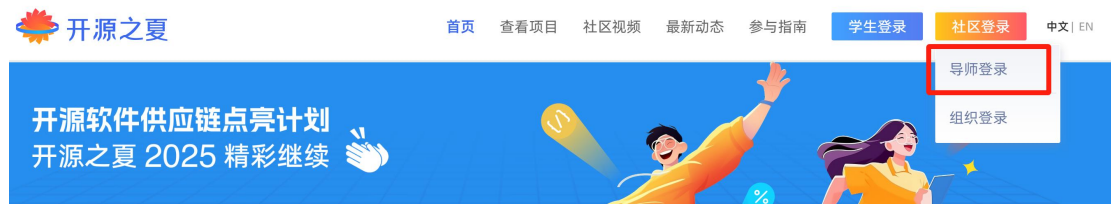


# 导师审核指南 Mentor Review Guide

## 导师账号登录

请导师通过官网 <https://summer-ospp.ac.cn/> 中的导师登录入口登录系统。如遇账号登录问题，请邮件联系组委会 [org@summer-ospp.ac.cn](mailto:org@summer-ospp.ac.cn)。



## 导师审核及排序

请点击左栏【项目管理与审核】进入项目界面，点击页面右侧【审核】按钮进入导师审核界面。



开启右上方【启用高级认证显示加密信息】，查看学生完整信息。认真审核学生项目申请书与简历内容，点击【选中并提交】选择一名学生作为中选学生，

点击【确认提交】按钮确认您的选择。



6月20日导师审核截止之前，导师可在系统中修改自己选择的学生。



审核标准

项目申请书的质量是评定学生申请是否通过的最主要标准,项目申请书将与中选名单同时在官网公示。

注意事项

- 1. 根据活动规则，学生项目申请书经过导师审核后还需社区、组委会审核通过，才能够确认中选。中选学生申请书将在活动官网公示，请各位导师务必对申请书质量进行把关。
- 2. 导师不可以通过代码的方式指导学生开发，也不可以帮助学生调试项目。项目的实现应由学生个人独立完成。导师可以围绕项目，指导学生制定计划、

提供相关推荐文档链接、指导学生如何参与社区、提供针对实现方案的改进方向性建议等。

3. 导师不可提前内定学生、不可私下向学生承诺中选结果，也不能要求学生提前开始开发工作。学生在中选结果公示之前进行的开发、提交的 PR/MR 链接不符合结项审核的时间要求，将不予认可。

### **项目预热与开发**

1. 在开发预热期间（6.28 – 6.30），指导学生使用社区常用沟通工具（电子邮箱、IRC、邮件列表等）、社区代码管理平台及其他工具，引导学生遵循社区规范，成为社区的一份子。
2. 在项目开发期间（7.1 – 9.30），与学生保持定期沟通，持续了解学生开发进度，督促学生按计划完成项目开发，建议至少保证**每周一到两次**沟通。
3. 导师不可以通过代码的方式指导学生开发，不可以帮助学生调试项目，也不能代替学生提交 PR/MR。
4. 活动无统一中期评审，社区与导师可对自行学生成果进行阶段性评估，若在开发过程中发现学生实际开发情况与预期有差异，导师有权力向组委会申请终止项目。若学生因特殊原因无法继续项目，需向导师、社区、组委会申请终止项目。
5. 提醒学生在 9 月 30 号前提交 PR/MR，并在系统中填写 PR/MR 链接、上传项目结项报告。

### **导师职责**

1. 对学生提出的项目问题进行解答。

2. 指导学生使用社区常用沟通工具（电子邮箱、IRC、邮件列表等）、社区代码管理平台及其他工具，引导学生遵循社区规范，成为社区的一份子。
3. 项目开发期间，与学生保持定期沟通，持续了解学生开发进度，督促学生按计划完成项目开发，组委会建议至少保证每周一到两次的沟通。
4. 应顾及到学生的专业水平和年龄，友好耐心地进行沟通，鼓励学生独立并积极地开展作品。
5. 推动学生的贡献被社区接受。
6. 按活动要求及时审核学生项目申请书与结项成果，按模板填写导师评语。

