siteteid --> select siteteid from schoolinfo	FALSE	The statement prints the field "siteteid" extracted from a SELECT query on the database ("schoolinfo") without sanitisation. Since the INSERT query does also not perform sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack.	4 | stored XSS
55	GradeReport.php	23	db: title --> SELECT semesterid, title FROM semesters	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
56	ViewStudents.php	24	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
57	PointsReport.php	24	db: title --> SELECT semesterid, title FROM semesters	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
58	AddTeacher.php	24	u.username --> SELECT u.username, u.frame FROM users u LEFT JOIN teachers t ON u.userid = t.userid WHERE t.userid IS NULL AND (u.type='Teacher' OR u.type='Substitute')	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
59	AddParent.php	25	s.frame,s.frame --> SELECT s.studentid,s.frame,s.frame FROM students s	TRUE	The statement prints some string extracted from the database. Since the application allows to modify these values without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
60	AddSemester.php	25	db: title --> SELECT termid, title FROM terms	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
61	GradeReport.php	26	-	FALSE	The statement prints a constant string that is not under control of the user.	- | -
62	AddStudent.php	25	u.username --> SELECT u.username, u.frame FROM users u LEFT JOIN students s ON u.userid = s.userid WHERE s.userid IS NULL AND (u.type='student')	TRUE	The statement prints some string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
63.a	AddTeacher.php	26	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
63.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
70.a	AddStudent.php	28	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
70.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
71.a	AddSemester.php	28	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
71.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
76.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
76.b			http: $_POST['assignment]	TRUE	The statement prints the variable $_POST['assignment] without any sanitisation.	3 | reflected XSS
76.c	EditGrade.php	31	http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
76.d			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
76.e			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
85.a	EditSemester.php	35	http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
85.b			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
85.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
87.a	ViewClassSettings.php	36	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
87.b			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
88			Identical to entry #87.			
89.a	ClassSettings.php	36	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
89.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
89.c			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
90.a	ParentViewStudents.php	37	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
90.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
92.a	ManageSchoolInfo.php	37	db: select schoolname from schoolinfo	FALSE	The statement prints a string extracted from the database. Since all writes to the given database field are sanitised properly against XSS attacks, the attacker has no way to store malicious content. This is a false positive.	- | -
92.b			db: address --> SELECT address FROM schoolinfo	TRUE	The statement prints a string coming from a SELECT query to the database without proper sanitisation. Since the corresponding UPDATE query does not perform any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack.	4 | stored XSS
92.c			db: phonenumber --> SELECT phonenumber FROM schoolinfo	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database has a length of at most 15 characters, which makes the attack difficult.	4 | stored XSS
92.d			db: SELECT numsemesters FROM schoolinfo	FALSE	The statement prints only a variable extracted from integer field in the database, which can not be used as a XSS vector.	4 | stored XSS
92.e			db: SELECT numperiods FROM schoolinfo	FALSE	The statement prints only a variable extracted from integer field in the database, which can not be used as a XSS vector.	4 | stored XSS
92.f	AddParent.php	39	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
92.g			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
93.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
93.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
105.a	Login.php	45	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
105.b			db: sitemessage --> select sitemessage from schoolinfo	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack.	4 | stored XSS
111.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
111.b	EditTeacher.php	49	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
111.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
115.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
115.b	EditStudent.php	51	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
115.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
126.a	ViewCourses.php	56	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
126.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
128.a	StudentViewCourses.php	63	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
128.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
141.a	AddClass.php	60	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
141.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
142.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
142.b	ParentViewCourses.php	64	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
142.c			http: $_POST['student]	TRUE	The statement prints the variable $_POST['student] without any sanitisation.	3 | reflected XSS
146.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
146.b	ViewAnnouncements.php	67	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
146.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
147			Identical to entry #146.			
148			Identical to entry #146.			
149.a	EditUser.php	68	http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
149.b			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
149.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
161.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
161.b	EditParent.php	73	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
161.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
165.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
165.b	StudentMain.php	75	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
165.c			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
180.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
180.b	TeacherMain.php	83	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
180.c			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
181.a			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
181.b	ViewStudents.php	83	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
181.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
183.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
183.b			http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
183.c	ViewAssignments.php	85	http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
183.d			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
184			Identical to entry #183.			
186.a	AdminMain.php	87	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
186.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
191.a	DeficiencyReport.php	90	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
191.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
194.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
194.b	ParentMain.php	93	http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
194.c			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
194.d			http: $_POST['student]	TRUE	The statement prints the variable $_POST['student] without any sanitisation.	3 | reflected XSS
200.a			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
200.b	ViewGrades.php	102	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
200.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
201			Identical to entry #200.			
207	ManageAssignment.php	115	coursename --> SELECT coursename FROM courses WHERE courseid = $_POST['selectclass]'	TRUE	The statement prints a string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database is limited to 20 characters, which makes the attack difficult.	4 | stored XSS
212.a	PointsReport.php	121	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
212.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
230.a	VisualizeClasses.php	136	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
230.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
234.a	ManageSemesters.php	139	db: title --> SELECT semesterid,title,startdate, midtermdate,enddate,type FROM semesters ORDER BY enddate DESC	TRUE	The statement prints some string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database is limited to 15 characters, which makes the attack difficult.	4 | stored XSS
234.b			db: title --> SELECT title FROM terms WHERE termid='$term[0]'	TRUE	The statement prints some string extracted from the database. Since the application allows to modify the value without any sanitisation, it is possible to inject arbitrary HTML content and perform a reflected XSS attack. NB: the column on the database is limited to 15 characters, which makes the attack difficult.	4 | stored XSS
238.a	VisualizeRegistration.php	142	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
238.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
239.a	EditClass.php	142	http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
239.b			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
239.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
241.a	GradeReport.php	144	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
241.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
257.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
257.b	ManageAnnouncements.php	161	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
257.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
260.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
260.b	ManageTerms.php	164	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
260.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
268.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
268.b	ManageSemesters.php	175	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
268.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
269.a			http: $_POST['delete]	TRUE	The statement prints the variable $_POST['delete] without any sanitisation.	3 | reflected XSS
269.b	AddClass.php	177	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
269.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
272.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
272.b	ManageAttendance.php	181	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
272.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
273.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
273.b	ManageTeachers.php	181	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
273.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
283.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
283.b	ManageUsers.php	188	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
283.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
288.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
288.b	ManageParents.php	203	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
288.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
293.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
293.b	ManageStudents.php	211	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
293.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
299.a	Registration.php	240	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
299.b			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
309.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
309.b	ManageAssignment.php	211	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
309.c			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
309.d			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
316.a			http: $_POST['selectclass]	TRUE	The statement prints the variable $_POST['selectclass] without any sanitisation.	3 | reflected XSS
316.b	ManageGrades.php	270	http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
316.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
320.a			http: $_POST['page2']	TRUE	The statement prints the variable $_POST['page2'] without any sanitisation.	2 | reflected XSS
320.b	ManageClasses.php	303	http: $_POST['pmpage]	TRUE	The statement prints the variable $_POST['pmpage] without any sanitisation.	3 | reflected XSS
320.c			http: $_POST['page]	TRUE	The statement prints the variable $_POST['page] without any sanitisation.	1 | reflected XSS
321	ReportCards.php	539	db: multiples	FALSE	The statement generates a PDF with the data from the database. This may be vulnerable to attacks against the PDF programming language (eg. including JavaScript code in the document), but this is outside the aim of this analysis.	- | -