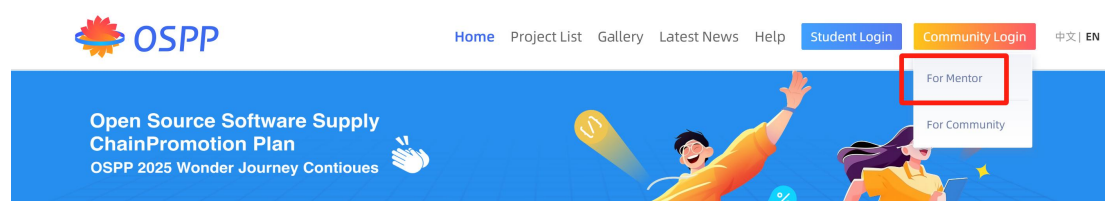
## **结项标准**

1. 学生承接的项目需要以 PR/MR 的形式提交到项目所在的开源社区仓库中并完成合并。
2. 所有 PR/MR 需由学生本人在活动规定时间内完成，学生在社区开源仓库中提交的 git email 必须使用学生报名时所用的邮箱。
3. 项目开发进程及成果与项目申请书中的计划方案及目标是否一致。
4. 评估项目产出是否符合项目设立的目标和要求。
5. 提交给社区的贡献的列表。
6. 评估项目产出运行情况。
7. 项目相关文档的完善程度。
8. 对于开发类项目，结项报告是否提供了对应的测试验证结果。
9. 社区及导师认为需要评估的其他方面。

# Mentor Review Guide

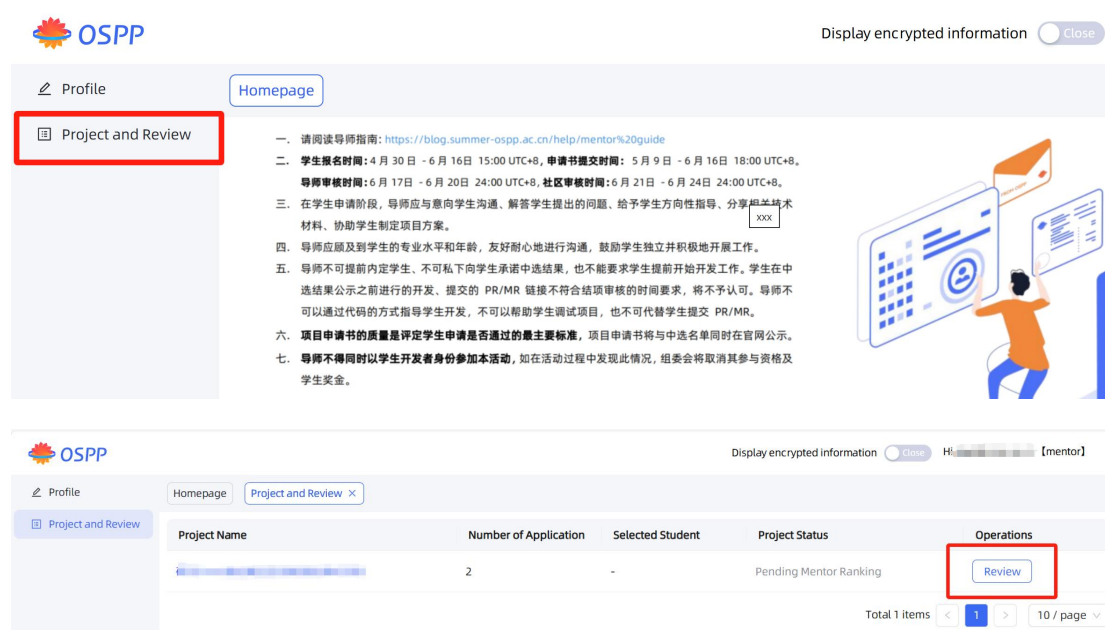
## Mentor Account Login

Please log in to the system via the mentor login portal on the official website ( <https://summer-ospp.ac.cn/> ). If you encounter any account login issues, kindly email the organizing committee at [org@summer-ospp.ac.cn](mailto:org@summer-ospp.ac.cn).



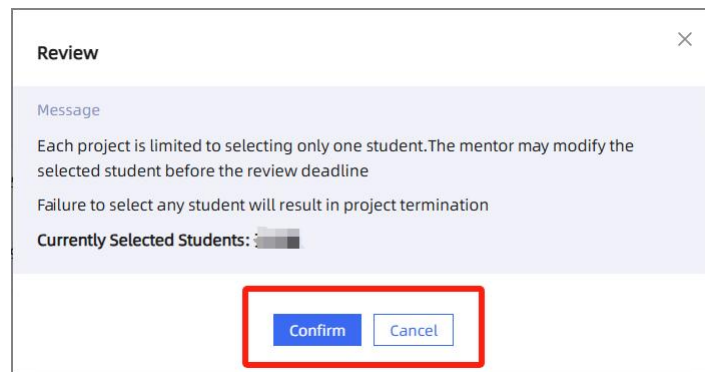
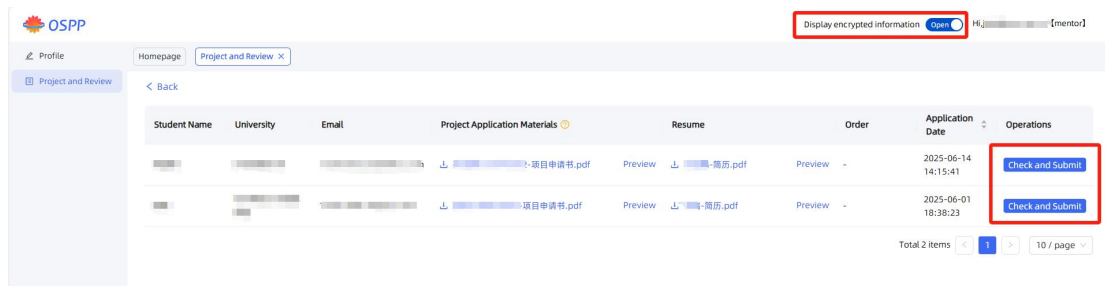
## Mentor Review and Ranking

Please click on [Project and Review] in the left column to enter the project interface, then click the [Review] button on the right side of the page to access the mentor review interface.



Enable [Enable a higher priority to display encrypted information] in the upper right corner to see the students' complete information. Carefully review the project applications and resumes.

Click the [Check and Submit] to choose one student as the selected candidate and confirm your selection.



Mentors can adjust the selection at any time in the system before the review deadline on June 20th.



## Mentor Review

The quality of the project application is the main criterion for evaluating whether the student's application is approved. Project applications will be displayed on the official website along with the list.

## Notes for Mentor Review

1. According to the rules, project applications must be approved by the community and the organizing committee after mentor approval to confirm the selection. The selected student's application will be posted on the official website. Mentors must

ensure the quality of the applications.

2. Mentors are not allowed to guide students through code, help debug projects, or submit PR/MR on behalf of students. The project should be independently completed by the student. Mentors can guide planning, relevant documentation links, community participation, and directional suggestions for project improvement.
3. Mentors are not allowed to pre-select students, make private commitments to students regarding selection, or require students to start development work early. Any development or PR/MR links submitted before the announcement will not meet the requirements and will not be recognized.

### **Bonding and Development**

1. **During the bonding period (6.28 - 6.30)**, mentors should guide students in using communication tools (email, IRC, mailing lists, etc.), community code management platforms, and other tools, and guide students to follow community rules and become a part of the community.
2. **During the project development period ( 7.1 - 9.30 )**, maintain regular communication with students to monitor their progress and ensure they complete the project as planned. Communicating **at least once or twice a week** is recommended.
3. Mentors should not guide students through code, debug projects for them, or submit PR/MR on their behalf.
4. There is no mid-term review in the program. Communities and mentors can conduct periodic evaluations of student progress. If discrepancies are found between actual student progress and expectations during development, mentors have the authority to request project termination from the committee. If a student is unable to continue the project due to some circumstances, they must apply for project termination to the mentor, community, and committee.
5. Remind students to submit PR/MR before September 30th, and upload the PR/MR

links and project report in the system.

### **Mentor Responsibilities**

1. Answer students' project-related questions.
2. Guide students to use communication tools in the community (email, IRC, mailing lists, etc.), community code management platforms, and other tools, and guide students to follow community rules and become a part of the community.
3. During the project development period, maintain regular communication with students to monitor their progress and ensure they complete the project as planned. Communicating at least once or twice a week is recommended.
4. Consider the student's professional level and age, communicate friendly and patiently, and encourage students to work independently and proactively.
5. Promote the student's contributions to be accepted by the community.
6. Timely review and evaluate students' project application and final report per the event requirements and fill out the mentor's comments using the provided template.

### **Review Criteria**

1. The project needs to be submitted to the open-source community repository in the form of PR/MR and merged.
2. All PR/MR must be completed by the student within the specified time. The git email used by the student for submissions in the community repository must match the email used during registration.
3. Whether the project development progress and outcomes meet the planned proposal and objectives outlined in the project application.
4. Evaluate whether the project output meets the goals and requirements of the project.
5. List the contributions made to the community.
6. Evaluate the running status of project outputs.
7. Evaluate the completeness of the project-related documents.



8. For R&D projects, whether the report provides the corresponding test verification results.
9. Other aspects that the community and mentor believe need to be evaluated.